



**ALAT PENGUKUR *POWER* TUNGKAI, *POWER ENDURANCE*,
DAN *WHOLE BODY REACTION* PADA ATLET
PROYEK AKHIR**

**Diajukan Kepada Fakultas Teknik Universitas Negeri Yogyakarta
Untuk Memenuhi Sebagian Persyaratan
Guna Memperoleh Gelar Ahli Madya Teknik**



**Oleh:
Rizky Maghfuri
NIM. 14507134002**

**PROGRAM STUDI TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2017**

**LEMBAR PERSETUJUAN
PROYEK AKHIR**

**ALAT PENGUKUR *POWER* TUNGKAI, *POWER*
ENDURANCE, DAN *WHOLE BODY REACTION* PADA
ATLET**



Oleh

RIZKY MAGHFURI

14507134002

Telah diperiksa dan disetujui oleh pembimbing:

Untuk diuji

Yogyakarta, 21 Agustus 2017

Mengetahui,

Kaprodi Teknik Elektronika

Dr. Sri Waluyanti, M.Pd.

NIP. 19581218 198603 2 001

Menyetujui,

Pembimbing Proyek Akhir

Drs. Totok Sukardiyono, M.T.

NIP. 19670930 199303 1 005

**LEMBAR PENGESAHAN
PROYEK AKHIR**

**ALAT PENGUKUR *POWER* TUNGKAI, *POWER*
ENDURANCE, DAN *WHOLE BODY REACTION* PADA
ATLET**

Dipersiapkan dan Disusun Oleh:

RIZKY MAGHFURI

14507134002

Telah dipertahankan didepan Dewan Penguji Proyek Akhir
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

Pada tanggal

Dan Dinyatakan Telah Memenuhi Syarat Guna Memperoleh Gelar
Ahli Madya Teknik

SUSUNAN DEWAN PENGUJI

Jabatan	Nama Lengkap dan Gelar	Tanda Tangan	Tanggal
1. Ketua Penguji	Drs. Totok Sukardiyono, M.T		26/09/2017
2. Sekretaris	Handaru Jati, Ph.D		2/10/2017
3. Penguji Utama	Dr. Putu Sudira, M.P.		15/09/2017

Yogyakarta, 24 Agustus 2017

Dekan Fakultas Teknik UNY



Dr. Widarto, M.Pd.

NIP. 19631230 198812 1 001A

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : Rizky Maghfuri
NIM : 14507134002
Program Studi : Teknik Elektronika D3
Judul Proyek Akhir : Alat Pengukur *Power* Tungkai, *Power Endurance*,
dan *Whole Body Reaction* Pada Atlet.

Menyatakan bahwa Proyek Akhir ini adalah hasil pekerjaan saya sendiri, dan sepanjang pengetahuan saya, tidak berisi materi yang ditulis oleh orang lain sebagai persyaratan penyelesaian studi di Universitas Negeri Yogyakarta atau perguruan tinggi lain, kecuali bagian-bagian tertentu saya ambil sebagai acuan dengan mengikuti kaidah penulisan karya ilmiah yang benar. Jika ternyata terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab saya.

Yogyakarta, 22 Agustus 2017

Yang menyatakan,



Rizky Maghfuri

NIM. 14507134002

MOTTO

*“ bersabar dan tetaplah bersabar menjalani kehidupan, yakinlah
keberuntungan dan keberhasilan senantiasa mengiringi ”*

(my self)

“Innallaha Ma’ash Shobirin”

(Al-Quran)

“Segala sesuatu bergantung pada niatnya ”

(Hadist Bukhari-Muslim)

“Sesungguhnya Allah Tidak Akan Mengubah Nasib Suatu Kaum Kecuali Kaum

Itu Sendiri Yang Mengubah Apa-Apa Yang Pada Diri Mereka”

(Al-Quran)

LEMBAR PERSEMBAHAN

Kupersembahkan Karyaku ini kepada orang terdekatku yang selalu mendukungku :

Untuk Orang Tuaku :

Ali Subkhan

Sunartin

Untuk Adikku :

Fatur Riyadlussholeh

Untuk Sahabatku :

Sulung Anas Abdillah

PROYEK AKHIR

ALAT PENGUKUR *POWER TUNGKAI POWER ENDURANCE* DAN *WHOLE BODY REACTION* PADA ATLET

Oleh : Rizky Maghfuri

NIM : 14507134002

ABSTRAK

Tujuan pembuatan alat ini adalah untuk mendapatkan rancang bangun *hadwere*, *software*, dan unjuk kerja Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* Pada Atlet.

Pembuatan Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* terdiri dari 8 tahapan yang mempunyai fungsi masing-masing. Pembuatan mekanik alat menggunakan bahan dasar *aluminium* yang dirangkai sedemikian rupa sesuai dengan desain alat. Alat ini menggunakan empat buah mikrokontroler yang meliputi Arduino Mega 2560 sebagai *Master Controller*, Arduino Nano v2.8 sebagai *Slave 1*, Arduino Nano v3.0 sebagai *Slave 2*, dan Wemos D1 sebagai *interface* penghubung antara hardware dengan *website*. Keempat mikrokontroler dihubungkan dengan protokol komunikasi serial UART. Komponen yang dipakai pada alat ini meliputi *Loadcell*, ultrasonik SRF04, motor Servo, motor DC *Power window*, *driver* motor mosfet H-Bridge, *rotary Encoder*, *micro SD Adapter*, *Buzzer*, Lampu Sorot *Halogen*, *proximity*, motor DC *gearbox* 400 rpm, LCD 20x4, LCD 16x2, modul *lcd to I2C*. Serta yang terakhir adalah *website* sebagai media penampil hasil pengukuran melalui *internet*.

Berdasarkan hasil pengujian unjuk kerja yang telah dilakukan dengan objek pengujian sejumlah 10 orang dari Fakultas Ilmu Keolahragaan, didapatkan hasil bahwa secara kualitas Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* pada atlet ini dapat di implementasikan dan digunakan dengan baik. Sementara secara kuantitatif diketahui bahwa rata-rata kesalahan pada semua jenis pengukuran adalah sekitar 1,2% dan dalam menjalankan algoritma, alat ini memiliki tingkat keberhasilan 100%.

Kata kunci : *Power*, *Reaction*, *Controller*, *Arduino*, *Sensor*

KATA PENGANTAR

Assamu'alaikum wr. wb.

Puji syukur penulis panjatkan kehadirat Allah SWT sehingga dengan rahmat dan hidayah-Nya Laporan Proyek Akhir ini dapat terselesaikan tanpa halangan yang berarti. Sholawat serta salam tercurah pada Qudwah kita Rasulullah SAW keluarga, sahabat dan orang-orang yang istiqomah di jalan-Nya.

Dalam menyusun Laporan Proyek Akhir ini penulis merasa banyak kekurangan karena terbatasnya kemampuan dan pengetahuan penulis. Oleh karena itu pada kesempatan ini penulis menyampaikan ucapan terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Drs. Totok Sukardiyono, M.T . selaku Dosen Pembimbing Penyusun Laporan Proyek akhir.
2. Dr. Sri Waluyanti, M.Pd. selaku Ketua Program Studi Teknik Elektronika dan Koordinator Proyek Akhir Jurusan Pendidikan Teknik Elektronika dan Informatika Universitas Negeri Yogyakarta.
3. Dr. Fatchul Arifin, M.T. selaku Ketua Jurusan Pendidikan Teknik Elektronika dan Informtaika Universitas Negeri Yogyakarta.
4. Dr. Widarto, M.Pd. selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.
5. Seluruh Dosen Pengajar dan staff karyawan Teknik Elektronika Universitas Negeri Yogyakarta atas bekal ilmu yang diberikan kepada penulis.

6. Teman-teman Fakultas Teknik UNY khususnya Teknik Elektronika kelas B 2014 yang telah memberikan bantuan sehingga pembuatan proyek akhir ini dapat terselesaikan.
7. Teman-teman Techno Kontrakan yang telah membantu dan memberikan dukungan.

Penulis menyadari bahwa laporan ini masih banyak kekurangan dan jauh dari sempurna walaupun penulis telah berusaha untuk mendekati kesempurnaan, maka penulis berharap para pembaca memberikan saran dan kritik yang membangun. Akhir kata penulis mohon maaf yang sebesar-besarnya apabila ada kekeliruan di dalam penulisan laporan ini.

Wassalamu'alaikum wr. wb.

Yogyakarta, 21 Agustus 2017

Penulis,

Rizky Maghfuri

DAFTAR ISI

PROYEK AKHIR	i
LEMBAR PERSETUJUAN PROYEK AKHIR... Error! Bookmark not defined.	
LEMBAR PENGESAHAN PROYEK AKHIR.....	ii
LEMBAR PERNYATAAN KEASLIAN	Error! Bookmark not defined.
MOTTO	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xix
DAFTAR LAMPIRAN.....	xxiii
BAB I PENDAHULUAN	1
A. Latar Belakang Masalah	1
B. Identifikasi Masalah.....	4
C. Batasan Masalah	5
D. Rumusan Masalah.....	5
E. Tujuan Proyek Akhir	6
F. Manfaat Proyek Akhir	6
G. Keaslian Gagasan.....	7
BAB II PENDEKATAN PEMECAHAN MASALAH	11
A. Power Otot Tungkai.....	11
B. Tes Pengukuran Power Tungkai	13
C. Whole Body Reaction	16
D. Tes Pengukuran Whole Body Reaction time	20
E. Power Endurance	23
F. Alat Pengukur Power Tungkai, Power Endurance, dan Whole Body Reaction	25
G. Mikrokontroler.....	28
H. Mikrokontroler Arduino	30

I. Mikrokontroler Wemos.....	48
J. Bahasa Pemrograman C.....	51
K. Perangkat Lunak IDE.....	56
L. Wireless	58
M. Website	63
N. Rotary Encoder	64
O. Liquid crystal Display (LCD).....	75
P. Bluetooth HC-05.....	78
Q. Buzzer	82
R. Sensor Ultrasonik.....	85
S. Motor DC.....	92
T. Driver Motor DC H-Bridge	96
U. Motor Servo	100
V. Sensor Berat.....	103
W. Sensor Switch	106
X. Catu Daya	110
Y. Media Penyimpanan	118
BAB III KONSEP RANCANGAN	123
A. Analisa Kebutuhan.....	123
B. Identifikasi Kebutuhan.....	124
C. Blok Diagram Sistem.....	132
D. Perancangan Rangkaian.....	135
E. Langkah Pembuatan Alat.....	141
F. Perangkat Lunak	151
G. Spesifikasi Alat	191
H. Rencana Pengujian Alat.....	193
I. Tabel Uji Alat	199
J. Pengoperasian Alat	206
BAB IV PENGUJIAN DAN PEMBAHASAN	221
A. Hasil Pengujian	221
B. Pembahasan	255
BAB V KESIMPULAN DAN SARAN.....	267

A. Kesimpulan	267
B. Keterbatasan Alat.....	270
C. Saran	271
DAFTAR PUSTAKA	272
LAMPIRAN.....	274

DAFTAR GAMBAR

Gambar	1. Autonomic Reflex Arc	19
Gambar	2. <i>Whole Body Reaction time test</i>	21
Gambar	3. Blok Pengukuran Tinggi Badan	25
Gambar	4. Blok Pengukur Berat Badan.....	25
Gambar	5. Blok Pengukur <i>Power</i> Tungkai <i>Horizontal Jump</i>	26
Gambar	6. Blok Pengukur <i>Power</i> Tungkai <i>Vertical Jump</i>	26
Gambar	7. Blok Pengukur <i>Whole Body Reaction</i>	27
Gambar	8. Blok Pengukur <i>Power Endurance</i>	27
Gambar	9. Blok diagram Alat Pengukur <i>Power</i> Tungkai <i>Power Endurance</i> dan <i>Whole Body Reaction</i>	28
Gambar	10. <i>Board</i> Arduino Mega 2560 Tampak Depan.....	32
Gambar	11. Pemetaan Pin ATmega 2560.....	33
Gambar	12. Bagian Depan Arduino Nano	41
Gambar	13. Konfigurasi Pin Layout Arduino Nano	42
Gambar	14. Mikrokontroler Wemos D1 mini.....	48
Gambar	15. Konfigurasi Pin Wemos D1 Mini.....	51
Gambar	16. IDE Perangkat Lunak Arduino.....	58
Gambar	17. <i>Rotary Encoder</i>	64
Gambar	18. Prinsip Kerja <i>Rotary Encoder</i>	65
Gambar	19. Contoh susunan pola 16 cincin konsentris pada absolut <i>encoder</i> ...	66
Gambar	20. Contoh piringan dengan 10 cincin dan 10 LED – photo-transistor untuk membentuk sistem biner 10 bit.....	67
Gambar	21. Contoh diagram keluaran absolut <i>encoder</i> 4-bit tipe gray code.....	68

Gambar 22.	Contoh diagram keluaran absolut <i>encoder</i> 4-bit tipe binary code ..	69
Gambar 23.	susunan piringan untuk <i>incremental encoder</i>	70
Gambar 24.	Contoh pola keluaran <i>incremental encoder</i>	71
Gambar 25.	<i>output</i> dan arah putaran pada resolusi yang berbeda-beda.....	72
Gambar 26.	Sinyal keluaran <i>encoder</i> untuk pengukuran kecepatan dengan <i>frequencymeter</i>	73
Gambar 27.	Pengukuran kecepatan dengan menggunakan Periodimeter	74
Gambar 28.	LCD karakter 20 x 4.....	75
Gambar 29.	Pengalamatan LCD 20 x 4	78
Gambar 30.	<i>Modul Bluetooth</i> HC-05	79
Gambar 31.	Konfigurasi Pin HC-05.....	80
Gambar 32.	<i>Bluetooth-to-Serial-Module</i> HC-05.....	80
Gambar 33.	<i>Buzzer</i>	83
Gambar 34.	<i>Piezoelectric Buzzer</i>	84
Gambar 35.	Sensor Ultrasonik	86
Gambar 36.	Prinsip Pemantulan Ultrasonik.....	87
Gambar 37.	Koneksi pada Sensor Ultrasonik SRF04	89
Gambar 38.	Konfigurasi Pin SRF 04	89
Gambar 39.	<i>Timing</i> Diagram Sensor Utrasonik SRF04.....	90
Gambar 40.	Perbandingan Sudut Pantul SRF04	92
Gambar 41.	Kaidah Tangan Kiri.....	92
Gambar 42.	Konduktor Berarus Listrik Dalam Medan Magnet	93
Gambar 43.	Bergeraknya Sebuah Motor.....	94
Gambar 44.	Konstruksi Dasar Motor DC	94

Gambar 45. Arah Putaran Motor DC	96
Gambar 46. Rangkaian Dasar H-Bridge	97
Gambar 47. H-Bridge dengan saklar Transistor	98
Gambar 48. Rangkaian <i>Driver Motor DC H-Bridge Transistor</i>	99
Gambar 49. Motor Servo.....	100
Gambar 50. Komponen Penyusun Motor Servo	101
Gambar 51. Sinyal Pulsa untuk mengendalikan Motor Servo	102
Gambar 52. Sensor <i>Loadcell</i>	103
Gambar 53. IC HX711	105
Gambar 54. <i>Proximity</i> Induktif	107
Gambar 55. <i>Proximity</i> Kapasitif	108
Gambar 56. <i>Limit switch</i>	109
Gambar 57. Blok Diagram <i>Switching Regulator</i>	111
Gambar 58. Variasi <i>Switching Regulator</i> Pada <i>Switching Power Supply</i>	112
Gambar 59. Baterai Lipo.....	116
Gambar 60. <i>Bentuk fisik</i> micro SD dan pin mapping.....	119
Gambar 61. <i>Micro SD</i> Card Adapter.....	120
Gambar 62. Skema <i>micro SD Card Adapter</i>	122
Gambar 63. Blok Diagram utama dari alat pengukur <i>power tungkai, Power Endurance</i> , dan <i>Whole Body Reaction</i> pada atlet	134
Gambar 64. Blok rangkaian Arduino Mega (<i>Master Controller</i>)	138
Gambar 65. Blok rangkaian Nano 1 (<i>Slave 1 Controller</i>)	140
Gambar 66. Blok rangkaian Nano 2 (<i>Slave 2 Controller</i>)	141
Gambar 67. Sketsa gambar mekanik rangka.....	143

Gambar 68.	Gambar 3D mekanik rangka	146
Gambar 69.	<i>Layout Driver Motor Mosfet H-Bride single channel</i>	147
Gambar 70.	Layout Shield Arduino NANO	147
Gambar 71.	<i>Layout Driver Motor Relay H-Bride single channel</i>	148
Gambar 72.	Layout Hub sensor	148
Gambar 73.	Gambar 2D Desain Box Kontrol Sistem	150
Gambar 74.	Gambar 3D Desain Box Kontrol Sistem	151
Gambar 75.	Tampilan Perangkat Lunak dari Arduino IDE	152
Gambar 76.	Tampilan dari Perangkat Lunak Notepad++	154
Gambar 77.	<i>Flowchart Sistem 1</i>	168
Gambar 78.	<i>Flowchart Sistem 2</i>	169
Gambar 79.	<i>Flowchart Master Controller Part1</i>	170
Gambar 80.	<i>Flowchart Master Controller Part2</i>	171
Gambar 81.	<i>Flowchart Master Controller Part3</i>	172
Gambar 82.	<i>Flowchart Master Controller Part4</i>	173
Gambar 83.	<i>Flowchart Master Controller Part5</i>	174
Gambar 84.	<i>Flowchart Master Controller Part6</i>	175
Gambar 85.	<i>Flowchart Master Controller Part7</i>	176
Gambar 86.	<i>Flowchart Master Controller Part8</i>	177
Gambar 87.	<i>Flowchart Master Kontroller Part9</i>	178
Gambar 88.	<i>Flowchart Master Controller Part10</i>	179
Gambar 89.	<i>Flowchart Master Controller Part11</i>	180
Gambar 90.	<i>Flowchart Master Controller Part12</i>	181

Gambar 91. <i>Flowchart Master Controller Part13</i>	182
Gambar 92. <i>Flowchart Master Controller Part14</i>	183
Gambar 93. <i>Flowchart Master Controller Part15</i>	184
Gambar 94. <i>Flowchart Master Controller Part16</i>	185
Gambar 95. <i>Flowchart Master Controller Part17</i>	186
Gambar 96. <i>Flowchart Master Controller Part18</i>	187
Gambar 97. <i>Flowchart Slave 1 Controller</i>	188
Gambar 98. <i>Flowchart Slave 2 Controller</i>	189
Gambar 99. <i>Flowchart Slave 3</i>	190
Gambar 100. Tombol <i>Emergency</i>	206
Gambar 101. Saklar <i>Driver Motor</i>	207
Gambar 102. Soket 1	207
Gambar 103. Jack DC 12V	208
Gambar 104. Soket 2.....	208
Gambar 105. Saklar <i>Slave 1 Controller</i>	209
Gambar 106. Sensor Pengukur Tinggi Lompatan pada <i>Slave 1 Controller</i>	210
Gambar 107. Kondisi pada saat <i>Slave 2 Controller</i> masih mati	210
Gambar 108. Kondisi pada saat <i>Slave 2 Controller</i> sudah hidup	211
Gambar 109. Tombol <i>Power</i> pada <i>Master Controller</i>	211
Gambar 110. Tampilan pada saat proses Booting.....	212
Gambar 111. Tampilan Menu StanBy	212
Gambar 112. Tampilan Menu <i>slide</i> satu	212
Gambar 113. Tampilan Menu <i>slide</i> dua.....	213

Gambar 114. Tampilan Hasil Pengukuran Tinggi Badan dan Berat Badan	244
Gambar 115. Tampilan Hasil Pengukuran Power Tungkai Vertikal	244
Gambar 116. Tampilan Hasil Kategori Pengukuran Power Tungkai Vertikal ...	244
Gambar 117. Tampilan Hasil Pengukuran Power Tungkai Horizontal	245
Gambar 118. Tampilan Hasil Kategori Pengukuran <i>Power</i> Tungkai Vertikal ...	245
Gambar 119. Tampilan Hasil Pengukuran <i>Whole Body Reaction</i>	247
Gambar 120. Tampilan Hasil Pengukuran <i>Power Endurance</i>	249
Gambar 121. Tampilan Menu Pada Slide pertama	251
Gambar 122. Tampilan Menu pada Slide kedua	253
Gambar 123. Diagram Hasil Pengujian Loadcell	257
Gambar 124. Diagram Hasil Pengujian Perangkat Pengukur Tinggi Badan	258
Gambar 125. Diagram Hasil Pengujian Perangkat Pengukur Panjang Lompatan	259
Gambar 126. Diagram Hasil Pengujian Sensor Ultrasonik SRF04 terhadap Tinggi Lompatan.....	260
Gambar 127. Diagram Hasil Pengujian Perangkat Pengukur <i>Whole Body Reaction</i>	261

DAFTAR TABEL

Tabel 1. Norma Tes Lompat Jauh Tanpa Awalan untuk Laki-laki.....	14
Tabel 2. Norma Tes Lompat Jauh Tanpa Awalan untuk Perempuan	15
Tabel 3. <i>Norma Lompat Tegak untuk Laki-laki</i>	16
Tabel 4. <i>Norma Lompat Tegak untuk Perempuan</i>	16
Tabel 5. Norma Tes <i>Whole Body Reaction</i>	20
Tabel 6. Spesifikasi dari Arduino Mega 2560	32
Tabel 7. Tabel Pin Serial RX dan TX	36
Tabel 8. Tabel Pin Eksternal Interupsi.....	36
Tabel 9. Tabel Pin <i>SPI</i>	36
Tabel 10. Konfigurasi Pin Arduino Nano	42
Tabel 11. Spesifikasi Wemos D1	50
Tabel 12. Konfigurasi Pin Wemos D1	51
Tabel 13. Beberapa <i>compiler C</i> untuk mikrokontroler AVR.....	53
Tabel 14. <i>Output</i> biner dan posisi yang bersesuaian pada absolute <i>encoder 4-bit</i>	69
Tabel 15. Fungsi-fungsi pin pada LCD 20x4.....	76
Tabel 16. Konfigurasi pin <i>Module Bluetooth HC-05</i>	81
Tabel 17. <i>AT Command Module Bluetooth HC-05</i>	81
Tabel 18. Identifikasi Kebutuhan.....	124
Tabel 19. Daftar komponen mekanik.....	144
Tabel 20. Rencana Pengukuran Catu Daya Tanpa Beban.....	199
Tabel 21. Rencana Pengukuran Catu Daya Dengan Beban	200
Tabel 22. Rencana pengujian Slot <i>Micro SD</i>	200

Tabel 23. Rencana Pengujian <i>Loadcell</i> terhadap Berat Badan	201
Tabel 24. Rencana Pengukuran Tinggi Badan.....	201
Tabel 25. Rencana Pengukuran Panjang Lompatan.....	201
Tabel 26. Rencana Pengukuran Ultrasonik SRF04 terhadap Tinggi Lompatan .	202
Tabel 27. Rencana Pengujian perangkat pengukur <i>Whole Body Reaction</i>	202
Tabel 28. Rencana pengujian sistem penyimpanan <i>micro SD</i>	203
Tabel 29. Rencana pengujian Komunikasi <i>Bluetooth</i>	203
Tabel 30. Rencana Pengujian <i>Pairing Bluetooth</i>	203
Tabel 31. Rencana pengujian sistem Telemetry	204
Tabel 32. Rencana Pengukuran <i>battery</i> lipo 12,6 volt 2,2A Tanpa Beban.....	204
Tabel 33. Rencana Pengukuran <i>battery</i> lipo 8,4 volt 850 mAh Tanpa Beban....	205
Tabel 34. Rencana Pengukuran <i>battery</i> lipo 12,6 volt 2,2A Dengan Beban	205
Tabel 35. Rencana Pengukuran <i>battery</i> lipo 8,4 volt 850 mAh Dengan Beban .	206
Tabel 36. Pengujian Validitas dan Reabilitas Catu Daya Tanpa Beban	222
Tabel 37. Pengujian Validitas dan Reabilitas Catu Daya Dengan Beban.....	222
Tabel 38. Pengujian Validitas dan Reabilitas Slot <i>Micro SD</i>	223
Tabel 39. Pengujian Validitas dan Reabilitas <i>Loadcell</i> terhadap Berat Badan...	224
Tabel 40. Pengujian Validitas dan Reabilitas perangkat pengukur Tinggi Badan	225
Tabel 41. Pengujian Validitas dan Reabilitas Panjang Lompatan	226
Tabel 42. Pengujian Validitas dan Reabilitas Ultrasonik SRF04 terhadap Tinggi Lompatan	227
Tabel 43. Pengujian Validitas dan Reabilitas perangkat pengukur <i>Whole Body Reaction</i>	229
Tabel 44. Pengujian Validitas dan Reabilitas sistem penyimpanan <i>micro SD</i> ...	230

Tabel 45. Pengujian Validitas dan Reabilitas Komunikasi <i>Bluetooth</i>	231
Tabel 46. Pengujian Validitas dan Reabilitas <i>Pairing Bluetooth</i>	231
Tabel 47. Pengujian Validitas dan Reabilitas sistem <i>Telemetry</i>	233
Tabel 48. Pengujian Validitas dan Reabilitas <i>battery</i> lipo 11,1 volt 2,2A Tanpa Beban	233
Tabel 49. Pengujian Validitas dan Reabilitas <i>battery</i> lipo 7,8 volt 850 mAh Tanpa Beban	234
Tabel 50. Pengujian Validitas dan Reabilitas <i>battery</i> lipo 11,1 volt 2,2A Dengan Beban	234
Tabel 51. Pengujian Validitas dan Reabilitas <i>battery</i> lipo 7,8 volt 850 mAh Dengan Beban	235
Tabel 52. Pengujian Catu Daya Tanpa Beban	235
Tabel 53. Pengujian Catu Daya Dengan Beban	236
Tabel 54. Unjuk Kerja Slot <i>Micro SD</i>	236
Tabel 55. Unjuk Kerja <i>Loadcell</i> terhadap Berat Badan	237
Tabel 56. Unjuk Kerja perangkat pengukur Tinggi Badan	237
Tabel 57. Unjuk Kerja Panjang Lompatan.....	238
Tabel 58. Unjuk Kerja Ultrasonik SRF04 terhadap Tinggi Lompatan	238
Tabel 59. Unjuk Kerja perangkat pengukur <i>Whole Body Reaction</i>	239
Tabel 60. Unjuk Kerja sistem penyimpanan <i>micro SD</i>	239
Tabel 61. Unjuk Kerja Komunikasi Bluetooth	240
Tabel 62. Unjuk Kerja <i>Pairing Bluetooth</i>	240
Tabel 63. Unjuk Kerja sistem <i>Telemetry</i>	240
Tabel 64. Unjuk Kerja <i>battery</i> lipo 11,1 volt 2,2A Tanpa Beban.....	241
Tabel 65. Unjuk Kerja <i>battery</i> lipo 7,8 volt 850 mAh Tanpa Beban.....	241
Tabel 66. Unjuk Kerja <i>battery</i> lipo 11,1 volt 2,2A Dengan Beban	242

Tabel 67. Unjuk Kerja <i>battery</i> lipo 7,8 volt 850 mAh Dengan Beban	242
Tabel 68. Tabel Uji Unjuk Kerja Pengukur Power Tungkai.....	243
Tabel 69. Tabel Uji Unjuk Kerja Pengukur Whole Body Reaction.....	246
Tabel 70. Tabel Uji Unjuk Kerja Pengukur Power Endurance	248
Tabel 71. Tabel Uji Unjuk Kerja Menu pada slide Pertama dari LCD	250
Tabel 72. Tabel Uji Unjuk Kerja Menu pada <i>slide</i> kedua dari LCD	252
Tabel 73. Tabel Uji Unjuk Kerja Tampilan Data.....	254

DAFTAR LAMPIRAN

Lampiran 1. Gambar Alat	275
Lampiran 2. Program <i>Master Controller</i> (Arduino Mega)	282
Lampiran 3. Program <i>Slave 1 Controller</i> (Arduino NANO 1)	356
Lampiran 4. Program <i>Slave 2 Controller</i> (Arduino NANO 2)	367
Lampiran 5. Program <i>Slave 3</i> (Wemos D1)	375
Lampiran 6. <i>Datasheet</i> Arduino Mega 2560.....	380
Lampiran 7. <i>Datasheet</i> Arduino Nano 328p	388
Lampiran 8. <i>Datasheet</i> Ultrasonik HCSR04	395
Lampiran 9. <i>Datasheet</i> modul Bluetooth HC05	401
Lampiran 10. <i>Datasheet</i> ESP8266	407
Lampiran 11. <i>Datasheet</i> LCD 16x2	410
Lampiran 12. <i>Datasheet Rotary Encoder Autronics</i>	412
Lampiran 13. Skema Rangkaian <i>Master Controller</i>	415
Lampiran 14. Skema Rangkaian <i>Slave 1 Controller</i>	416
Lampiran 15. Skema Rangkaian <i>Slave 2 Controller</i>	417
Lampiran 16. Skema Rangkaian <i>Shield</i> Arduino Nano	418
Lampiran 17. Skema Rangkaian <i>Driver Motor H-Bridge Mosfet</i>	419
Lampiran 18. Skema Rangkaian HUB Sensor	420
Lampiran 19. Daftar List Komponen dan Biaya	421

BAB I PENDAHULUAN

A. Latar Belakang Masalah

Inovasi di bidang teknologi instrumentasi dan kendali berkembang dengan cepat. Hal ini selaras dengan perkembangan karakteristik masyarakat yang memiliki mobilitas tinggi. Masyarakat kini menginginkan layanan yang fleksibel, mudah, memuaskan, efisien, dan aman tak terkecuali di bidang olahraga. Olahraga merupakan ilmu terapan yang terpengaruh oleh adanya teknologi untuk mendukung ketika melakukan suatu aktivitas olahraga. Olahraga prestasi merupakan aktivitas fisik yang dilakukan untuk mencapai target prestasi setinggi-tingginya. Olahraga prestasi memerlukan banyak dukungan dari berbagai disiplin ilmu demi mencapai prestasi secara maksimal. Teknologi dalam olahraga prestasi digunakan oleh para pelatih dan atlet untuk menunjang proses latihan agar maksimal.

Cabang olahraga atletik merupakan aktivitas jasmani yang terdiri dari gerakan-gerakan dasar yang dinamis dan harmonis, yaitu jalan, lompat, lari, dan lempar (Edy Purnomo, 2007:1). Dari keempat nomor tersebut secara resmi ada 44 event yang dipertandingkan pada kegiatan multievent seperti PON, *Sea Games* dan Olimpiade. Setiap nomor cabang atletik memiliki karakteristik yang berbeda. Banyak faktor yang mempengaruhi prestasi atlet diantaranya: kecepatan, kekuatan, kordinasi, fleksibilitas dan daya tahan. Semua aspek ini sangatlah dibutuhkan oleh setiap atlet. Setiap pelatih harus mengetahui seberapa jauh kemampuan seorang atlet dalam menguasai semua aspek ini.

Dengan mengetahui kekuatan, ketahanan, dan kecepatan setiap atlet, pelatih dapat memberikan evaluasi dan koreksi terhadap atlet tentang kelebihan dan kekurangannya. Pengukuran terhadap aktivitas latihan prestasi pada atlet seperti pengukuran kekuatan tungkai/*power tungkai* ada dua macam (*vertikal jump* dan *standing board jump*) *vertical jump* dilakukan menggunakan alat jam meter digital dan *standing board jump* dilakukan menggunakan matras dengan pita meteran, pengukuran kekuatan dan ketahanan atau *Power Endurance* dilakukan dengan tes (2,4 km lari, 4,8 km jalan, lari 12 menit, lari 15 menit *harvard test*, dan *sharky test*), dan pengukuran kecepatan reaksi atau *Whole Body Reaction* ada dua macam (*visual* dan *audiovisual*) dengan keterangan *visual* menggunakan alat indra mata atau cahaya sedangkan *audiovisual* menggunakan suara dari *buzzer* dan cahaya. Dengan mengetahui semua aspek tersebut pelatih dapat mengetahui karakteristik atlet yang dilatihnya. Fakta yang terjadi dilapangan pelatih maupun atlet sulit mengontrol dan mengetahui berapa kekuatan, ketahanan, dan kecepatan pada saat latihan maupun perlombaan. Kendala lain yang terjadi dilapangan adalah perlu banyaknya orang yang diperlukan dalam setiap pengambilan data dilapangan kemudian peralatan yang dibutuhkan juga banyak. Hal itu menjadi kesulitan tersendiri bagi pelatih dalam melakukan koreksi terhadap atletnya.

Melalui bantuan teknologi yang telah berkembang misalkan *stopwatch* dan *dynamometer*, pelatih dapat terbantu dalam melakukan analisis dengan mengambil data kekuatan, ketahanan, dan kecepatan setiap atletnya. Namun hasil analisis menggunakan alat tersebut kurang akurat, harus dikonversi lagi menjadi data yang sesuai dengan harapan pelatih. Teknologi lain yang telah

ada saat ini adalah kamera yang dapat mengikuti pergerakan atlet. Teknologi ini membutuhkan biaya yang sangat mahal dikarenakan harus membeli komputer dengan spesifikasi yang tinggi agar dapat mengukur secara *realtime*. Kamera yang dibutuhkan juga harus memiliki resolusi yang tinggi untuk mendeteksi setiap gerakan atlet ketika bergerak.

Oleh karena itu diperlukan alat yang dapat memantau aktivitas atlet pada saat melakukan pengukuran yang akurat, reliabel dan dengan biaya yang cukup murah. Berdasarkan permasalahan diatas penulis mempunyai tawaran solusi yaitu dengan membuat “Alat Pengukur *Power Tungkai Power Endurance* dan *Whole Body Reaction* Pada Atlet”. Alat ini merupakan gabungan dari beberapa elemen yaitu mekanik, elektronik dan program. Mekanik dari alat ini terbuat dari bahan aluminium yang ringan dan kuat. Beberapa sensor digunakan untuk melakukan pemantauan terhadap fenomena fisik seperti kekuatan, ketahanan, dan kecepatan atau aliran ke dalam besaran listrik. Besaran listrik yang dihasilkan sebanding dengan percepatan yang dialaminya. Nilai parameter fisik awal dapat dihitung ulang dengan mengacu karakteristik sinyal listrik *amplitude, frekuensi* atau lebar denyut listrik. Sinyal listrik tersebut selanjutnya dapat dianalisis secara *real time* maupun untuk keperluan analisa kemudian. Gabungan antara mekanik dan penerapan sensor-sensor ini akan diintegrasikan ke dalam beberapa buah sistem pengontrol. Ada empat sistem kontrol yang dipakai yaitu Arduino Mega 2560 (*Master Controller*) sebagai pusat dari semua sistem yang terintegrasi, Arduino Nano 1 (*Slave 1 Controller*) sebagai pengendali sensor yang dipergunakan untuk pengukuran tinggi lompatan / *vertical jump*, Arduino Nano 2 (*Slave 2 Controller*) sebagai

pengendali mekanisme sistem pengukuran panjang lompatan / *standing board jump*, dan yang terakhir adalah Wemos D1 (*Slave 3 Controller*) sebagai perangkat yang digunakan untuk media pengiriman data pengukuran yang sudah dilakukan ke *database website* secara *online*.

Dibuatnya “Alat Pengukur *Power Tungkai Power Endurance* dan *Whole Body Reaction* Pada Atlet” ini diharapkan akan mempermudah kinerja pelatih dalam melakukan tes pengukuran. Selain itu, kemampuan alat ini untuk mengupload data ke web akan mempermudah pelatih maupun asisten pelatih untuk sama-sama melihat dan mengoreksi data hasil latihan dari atletnya itu sendiri. Kemampuan ini sangatlah berguna untuk menunjang pelatih atau asisten pelatih bahkan orang lain untuk sama-sama melihat dan menganalisis.

B. Identifikasi Masalah

Berdasarkan latar belakang di atas maka dapat diketahui permasalahan yang ada. Permasalahan tersebut dapat diidentifikasi sebagai berikut:

1. Belum banyak inovasi terhadap peralatan olahraga khususnya cabang olahraga atletik.
2. Peralatan yang dapat membantu pelatih dalam menganalisis atletnya masih minim dan harga masih relatif mahal.
3. Pengukuran kemampuan fisik dilakukan dengan sistem manual menggunakan alat konvensional sehingga hasil pengukuran kurang maksimal.
4. Alat olahraga multifungsi sebagai pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* yang bersifat terintegrasi masih jarang bahkan belum ada khususnya di Indonesia.

C. Batasan Masalah

Berdasarkan identifikasi masalah di atas permasalahan dibatasi pada point 4, yaitu belum adanya alat pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* yang bersifat terintegrasi. Alat pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* yang akan dibuat dengan komponen utama Mikrokontroler Arduino Mega 2560 sebagai *Master Controller* untuk mengontrol seluruh sistem dan didalamnya terdiri dari *loadcell*, *rotary encoder*, ultrasonik, *limit switch*, *modul micro SD Adapter*, modul *Bluetooth HC-05*, lcd karakter 20x4, *push button*, motor DC *power window*, *driver motor*, dan *buzzer*. Arduino Nano 1 sebagai *Slave 1 Controller* untuk mengukur *vertical jump* dan didalamnya terintegrasi sensor ultrasonik, sensor *proximity*, dan modul *Bluetooth HC-05*. Arduino Nano 2 *Slave 2 Controller* untuk mengukur *standing board jump* dan didalamnya terintegrasi sensor *rotary encoder*, sensor warna LDR dan led, sensor *proximity*, motor DC *gearbox 400 rpm*, *driver motor relay*, lcd karakter 16x2, dan modul *Bluetooth HC-05*. Wemos D1 sebagai *Slave 3 controller* sebagai perangkat untuk mengirim data hasil pengukuran ke *database website* secara *online*.

D. Rumusan Masalah

Berdasarkan identifikasi masalah diatas maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana membangun Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* Pada Atlet?
2. Bagaimana membangun *software* Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* Pada Atlet?

3. Bagaimana unjuk kerja Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* Pada Atlet?

E. Tujuan Proyek Akhir

Adapun tujuan dari penelitian ini adalah untuk:

1. Merealisasikan rancangan Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* Pada Atlet.
2. Merealisasikan rancangan *software* Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* Pada Atlet.
3. Mengetahui unjuk kerja Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* Pada Atlet.

F. Manfaat Proyek Akhir

Dari pembuatan Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* Pada Atlet ini dapat diperoleh beberapa manfaat bagi pihak-pihak terkait, antara lain:

1. Bagi Institusi / Perguruan Tinggi

Dapat dijadikan tolak ukur daya serap mahasiswa yang bersangkutan selama menempuh pendidikan di perguruan tinggi dan mampu menerapkan ilmunya secara praktis pada bidang-bidang yang sesuai dengan program studi yang dipelajari.

2. Bagi Mahasiswa

- a. Mendapat pengalaman mengaplikasikan ilmu pengetahuan yang diperoleh di bangku kuliah untuk mengatasi permasalahan dalam kehidupan *real*.

- b. Menghasilkan karya yang dapat dipakai sebagai sumber informasi dan referensi untuk pengembangan selanjutnya dalam rangka peningkatan ilmu pengetahuan dan teknologi.
3. Bagi Dunia Olahraga
 - a. Memaksimalkan kinerja pelatih dalam melakukan pengukuran *power tungkai*, *Power Endurance*, dan *Whole Body Reaction* pada atlet.
 - b. Mewujudkan alat pengukuran pada olahraga prestasi yang memenuhi syarat sehingga dapat menunjang prestasi atlet untuk meningkatkan kemampuannya.
 - c. Merupakan inovasi terbaru terhadap Alat konvensional yang lebih efektif dan efisien saat digunakan untuk latihan.

G. Keaslian Gagasan

Pembuatan Proyek akhir ini sebagai riset dari penelitian dengan dosen Fakultas Ilmu Keolahragaan Universitas Negeri Yogyakarta yang merupakan inovasi baru dalam dunia industri olahraga. Alat ini terinspirasi dari banyaknya tes pengukuran dalam olahraga prestasi yang masih menggunakan alat ukur konvensional.

1. Alat yang hampir sama dengan tugas akhir ini yang pernah dibuat beserta kekurangannya yaitu:
 - a. Pengembangan *Jump Power Meter* Sebagai Alat Pengukur *Power Tungkai*. Dibuat oleh Sri Haryono dkk, dari Universitas Negeri Semarang tahun 2012. Kekurangan dari alat ini yaitu:
 - 1) Hanya mengukur *vertical jump*, berat badan, dan *power tungkai*.

- 2) Sistem pengukuran tinggi lompatan menggunakan *infrared* yang belum menggunakan metode parsing data.
- 3) Masih menggunakan *plate* atau papan lompat.
- 4) Sensor *infrared* pendeteksi tinggi lompatan diletakkan ditengah-tengah *plate*. Kelemahannya ketika kaki mulai lompat dan melenceng dari *plate* maka tinggi lompatan tidak bisa terdeteksi.

b. Pengembangan Alat Ukur Tinggi Badan dan Berat Badan Digital yang terintegrasi. Dibuat oleh Afif Muhammad Khoiruddin dari Universitas Negeri Yogyakarta tahun 2015. Kekurangan dari alat ini adalah:

- 1) Masih menggunakan sensor ultrasonic untuk mengukur tinggi badan.
- 2) Proses pengukuran tinggi dan berat badan masih secara sederhana, data hasil pengukuran tidak disimpan pada memori.

c. Gym Matras Meter (pengukur *Power Tungkai Standing board jump* manual)

Kekurangan dari alat ini yaitu proses pengukuran masih bersifat konvensional. Data hasil pengukuran masih diperoleh secara manual dan hal itu bisa menghambat pada saat pengukuran dengan banyak testi.

d. *Takei equipment industrial T.K.K.5408 reaction Whole Body Reaction measuring instrument*. Alat ini dibuat oleh produsen jepang yang cukup terkenal yaitu perusahaan *Takei*. Kekurangan dari alat ini hanya harganya saja yang sangat mahal, tidak semua lab olahraga prestasi di Indonesia mempunyai alat ini.

2. Kelebihan Alat Pengukur *Power Tungkai Power Endurance* dan *Whole Body Reaction* Pada Atlet yang saya buat pada proyek akhir ini adalah:

- a. Pengukuran tinggi badan, berat badan, *vertical jump*, *standing board jump*, *Whole Body Reaction*, dan *Power Endurance* terintegrasi jadi satu kesatuan sistem.
- b. Pengukuran tinggi badan tidak menggunakan ultrasonik atau infrared lagi, melainkan menggunakan mekanik tiang pengukur yang bergerak otomatis kebawah yang digerakkan oleh motor DC *power window* dan diukur oleh sensor *rotary encoder*. Hasil dari pengukuran ini lebih akurat dan presisi.
- c. Pengukuran *standing board jump* dilakukan secara otomatis menggunakan mekanik *slider* yang bergerak pada tiang lurus dan dilengkapi dengan sensor *rotary encoder* untuk menghitung jarak lompatan yang terdeteksi. Pengiriman data dari marker ini ke kontroler utama menggunakan *Bluetooth* secara serial, sehingga lebih praktis dan efisien.
- d. Pengukuran *vertical jump* menggunakan marker yang didalamnya terdapat sensor ultrasonik untuk mengukur tinggi lompatan dan sebuah mikrokontroler untuk mengolah data yang didapat menjadi data parsing. Pengiriman data dari marker ini ke kontroler utama menggunakan *Bluetooth* secara serial, sehingga lebih praktis dan efisien.
- e. Mampu melakukan pengukuran *Power Endurance* dengan menggunakan perangkat yang sama seperti yang digunakan pada pengukuran *vertical jump*.

- f. Semua pengukuran yang sudah dilakukan akan langsung di analisis saat itu juga dengan menggunakan rumus baku yang sudah di masukkan ke dalam pemrograman alat.
- g. Mempunyai dua sistem penyimpanan yaitu menggunakan memori internal EEPROM dan memori eksternal *Micro SD Card*.
- h. Bisa mengupload data ke *database* secara *online* melalui mikrokontroler Wemos D1.
- i. Semua data hasil pengukuran akan otomatis tampil pada website www.smartech515.xyz

BAB II

PENDEKATAN PEMECAHAN MASALAH

A. *Power* Otot Tungkai

Power merupakan kombinasi dari komponen kondisi fisik kekuatan dan kecepatan. Stamina merupakan kombinasi dari komponen kondisi fisik daya tahan dan kecepatan, sedangkan daya tahan kekuatan kombinasi dari komponen kondisi fisik daya tahan dan kekuatan. Dikutip dari Pyke & Watson (1978) Oleh Ismaryati (2008: 59), *power* atau daya ledak disebut juga sebagai kekuatan eksplosif. Daya ledak menurut Suharno H.P (1981: 23), yaitu kemampuan sebuah otot atau segerombolan otot untuk mengatasi tahanan beban dengan kekuatan dan kecepatan tinggi dalam satu gerakan yang utuh. Sedangkan Sukadiyanto (2002: 96), berpendapat *Power* sebagai kemampuan otot untuk menggerakkan kekuatan maksimal dalam waktu yang sangat singkat. Berdasarkan Tim Fisiologi Manusia (2010: 45), *Power* adalah hasil kali kekuatan dengan kecepatan. sehingga satuan *power* adalah kg (berat) * meter/detik. Sedangkan Kg * meter adalah satuan usaha, dengan demikian *power* dapat diartikan usaha per detik.

Daya eksplosif atau tenaga cepat adalah kemampuan sistem otot untuk mengatasi tahanan dengan kontraksi yang tinggi (U. Jonath, dkk 1985: 15). Sedangkan daya ledak menurut Tjaliek Soegiardo, 1992: 79), mengemukakan kemampuan kerja otot (usaha) dalam satuan waktu (detik). *Power* berbanding lurus dengan kekuatan otot, maka besar kecilnya *power* antara lain juga ditentukan oleh besar kecilnya kekuatan otot.

Power merupakan komponen yang sangat penting dan bermanfaat untuk mencapai prestasi yang optimal bagi setiap cabang olahraga baik putra maupun putri. Menurut Febri Ikhwanudin (2011: 14) Berikut ini faktor yang mempengaruhi *explosive power*, yaitu:

1. Banyak sedikitnya macam fibril otot putih/serabut otot cepat (*Fast Twitch*).
2. Kekuatan dan kecepatan otot, $power (P) = Force (F) \times Velocity (V)$.
3. Banyak sedikitnya zat kimia dalam otot (ATP).
4. Koordinasi gerak yang harmonis. Menurut Suharno H.P. yang dikutip Ridwan Maulana. (2010: 11), faktor yang mempengaruhi daya ledak atau *power* adalah:

- a. Banyak sedikitnya macam fibril otot putih tiap individu.
- b. Kekuatan otot dan kecepatan otot. Rumus *power* adalah sebagai berikut:

Keterangan: P : *Power* (daya ledak = kg.m/detik F : *Force* (kuat = kg) V : *Velocity* (kecepatan = m/detik

- c. Koordinasi gerak yang harmonis.
- d. Tergantung banyak sedikitnya zat kimia dalam otot.
- e. Pelaksanaan teknik yang betul. Dalam Buku Petunjuk Praktikum Fisiologi Manusia (2010: 45). *Power* (daya ledak) ada 2 bagian yaitu:

- 1) Kekuatan daya ledak

kekuatan daya ledak merupakan kekuatan yang digunakan untuk mengatasi resistensi yang lebih rendah, tetapi dengan percepatan daya ledak maksimal. *Power* ini sering digunakan untuk melakukan satu gerakan atau satu ulangan (lompat jauh, lempar cakram, lempar lembing, dan tolak peluru).

2) kekuatan gerak cepat

kekuatan gerak cepat merupakan gerakan yang dilakukan terhadap resistensi dengan percepatan di bawah maksimal, jenis ini digunakan untuk melakukan gerakan yang berulang-ulang. (berlari, dan mengayuh).

Berdasarkan pengertian dan pendapat mengenai *power*, maka dapat disimpulkan bahwa *power* adalah perpaduan atau penggabungan antara kekuatan dan kecepatan. Maksudnya kekuatan dapat dikatakan *power* apabila dilakukan dengan sangat cepat. Dalam rangka peningkatan prestasi olahraga, komponen kondisi fisik *power* perlu mendapatkan perhatian khusus, terutama *power* tungkai dan *power* lengan. Karena untuk mencapai prestasi puncak bukan hanya kekuatan saja yang diperlukan, tetapi diperlukan juga peningkatannya bagi komponen fisik kekuatan ini, yaitu *power*. Dan kekuatan merupakan dasarnya untuk membentuk *power*. Sesuai pendapat Harsono (2008:177) bahwa “*strength* tetap merupakan dasar (basis) dari *power* dan daya tahan otot”.

B. Tes Pengukuran *Power* Tungkai

Tes menurut Rusli Lutan dan Ismaryati (2006: 1), tes adalah instrumen atau alat yang digunakan untuk memperoleh informasi tentang individu atau objek. Dari ketiga pendapat para ahli tentang tes adalah alat atau instrumen yang dibuat oleh testor kepada testee dalam bentuk teori atau praktik (keterampilan, pengetahuan, inteligensi, kemampuan atau bakat) untuk memperoleh data atau informasi tentang testee tersebut dan dibandingkan dengan standar atau norma yang sudah ada. Pengukuran merupakan kumpulan

informasi dari suatu yang diukur, hasilnya hanyalah data-data atau angka-angka hasil pengukuran (Mochamad Sajoto, 1988: 60). Tes pengukuran yang dilakukan untuk mengetahui *power* seseorang terdapat beberapa cara dalam pengambilannya. Pengukuran dilakukan secara langsung dan tidak langsung. Pengukuran langsung dilakukan dengan menggunakan *vertical jump*, *margaria test* dan *standing long jump*, sedangkan tak langsung diaplikasikan penggunaan *power* seperti lompat jauh, tes lempar, Smash dll.

1. Pengukuran langsung yang dilakukan untuk mengetahui *power* otot tungkai terdapat 2 (tiga) cara:

a. tes lompat jauh tanpa awalan (*standing broad jump*) untuk mengukur daya ledak otot tungkai ke arah depan. Adapun prosedur pelaksanaannya adalah sebagai berikut:

- 1) Atlet berdiri di belakang garis batas, kedua kaki sejajar, lutut ditekuk dan kedua lengan ke belakang.
- 2) Tanpa menggunakan awalan, kedua kaki menolak secara bersamaan dan melompat kedepan sejauh-jauhnya.
- 3) Pelaksanaan lompatan dilakukan dengan bantuan ayunan lengan.
- 4) Jarak lompatan dihitung dari garis batas awal saat melompat sampai dengan batas akhir terdekat anggota badan yang menyentuh matras.

Tabel 1. Norma Tes Lompat Jauh Tanpa Awalan untuk Laki-laki

Norma	Usia							
	10	11	12	13	14	15	16	17>
Baik Sekali	5'8"- 6'5"	5'10"- 8'5"	6'1"- 7'5"	6'8"- 8'6"	6'11"- 9'0"	7'5"- 9'0"	7'9"- 9'2"	8'0"- 9'10"

Baik	5'1"- 5'6"	5'6"- 5'9"	5'8"- 6'0"	6'0"- 6'5"	6'6"- 6'11"	6'11"- 7'3"	7'3"- 7'6"	7'6"- 7'10"
Cukup	4'10"- 5'1"	5'2"- 5'6"	5'4"- 5'7"	5'7"- 6'0"	6'1"- 6'4"	6'6"- 6'10"	6'11"- 7'2"	7'1"- 7'5"
Kurang	4'6"- 4'9"	4'8"- 5'0"	5'0"- 5'3"	5'2"- 5'6"	5'6"- 5'11"	6'1"- 6'5"	6'6"- 6'9"	6'6"- 7'0"
Kurang Sekali	3'10"- 4'5"	4'0"- 4'7"	4'2"- 4'10"	4'4"- 5'0"	4'8"- 5'4"	5'2"- 5'11"	5'5"- 6'4"	5'3"- 6'4"

(Sumber : Johnson & Nelson, 2000)

Tabel 2. Norma Tes Lompat Jauh Tanpa Awalan untuk Perempuan

Norma	Usia							
	10	11	12	13	14	15	16	17>
Baik Sekali	5'5"- 7'11"	5'7"- 7'0"	5'9"- 7'0"	6'0"- 8'0"	6'3"- 7'5"	6'1"- 8'0"	6'0"- 7'7"	6'3"- 7'6"
Baik	5'0"- 5'2"	5'2"- 5'5"	5'4"- 5'8"	5'6"- 5'10' '	5'8"- 6'0"	5'8"- 6'0"	5'6"- '	5'10"- -6'2"
Cukup	4'7"- 4'10"	4'10"- -5'1"	4'11"- -5'2"	5'2"- 5'5"	5'3"- 5'7"	5'3"- 5'6"	5'2"- 5'6"	5'4"- 5'9"
Kurang	4'1"- 4'6"	4'4"- 4'8"	4'6"- 4'10"	4'9"- 5'1"	4'10"- -5'2"	4'11"- -5'2"	4'9"- 5'1"	4'11"- -5'3"
Kurang Sekali	3'5"- 4'0"	3'8"- 4'3"	3'10"- -4'5"	4'0"- 4'8"	4'0"- 4'9"	4'2"- 4'9"	4'0"- 4'7"	4'1"- 4'9"

(Sumber : Johnson & Nelson, 2000)

- b. tes lompat tegak (vertical jump) untuk mengukur daya ledak otot tungkai ke arah atas. Adapun prosedur pelaksanaannya adalah sebagai berikut:
- 1) Gantungkan papan ukuran lompat tegak di tembok.
 - 2) Atlet berdiri menyamping dan kaki kanan/kiri merapat ke tembok.
 - 3) Tangan kanan/kiri memegang kapur dan diluruskan ke atas setinggi-tingginya dan disentuh pada papan ukuran lompat tegak. Bekas sentuhan yang tertinggi merupakan tinggi raihan. Atlet siap melompat.

- 4) Atlet melompat setinggi-tingginya dengan bantuan ayunan kedua tangannya.
- 5) Saat melompat, sentuhkan jari-jari tangan yang berkapur ke papan ukuran.
- 6) Selisihkan tinggi raihan dengan hasil raihan pada saat melompat.

Tabel 3. Norma Lompat Tegak untuk Laki-laki

Norma	Usia									
	9	10	11	12	13	14	15	16	17	18>
Baik Sekali	16"	16"	16"	20"	20"	20"	25"	25"	25"	26"
Baik	14"	14"	14"	17"	17"	17"	23"	23"	23"	24"
Cukup	11"	11"	11"	14"	14"	14"	19"	19"	19"	19"
Kurang	9"	9"	9"	11"	11"	11"	12"	12"	12"	13"
Kurang Sekali	4"	4"	4"	5"	5"	5"	5"	5"	5"	8"

(Sumber : Johnson & Nelson, 2000)

Tabel 4. Norma Lompat Tegak untuk Perempuan

Norma	Usia									
	9	10	11	12	13	14	15	16	17	18>
Baik Sekali	16"	16"	16"	16"	16"	16"	17"	17"	17"	14"
Baik	14"	14"	14"	14"	14"	14"	15"	15"	15"	13"
Cukup	11"	11"	11"	12"	12"	12"	13"	13"	13"	10"
Kurang	9"	9"	9"	10"	10"	10"	8"	8"	8"	6"
Kurang Sekali	4"	4"	4"	4"	4"	4"	3"	3"	3"	2"

(Sumber : Johnson & Nelson, 2000)

C. *Whole Body Reaction*

Semua respon yang timbul secara otomatis dapat disebut dengan *reflex*. Tentu saja jenis *reflex* ini bermacam-macam. Ada yang tergolong *reflex* dasar (sudah ada sejak lahir dan tidak perlu dipelajari) misalnya : *reflex* berkedip, *reflex* bersin, *reflex* menjauhi rangsang nyeri. Namun ada juga *reflex* yang

didapat dengan latihan atau belajar (*acquired atau conditional reflex*), misalnya pada pemain piano yang sudah terlatih bertahun-tahun, maka begitu dia melihat not balok tertentu maka jarinya secara otomatis menekan tuts piano tertentu. Karena itu Sherwood mendefinisikan *reflex* sebagai “*any response that occur automatically without conscious effort*”.

Menurut Suharno HP (1981: 25) kecepatan reaksi adalah kemampuan organisme anak latih untuk menjawab suatu rangsang secepat mungkin dalam mencapai hasil yang sebaik – baiknya. Menurut Devi Tirtawirya (2006: 68-69) kecepatan reaksi dibagi menjadi dua macam yaitu :

1. kecepatan reaksi tunggal adalah kecepatan reaksi yang dalam pelaksanaannya sudah diketahui arah dan sasarannya.
2. kecepatan reaksi majemuk adalah kemampuan seorang atlet dalam melakukan suatu gerakan akibat rangsangan yang belum diketahui arah dan sasarannya dalam waktu yang sesingkat mungkin.

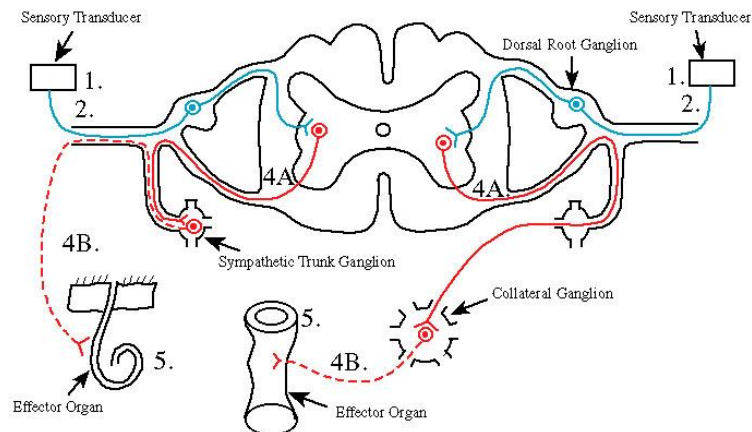
Menurut Sukadiyanto dalam Devi Tirtawirya (2006: 68) kecepatan reaksi adalah kemampuan seseorang dalam menjawab rangsangan dalam waktu sesingkat mungkin. Andi Suhendro (2002: 4. 25) mengungkapkan bahwa kecepatan reaksi adalah kecepatan menjawab suatu rangsangan dengan cepat, rangsangan itu berupa suara atau pendengaran. Menurut Muhajir (2007: 175) kecepatan reaksi adalah kemampuan seseorang untuk menjawab sebuah rangsangan secepat mungkin untuk mendapatkan hasil yang sebaik baiknya. Hampir semua cabang olahraga memerlukan kecepatan reaksi di dalam pertandingan.

Gerak merupakan pola koordinasi yang sangat sederhana untuk menjelaskan penghantaran impuls oleh saraf. Gerak pada umumnya terjadi secara sadar, namun, ada pula gerak yang terjadi tanpa disadari yaitu gerak *reflex*. Impuls pada gerakan sadar melalui jalan panjang, yaitu dari reseptor ke saraf sensori lalu dibawa ke otak untuk selanjutnya diolah oleh otak, Kemudian hasil olahan oleh otak berupa tanggapan dibawa oleh saraf motor sebagai perintah yang harus dilaksanakan oleh efektor.

Gerak *reflex* berjalan sangat cepat dan tanggapan terjadi secara otomatis terhadap rangsangan tanpa memerlukan kontrol dari otak. Jadi dapat dikatakan gerakan terjadi tanpa dipengaruhi kehendak atau tanpa disadari terlebih dahulu. Contoh gerak *reflex* misalnya berkedip, bersin, atau batuk.

Pada gerak *reflex*, impuls melalui jalan pendek atau jalan pintas, yaitu dimulai dari reseptor penerima rangsang, kemudian diteruskan oleh saraf sensori ke pusat saraf terdekat, diterima oleh set saraf penghubung (asosiasi) tanpa diolah di dalam otak langsung dikirim tanggapan ke saraf motor untuk disampaikan ke efektor, yaitu otot atau kelenjar. Jalan pintas ini disebut *lengkung refleks*. Gerak refleks dapat dibedakan atas refleks otak bila saraf penghubung (asosiasi) berada di dalam otak, misalnya, gerak mengedip atau mempersempit pupil bila ada sinar dan refleks sumsum tulang belakang bila set saraf penghubung berada di dalam sumsum tulang belakang misalnya refleks pada lutut. *Autonomic Reflex Arc* dapat dilihat pada gambar 1.

Autonomic Reflex Arc



Gambar 1. *Autonomic Reflex Arc*

(Sumber : <http://www.doctorc.net/Labs/Lab9/Examples/exautarc.htm>)

Aksi *reflex* secara garis besar dibedakan menjadi dua:

1. *Condition reflex*, merupakan gerak tangkas yang dapat dilatih; misalnya pada gerakan-gerakan repetisi saat latihan olahraga.
2. *Uncondition reflex*, merupakan gerakan involunter yang tidak dapat dilatih.

Gerak *reflex* sering kali diuji sebagai bagian dari uji neurologis untuk mengetahui apakah ada kerusakan pada fungsi sistem saraf pusat maupun tepi. Salah satu cara untuk menguji gerak *reflex* seseorang adalah dengan sebuah metode bernama *Whole Body Reaction time* dengan alat bernama *reaction time meter*.

Berdasarkan pengertian dan pendapat mengenai *whole body reaction*, dapat disimpulkan bahwa kecepatan reaksi adalah kemampuan seseorang untuk menjawab rangsangan baik dari penglihatan atau suara dengan waktu yang sesingkat mungkin dan hasil yang maksimal. Dengan mempunyai kecepatan reaksi yang baik maka atlet akan mampu melakukan ketepatan.

D. Tes Pengukuran *Whole Body Reaction time*

Untuk melakukan tes *Whole Body Reaction time* peneliti menggunakan *Whole Body Reaction time test*. Tes ini dilakukan untuk mengetahui tingkat *Whole Body Reaction time* dari seluruh sampel. Jenis tes ini terdapat 2 macam, Visual (Yaitu melakukan tes dengan cara menggunakan indra penglihatan) dan Audio (Yaitu melakukan tes dengan cara menggunakan indra pendengaran). Dalam tes *Whole Body Reaction time* ini peneliti menggunakan cara visual karena bertujuan untuk lebih sesuai dengan kenyataannya dilapangan. Untuk gambar 2 adalah contoh *Whole Body Reaction time test*. Alat ini sudah teruji validitasnya dengan nilai 0.86 oleh perusahaan Takei Co.Op, Tokyo, Japan. Miyatake, N. (2012, hlm. 4) menyatakan bahwa norma *Whole Body Reaction time tes* sebagai berikut :

Tabel 5. Norma Tes *Whole Body Reaction*

Waktu (detik)	Keterangan
0,001 – 0,100	Istimewa
0,101 – 0,200	Bagus Sekali
0,201 – 0,300	Bagus
0,301 – 0,400	Cukup / Sedang
0,401 – 0,500	Kurang
>0,501	Kurang Sekali

(Sumber : repository.upi.edu)



Gambar 2. *Whole Body Reaction time test*

(Sumber : repository.upi.edu)

Berdasarkan cara kerja alat yang sudah ada, nantinya alat ini memberikan dua jenis stimuli yaitu visual dan auditori. Alat ini memiliki tombol-tombol yang harus ditekan oleh testi segera setelah testi mendapatkan stimuli tersebut. Alat ini kemudian akan merekam waktu yang diperlukan testi untuk bereaksi (*reaction time*). Berikut adalah cara kerja dari tes *Whole Body Reaction time* ini:

1. Tujuan untuk mengukur waktu reaksi tangan dan kaki dengan rangsang penglihatan atau pendengaran.
2. Perlengkapan *Reaction Time Meter*, dengan ketelitian sampai dengan per 10.000 detik. Alat ini terdiri dari unit operator, unit penjawab dan 4 lampu perangsang dengan warna berbeda, serta bel.
3. Pelaksanaan
 - a. Lampu perangsang diletakkan di depan testi, terpisah sejauh 3 meter. Tinggi lampu sedikitnya 30° dari pandangan testi.

- b. Unit penjawab (*reaction key*) diletakkan di lantai atau di atas meja di depan testi.
 - c. Testi duduk dengan rileks, jari-jari di-letakkan di atas tombol penjawab.
 - d. Unit operator (regulator) diletakkan di tempat yang tidak menghalangi pandangan testi terhadap lampu perangsang.
 - e. *Display* angka pada unit operator harus menunjukkan angka 00.0000.
 - f. Operator menekan tombol untuk menyalakan lampu perangsang (*display* angka berjalan).
 - g. Testi menekan tombol penjawab sesuai dengan warna lampu yang menyala (*display* angka berhenti).
 - h. Bila yang akan diukur waktu reaksi kaki, kaki diletakkan di atas tombol penjawab.
 - i. Bila akan menggunakan rangsang audio, operator menekan bel dan testi menjawabnya dengan menekan tom-bol penjawab bel.
4. Penilaian
- a. Angka yang tertera pada *display* angka ketika orang coba menjawab rangsang menunjukkan waktu reaksinya.
 - b. Waktu reaksi yang tercepat yang digunakan untuk menilai waktu reaksi testi.

Response time paling rendah jika hanya terdapat satu macam stimulus (simple) dan menjadi semakin lambat pada tiap penambahan jenis stimulus (*choice*). *Response time* paling rendah saat dewasa muda dan perlahan

meningkat seiring dengan usia. *Response time* dapat memendek dengan latihan dan dapat memanjang dalam kondisi kelelahan atau adanya gangguan.

E. *Power Endurance*

Power Endurance adalah kemampuan seseorang untuk melakukan aktifitas secara maksimal dalam waktu yang singkat dan secara terus menerus dalam waktu yang lama dalam tempo sedang sampai cepat tanpa merasa sangat lelah atau sakit. Satuan *power* adalah kg (berat) * meter/detik, sedangkan Kg * meter adalah satuan usaha. Dengan demikian *power* dapat diartikan usaha per detik. Untuk *power endurance* adalah rata-rata *power* dari banyak lompatan yang dilakukan, kemudian di ketahui maksimal *power* dan minimal *power* yang diperoleh. Satuan *Power endurance* yang didapatkan adalah kgm/s dengan ketentuan nilai rata-rata *power*, maksimal *power*, dan minimal *power*.

1. *Power*

Power adalah kemampuan dalam mempergunakan otot untuk menerima beban sewaktu bekerja. *Power* otot dapat diraih dari latihan dengan beban berat dan frekuensi sedikit. Kita dapat melatih kekuatan otot lengan dengan latihan angkat beban, jika beban tersebut hanya dapat diangkat 8-12 kali saja. Contoh latihannya adalah sebagai berikut:

- a. *squat jump*, melatih kekuatan otot tungkai dan otot perut.
- b. *push up*, melatih kekuatan otot lengan.
- c. *sit up*, melatih kekuatan otot perut.
- d. angkat beban, melatih kekuatan otot lengan.
- e. *BACK up*, melatih kekuatan otot perut.

2. Daya tahan otot (*Muscle Endurance*)

Adalah kemampuan otot untuk melakukan suatu gerakan secara terus menerus dalam waktu yang lama dalam tempo sedang sampai cepat tanpa merasa sangat lelah atau sakit. Misalnya daya tahan otot untuk mengangkat beban 15 kg selama beberapa waktu tanpa berhenti atau ketahanan seseorang berlari dalam tempo tertentu selama setengah jam tanpa ada jeda, dalam hal ini menunjukkan ketahanan otot kakinya.

Untuk mendapatkan daya tahan otot tentu memerlukan latihan. Diantara bentuk latihan untuk daya tahan otot adalah:

a. squats

dilakukan dengan cara meletakkan beban di atas pundak di bagian belakang kepala, kemudian lutut dibengkokkan dengan catatan pantat tidak boleh menyentuh tumit.

b. rowing

dilakukan dengan cara membungkukkan badan sampai punggung sejajar membentuk sudut 90 derajat. Kemudian tangan menghadap lurus ke bawah ke arah beban. Angkat beban ke arah dada, untuk menjaga keseimbangan kita bisa menyenderkan dahi di atas meja.

c. triceps stretch

merupakan gerakan mengangkat beban yang ada di belakang kepala. Posisi siku bengkok di samping telinga dengan tangan menghadap kebawah memegang beban yang ada di belakang kepala. Kemudian beban diangkat keatas hingga kedua lengan lurus keatas.

d. wrist roll

dilakukan dengan cara duduk dan meletakkan lengan bawah di atas paha dengan posisi telapak tangan menghadap ke atas. Kemudian gelindingkan beban melalui ujung jari ke telapak tangan dengan cara membengkokkan pergelangan tangan ke arah atas.

Dengan demikian yang dimaksud daya tahan *power* adalah kemampuan atlet mengatasi kelelahan akibat melakukan gerakan dengan cepat dan kuat dilakukan secara berulang-ulang dengan kualitas yang sama selama 10 kali lompatan pada pengukuran *power vertical jump*.

F. Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction*

Dari teori yang ada, sistem pengukuran untuk *power tungkai*, *Power Endurance*, dan *Whole Body Reaction* ini masih bersifat konvensional dan belum terintegrasi. Hal ini di jelaskan pada blok diagram di gambar 3.



Gambar 3. Blok Pengukuran Tinggi Badan

Penjelasan dari gambar 3 adalah atlet melakukan pengukuran menggunakan papan ukur konvensional untuk mengetahui tinggi badan. Tinggi badan nantinya dipergunakan untuk proses pengukuran *Whole Body Reaction* untuk menyesuaikan tinggi reflektor cahaya dengan tinggi orang yang akan diukur.



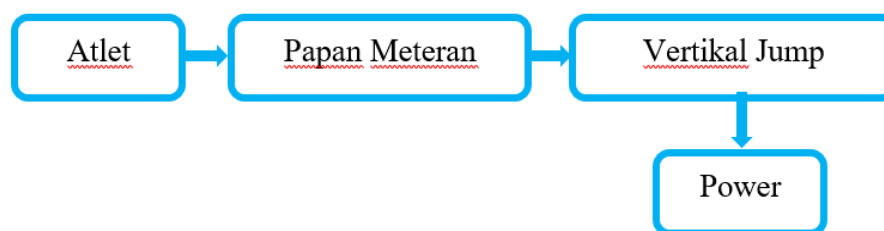
Gambar 4. Blok Pengukur Berat Badan

Penjelasan dari gambar 4 adalah atlet melakukan pengukuran berat badan menggunakan timbangan konvensional. Berat badan ini akan digunakan untuk pengukuran *power* tungkai sesuai dengan rumus yang ada.



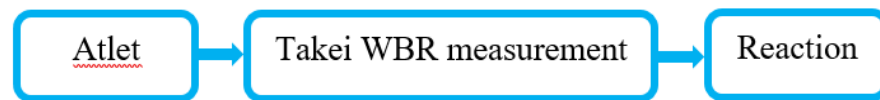
Gambar 5. Blok Pengukur *Power* Tungkai *Horizontal Jump*

Penjelasan dari gambar 5 adalah atlet melakukan proses pengukuran *power* tungkai *standing board jump* menggunakan matras yang diletakkan di lantai dengan di beri meteran sebagai media pengukurnya.



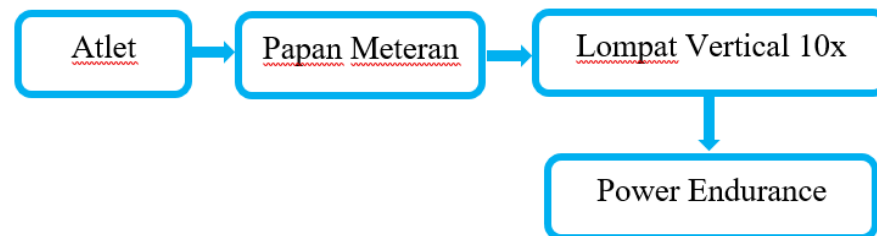
Gambar 6. Blok Pengukur *Power* Tungkai *Vertical Jump*

Penjelasan dari gambar 6 adalah atlet melakukan pengukuran *power* tungkai *vertical jump* menggunakan papan meteran yang di lekatkan pada tembok. Setelah itu atlet melakukan lompatan secara *vertical*, kemudian dari lompatan itu tinggi lompatan yang terbaca akan dikurangi dengan tinggi badan untuk mengetahui *powernya*.



Gambar 7. Blok Pengukur *Whole Body Reaction*

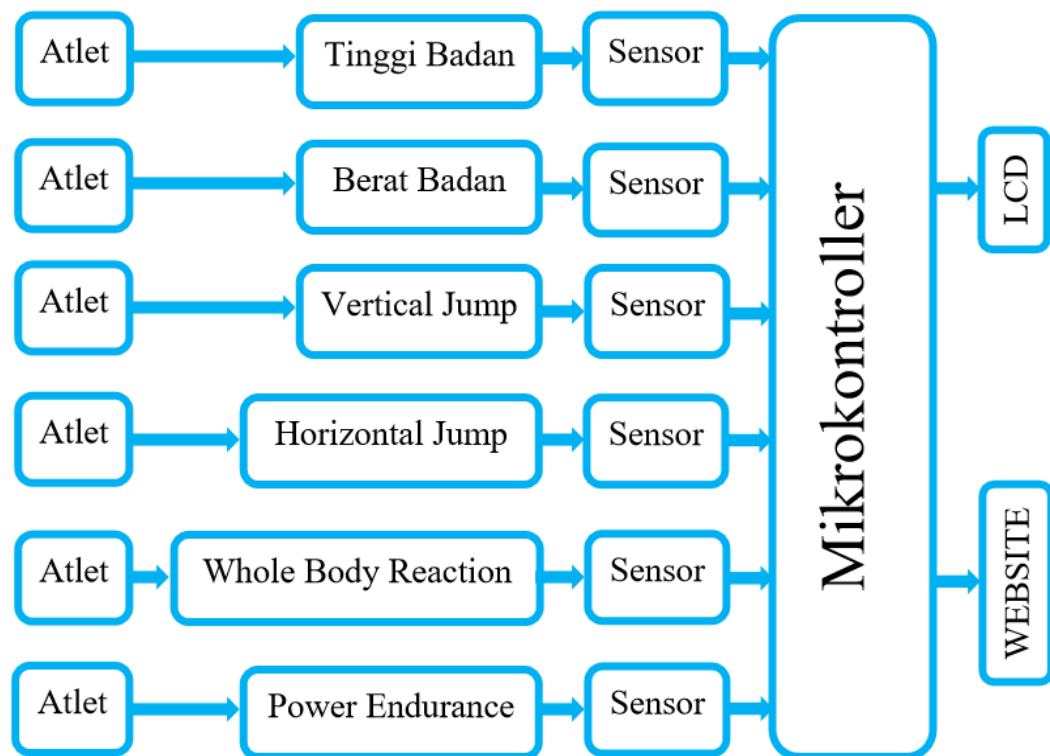
Penjelasan dari gambar 7 adalah atlet melakukan pengukuran menggunakan alat *Takei equipment industrial T.K.K.5408 Whole Body Reaction measuring instrument*. Hasil keluaran dari alat ini masih berupa data kecepatan reaksi dan belum ada analisis dan klasifikasinya.



Gambar 8. Blok Pengukur *Power Endurance*

Penjelasan dari gambar 8 adalah atlet melakukan pengukuran *Power Endurance* menggunakan papan meteran yang di lekatkan pada tembok, lalu atlet mulai lompat secara *vertical* dan sejajar terhadap papan ukur. Lompatan ini dilakukan sebanyak 10 kali untuk mengetahui seberapa maksimal kekuatan otot, seberapa minimal kekuatan otot, dan rata-rata *power* dari 10 kali lompatan yang dilakukan.

Dari blok diagram yang sudah ada diatas, maka akan muncul blok diagram dengan *loop* terbuka, yang akan menggabungkan semua blok diagram pengukuran diatas menjadi sebuah sistem yang terintegrasi. Adapun blok diagram nya adalah seperti gambar dibawah berikut.



Gambar 9. Blok diagram Alat Pengukur *Power* Tungkai *Power Endurance* dan *Whole Body Reaction*

Penjelasan dari gambar 9 adalah semua blok pengukuran yang terpisah tadi, akan digabungkan menjadi satu sistem yang terintegrasi mulai dari pengukuran tinggi badan, berat badan, *power* tungkai *vertical jump*, *power* tungkai *horizontal jump*, *Power Endurance*, dan *Whole Body Reaction*. Semua pengukuran dijadikan satu dan dikontrol oleh mikrokontroler dan *output*-nya akan ditampilkan pada LCD dan *website*.

G. Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah chip. Di dalamnya terkandung sebuah inti *processor*, *memory* (sejumlah kecil RAM, *memory* program, atau keduanya), dan perlengkapan *input output*.

Dengan kata lain, Mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data. Sekedar contoh, bayangkan diri Anda saat mulai belajar membaca dan menulis, ketika Anda sudah bisa melakukan hal itu Anda bisa membaca tulisan apapun baik buku, cerpen, artikel dan sebagainya, dan Andapun bisa pula menulis hal-hal sebaliknya. Begitulah pula jika anda sudah mahir membaca dan menulis data maka anda dapat membuat program untuk sistem pengaturan otomatis menggunakan mikrokontroler sesuai keinginan anda. Mikrokontroler merupakan komputer didalam chip yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut “pengendali kecil” dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti IC TTL dan CMOS dapat direduksi/diperkecil dan akhirnya terpusat serta dikendalikan oleh mikrokontroler ini.

Mikrokontroler digunakan dalam produk dan alat yang dikendalikan secara otomatis, seperti sistem kontrol mesin, *remote control*, mesin kantor, peralatan rumah tangga, alat berat, dan mainan. Dengan mengurangi ukuran, biaya, dan konsumsi tenaga dibandingkan dengan mendesain menggunakan mikroprosesor memori, dan alat *input output* yang terpisah, kehadiran mikrokontroler membuat kontrol elektrik untuk berbagai proses menjadi lebih ekonomis. Dengan penggunaan mikrokontroler ini maka :

1. Sistem elektronik akan menjadi lebih ringkas

2. Rancang bangun sistem elektronik akan lebih cepat karena sebagian besar dari sistem adalah perangkat lunak yang mudah dimodifikasi
 3. Pencarian gangguan lebih mudah ditelusuri karena sistemnya yang kompak
- Agar sebuah mikrokontroler dapat berfungsi, maka mikrokontroler tersebut memerlukan komponen eksternal yang kemudian disebut dengan sistem minimum. Untuk membuat sistem minimal paling tidak dibutuhkan sistem *clock* dan *Reset*, walaupun pada beberapa mikrokontroler sudah menyediakan sistem *clock internal*, sehingga tanpa rangkaian eksternal pun mikrokontroler sudah beroperasi.

H. Mikrokontroler Arduino

1. Arduino Mega 2560

Arduino Mega 2560 adalah papan mikrokontroler berbasis Atmega 2560. Arduino Mega 2560 seperti gambar 2 memiliki 54 pin digital *input / output*, dimana 15 pin dapat digunakan sebagai *output* PWM, 16 pin sebagai *input analog*, dan 4 pin sebagai UART (*port serial hardware*), 16 MHz *crystal oscillator*, koneksi USB, *jack power*, *header ICSP*, dan tombol *Reset*. Ini semua yang diperlukan untuk mendukung mikrokontroler. Cukup dengan menghubungkannya ke komputer melalui kabel USB atau *power* dihubungkan dengan adaptor AC – DC atau baterai untuk mulai mengaktifkannya. Arduino Mega 2560 kompatibel dengan sebagian besar *shield* yang dirancang untuk Arduino Duemilanove atau Arduino Diecimila. Arduino Mega 2560 adalah versi terbaru yang menggantikan versi Arduino Mega.

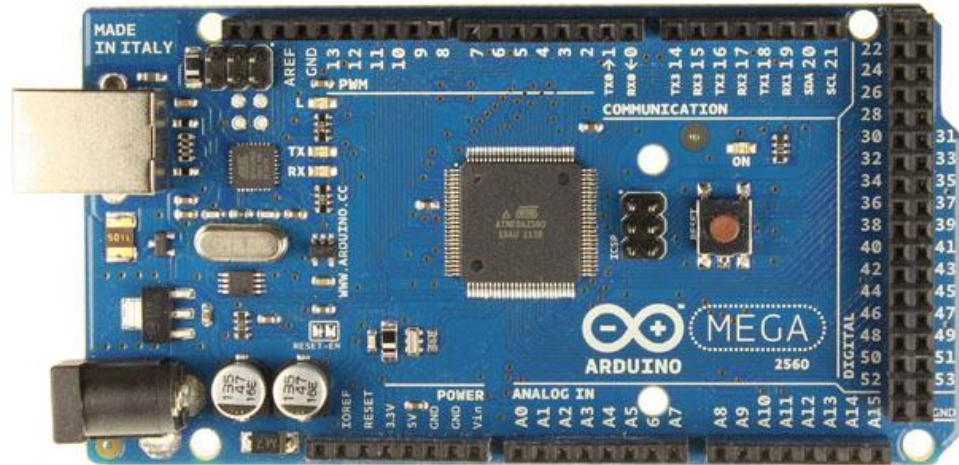
Arduino Mega 2560 berbeda dari papan sebelumnya, Karena versi terbaru sudah tidak menggunakan *chip driver* FTDI USB-to-Serial. Tapi, menggunakan *chip* Atmega 16U2 (Atmega 8U2 pada papan Revisi dan Revisi 2) yang deprogram sebagai *converter* USB-to-Serial. Arduino Mega 2560 Revisi 2 memiliki resistor penarik jalur HWB 8U2 ke *Ground*, sehingga lebih mudah untuk dimasukkan dalam mode DFU. Untuk penampakan hardware dari atmega 2560 dapat dilihat pada gambar 10 dan pemetaan pin atmega 2560 dapat dilihat pada gambar 11. Fitur-fitur yang terdapat pada Arduino Mega 2560 antara lain adalah sebagai berikut:

a. Pinout

Ditambahkan pin SDA dan pin SCL yang dekat dengan pin AREF dan dua pin baru lainnya ditempatkan dekat dengan pin *RESET*, IOREF memungkinkan *shield* untuk beradaptasi dengan tegangan yang tersedia pada papan. Di masa depan, *shield* akan kompatibel baik dengan papan yang menggunakan AVR yang beroperasi dengan 5 Volt dan dengan Arduino Due yang beroperasi dengan tegangan 3.3 Volt. Dan ada dua pin yang tidak terhubung, yang disediakan untuk tujuan masa depan.

b. Sirkuit *RESET*.

c. *Chip* ATmega16U2 menggantikan *chip* Atmega 8U2.



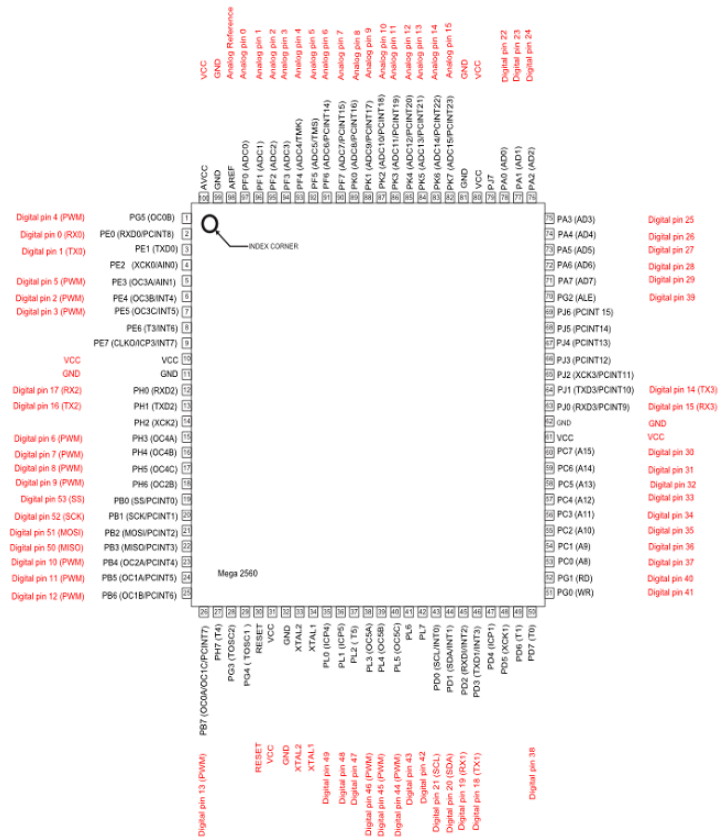
Gambar 10. Board Arduino Mega 2560 Tampak Depan

(Sumber : <https://www.arduino.cc/>)

Tabel 6. Spesifikasi dari Arduino Mega 2560

Mikrokontroler	ATmega 2560
Tegangan Operasi	5V
<i>Inputvoltage</i> (disarankan)	7-12V
<i>InputVoltage</i> (limit)	6-20V
Jumlah pin I/O digital	54 (15 pin digunakan sebagai <i>output</i> PWM)
Jumlah pin <i>input</i> analog	16
Arus DC tiap pin I/O	40 mA
Arus DC untuk pin 3.3V	50 mA
<i>Flash Memory</i>	256 KB (8 KB digunakan untuk <i>bootloader</i>)
SRAM	8 KB
EEPROM	4 KB
<i>Clock Speed</i>	16 MHz

(Sumber : <https://www.arduino.cc/>)



Gambar 11. Pemetaan Pin ATmega 2560

(Sumber : <https://www.arduino.cc/>)

d. *Catu Daya (Power supply)*

Arduino Mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Sumber daya eksternal (non-USB) dapat berasal baik dari adaptor AC-DC atau baterai. Adaptor dapat dihubungkan dengan mencolokkan steker 2,1 mm yang bagian tengahnya terminal positif ke ke *jack* sumber tegangan pada papan. Jika tegangan berasal dari baterai dapat langsung dihubungkan melalui header pin Gnd dan pin Vin dari konektor *POWER*.

Papan Arduino ATmega 2560 dapat beroperasi dengan pasokan daya eksternal 6 Volt sampai 20 volt. Jika diberi tegangan kurang dari 7 Volt, maka, pin 5 Volt mungkin akan menghasilkan tegangan kurang dari 5 Volt dan ini akan membuat papan menjadi tidak stabil. Jika sumber tegangan menggunakan lebih dari 12 Volt, regulator tegangan akan mengalami panas berlebihan dan bisa merusak papan. Rentang sumber tegangan yang dianjurkan adalah 7 Volt sampai 12 Volt. Pin tegangan yang tersedia pada papan Arduino adalah sebagai berikut:

- 1) VIN: Adalah *input* tegangan untuk papan Arduino ketika menggunakan sumber daya eksternal (sebagai cadangan dari tegangan 5 Volt dari koneksi USB atau sumber daya *ter-regulator* lainnya). Anda dapat memberikan tegangan melalui pin ini, atau jika memasok tegangan untuk papan melalui jack *power*, kita bisa mengakses/mengambil tegangan melalui pin ini.
- 2) 5V: Sebuah pin yang mengeluarkan tegangan *ter-regulator* 5 Volt, dari pin ini tegangan sudah diatur (*ter-regulator*) dari *regulator* yang tersedia (*built-in*) pada papan. Arduino dapat diaktifkan dengan sumber daya baik berasal dari *jack power* DC (7-12 Volt), konektor USB (5 Volt), atau pin VIN pada *board* (7-12 Volt). Memberikan tegangan melalui pin 5V atau 3,3V secara langsung tanpa melewati *regulator* dapat merusak papan Arduino.
- 3) 3V3: Sebuah pin yang menghasilkan tegangan 3,3 Volt. Tegangan ini dihasilkan oleh *regulator* yang terdapat pada papan (*on-board*). Arus maksimum yang dihasilkan adalah 50 mA.

4) GND: Pin *Ground* atau *Massa*.

5) IOREF: Pin ini pada papan Arduino berfungsi untuk memberikan referensi tegangan yang beroperasi pada mikrokontroler. Sebuah perisai (*shield*) dikonfigurasi dengan benar untuk dapat membaca pin tegangan IOREF dan memilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan (*voltage translator*) pada *output* untuk bekerja pada tegangan 5 Volt atau 3,3 Volt.

e. Memori

Arduino ATmega 2560 memiliki 256 KB *flash memory* untuk menyimpan kode (8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

f. *Input* dan *Output*

Masing-masing dari 54 digital pin pada Arduino Mega dapat digunakan sebagai *input* atau *output*, menggunakan fungsi *pinMode()*, *digitalwrite()*, dan *digitalread()*. Arduino Mega beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor *pull-up* internal (yang terputus secara *default*) sebesar 20 – 50 kilo ohms. Selain itu, beberapa pin memiliki fungsi khusus, antara lain:

1) Serial yang digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL.

Tabel 7. Tabel Pin Serial RX dan TX

Nomor Pin	Nama Pin	Peta Nama Pin
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)
3	PE1 (TXD0)	Digital pin 1 (TX0)
12	PH0 (RXD2)	Digital pin 17 (RX2)
13	PH1 (TXD2)	Digital pin 16 (TX2)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)
63	PJ0 (RXD3/PCINT9)	Digital pin 15 (RX3)
64	PJ1 (TXD3/PCINT10)	Digital pin 14 (TX3)

(Sumber : <https://www.arduino.cc/>)

- 2) Eksternal Interupsi: Pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun , atau perubah nilai.

Tabel 8. Tabel Pin Eksternal Interupsi

Nomor Pin	Nama Pin	Peta Nama Pin
6	PE4 (OC3B/INT4)	Digital pin 2 (PWM)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)
43	PD0 (SCL/INT0)	Digital pin 21 (SCL)
44	PD1 (SDA/INT1)	Digital pin 20 (SDA)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)

(Sumber : <https://www.arduino.cc/>)

- 3) SPI: Pin ini mendukung komunikasi SPI menggunakan SPI library. Pin SPI juga terhubung dengan header ICSP, yang secara fisik kompatibel dengan Arduino Uno, Arduino Duemilanove dan Arduino Diecimila.

Tabel 9. Tabel Pin *SPI*

Nomor Pin	Nama Pin	Peta Nama Pin
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)
21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)

22	PB3 (MISO/PCINT3)	Digital pin 50 (MISO)
----	-------------------	-----------------------

(Sumber : <https://www.arduino.cc/>)

- 4) LED: Pin 13. Tersedia secara *built-in* pada papan Arduino ATmega LED terhubung ke pin digital 13. Ketika pin diset bernilai *HIGH*, maka LED menyala (ON), dan ketika pin diset bernilai *LOW*, maka LED padam (OFF).
- 5) TWI: Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan *Wire library*. Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin TWI pada Arduino Duemilanove atau Arduino Diecimila.

Arduino Mega 2560 memiliki 16 pin sebagai analog *input*, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara *default* pin ini dapat diukur/diatur dari mulai *Ground* sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin AREF dan fungsi *Analog Reference()*. Ada beberapa pin lainnya yang tersedia, antara lain:

- 7) AREF: Referensi tegangan untuk *input* digunakan dengan *Analog Reference()*;
- 8) *RESET*: Jalur *LOW* ini digunakan untuk me-*Reset* (menghidupkan ulang) mikrokontroler. Jalur ini bisaanya digunakan untuk menambahkan tombol *Reset* pada *shield* yang menghalangi papan utama Arduino.

g. Komunikasi

Arduino Mega 2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan Arduino lain, atau dengan mikrokontroler lainnya. Arduino ATmega 328 menyediakan 4 *hardware* komunikasi serial UART TTL (5 Volt). Sebuah *chip* ATmega 16U2 (ATmega 8U2 pada papan Revisi 1 dan Revisi 2) yang terdapat pada papan digunakan sebagai media komunikasi serial melalui USB dan muncul sebagai *COM Port Virtual* (pada Device komputer) untuk berkomunikasi dengan perangkat lunak pada komputer, untuk sistem operasi Windows masih tetap memerlukan *file* inf, tetapi untuk sistem operasi OS X dan Linux akan mengenali papan sebagai *port* COM secara otomatis. Perangkat lunak Arduino termasuk didalamnya serial monitor memungkinkan data tekstual sederhana dikirim ke dan dari papan Arduino. LED RX dan TX yang tersedia pada papan akan berkedip ketika data sedang dikirim atau diterima melalui *chip* USB-to-serial yang terhubung melalui USB komputer (tetapi tidak untuk komunikasi serial seperti pada pin 0 dan 1). Sebuah *Software Serial library* memungkinkan untuk komunikasi serial pada salah satu pin digital Mega 2560. ATmega 2560 juga mendukung komunikasi TWI dan SPI. Perangkat lunak Arduino termasuk *Wirelibrary* digunakan untuk menyederhanakan penggunaan bus TWI. Untuk komunikasi SPI, menggunakan *SPI library*.

h. *Reset* Otomatis

Daripada menekan tombol *Reset* sebelum *upload*, Arduino Mega 2560 didesain dengan cara yang memungkinkan Anda untuk me-

Reset melalui perangkat lunak yang berjalan pada komputer yang terhubung. Salah satu jalur kontrol *hardware* (DTR) mengalir dari ATmega 8U2 / 16U2 dan terhubung ke jalur *Reset* dari ATmega 2560 melalui kapasitor 100 nanofarad. Bila jalur ini di-set rendah / *low*, jalur *Resetdrop* cukup lama untuk me-*Reset* chip. Perangkat lunak Arduino menggunakan kemampuan ini untuk memungkinkan Anda meng-*upload* kode dengan hanya menekan tombol *upload* pada perangkat lunak Arduino. Ini berarti bahwa *bootloader* memiliki rentang waktu yang lebih pendek, seperti menurunkan DTR dapat terkoordinasi (berjalan beriringan) dengan dimulainya *upload*.

Pengaturan ini juga memiliki implikasi lain. Ketika Mega 2560 terhubung dengan komputer yang menggunakan sistem operasi Mac OS X atau Linux, papan Arduino akan di-*Reset* setiap kali dihubungkan dengan *software* komputer (melalui USB). Dan setengah detik kemudian atau lebih, *bootloader* berjalan pada papan Mega 2560. Proses *Reset* melalui program ini digunakan untuk mengabaikan data yang cacat (yaitu apapun selain meng-*upload* kode baru), ia akan memotong dan membuang beberapa byte pertama dari data yang dikirim ke papan setelah sambungan dibuka. Jika sebuah sketsa dijalankan pada papan untuk menerima satu kali konfigurasi atau menerima data lain ketika pertama kali dijalankan, pastikan bahwa perangkat lunak diberikan waktu untuk berkomunikasi dengan menunggu satu detik setelah terkoneksi dan sebelum mengirim data.

Mega 2560 memiliki trek jalur yang dapat dipotong untuk menonaktifkan fungsi *auto-Reset*. Pad di kedua sisi jalur dapat dihubungkan dengan disolder untuk mengaktifkan kembali fungsi *auto-Reset*. Pad berlabel “*RESET-EN*”. Anda juga dapat menonaktifkan *auto-Reset* dengan menghubungkan resistor 110 ohm dari 5V ke jalur *Reset*.

i. Perlindungan Beban Berlebih pada USB

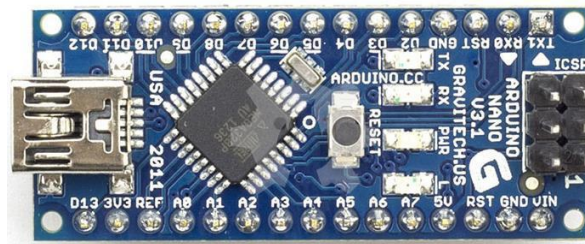
Arduino Mega 2560 memiliki *polyfuse Reset* yang melindungi *port* USB komputer Anda dari hubungan singkat dan arus lebih. Meskipun pada dasarnya komputer telah memiliki perlindungan *internal* pada *port* USB mereka sendiri, sekring memberikan lapisan perlindungan tambahan. Jika arus lebih dari 500 mA dihubungkan ke *port* USB, sekring secara otomatis akan memutuskan sambungan sampai hubungan singkat atau *overload* dihapus/dibuang.

2. Arduino NANO

Arduino merupakan sebuah *platform* dari *physical computing* yang bersifat *open source*. Pertama-tama perlu dipahami bahwa kata “*platform*” di sini adalah sebuah pilihan kata yang tepat. Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi ia adalah kombinasi dari *hardware*, bahasa pemrograman dan *Integrated Development Environment* (IDE) yang canggih. IDE adalah sebuah *software* yang sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan meng-upload ke dalam memory mikrokontroler

Arduino Nano adalah salah satu papan pengembangan *microcontroller* yang berukuran kecil, lengkap dan mendukung penggunaan

breadboard. Arduino Nano diciptakan dengan basis *microcontroller* ATmega328 (untuk Arduino Nano versi 3.x) atau ATmega 168 (untuk Arduino versi 2.x). Arduino Nano kurang lebih memiliki fungsi yang sama dengan Arduino Duemilanove, tetapi dalam paket yang berbeda. Arduino Nano tidak menyertakan colokan DC berjenis Barrel Jack, dan dihubungkan ke komputer menggunakan port USB Mini-B. Arduino Nano dirancang dan diproduksi oleh perusahaan Gravitech. Penampakan hardware Arduino Nano dapat dilihat pada gambar 12 berikut.



Gambar 12. Bagian Depan Arduino
Nano

(Sumber : <https://www.arduino.cc/>)

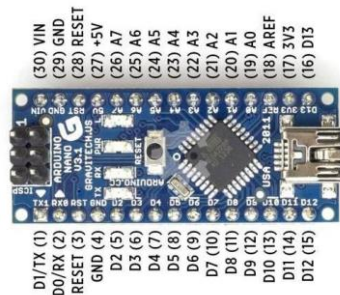
a. Spesifikasi Arduino Nano

Berikut ini adalah Spesifikasi yang dimiliki oleh Arduino Nano:

- 1) Mikrokontroler Atmel ATmega168 atau ATmega328.
- 2) 5 V *Operation Voltage*.
- 3) 7-12V *Input Voltage* (disarankan).
- 4) 6-20V *Input Voltage* (limit).
- 5) Pin Digital I/O14 (6 pin digunakan sebagai *output* PWM).
- 6) 8 Pin *Input* Analog.

- 7) 40 mA Arus DC per pin I/O.
 - 8) *Flash Memory* 16KB (ATmega168) atau 32KB (ATmega328) 2KB digunakan oleh *Bootloader*.
 - 9) 1 *Kbyte* SRAM (ATmega168) atau 2 *Kbyte* (ATmega328).
 - 10) 512 *Byte* EEPROM (ATmega168) atau 1 *Kbyte* (ATmega328).
 - 11) 16 MHz *Clock Speed*.
 - 12) Ukuran 1.85cm x 4.3cm.
- b. Konfigurasi Pin Arduino Nano

Arduino nano memiliki 30 Pin. Berikut konfigurasi pin Arduino Nano dapat terlihat pada gambar 13..



Gambar 13. Konfigurasi Pin Layout Arduino Nano
(Sumber : <https://www.arduino.cc/>)

Tabel 10. Konfigurasi Pin Arduino Nano

Nomor Pin Arduino NANO	Nama Pin Arduino NANO
1	Digital Pin 1 (TX)
2	Digital Pin 0 (RX)
3 & 28	<i>Reset</i>
4 & 29	GND
5	Digital Pin 2
6	Digital Pin 3 (PWM)
7	Digital Pin 4
8	Digital Pin 5 (PWM)
9	Digital Pin 6 (PWM)
10	Digital Pin 7
11	Digital Pin 8
12	Digital Pin 9 (PWM)

13	Digital Pin 10 (PWM-SS)
14	Digital Pin 11 (PWM-MOSI)
15	Digital Pin 12 (MISO)
16	Digital Pin 13 (SCK)
17	3V3
18	AREF
19	Analog <i>Input</i> 0
20	Analog <i>Input</i> 1
21	Analog <i>Input</i> 2
22	Analog <i>Input</i> 3
23	Analog <i>Input</i> 4
24	Analog <i>Input</i> 5
25	Analog <i>Input</i> 6
26	Analog <i>Input</i> 7
27	+5V
28	<i>RESET</i>
29	GND
30	VIN

(Sumber : <https://www.arduino.cc/>)

Dari Tabel tersebut dapat dijelaskan secara fungsional konfigurasi pin Arduino Nano sebagai berikut:

- 1) VCC merupakan pin yang berfungsi sebagai pin masukan catu daya digital.
- 2) GND merupakan pin ground untuk catu daya digital.
- 3) AREF merupakan Referensi tegangan untuk *input* analog. Digunakan dengan fungsi *analogReference()*.
- 4) *RESET* merupakan Jalur *LOW* ini digunakan untuk me-Reset (menghidupkan ulang) mikrokontroler. Bisaanya digunakan untuk menambahkan tombol *Reset* pada shield yang menghalangi papan utama Arduino
- 5) Serial RX (0) merupakan pin yang berfungsi sebagai penerima TTL data serial.

- 6) Serial TX (1) merupakan pin yang berfungsi sebagai pengirim TT data serial.
- 7) *External Interrupt* (Interupsi Eksternal) merupakan pin yang dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubahan nilai.
- 8) *Output PWM 8-Bit* merupakan pin yang berfungsi untuk *analogWrite()*.
- 9) SPI merupakan pin yang berfungsi sebagai pendukung komunikasi.
- 10) LED merupakan pin yang berfungsi sebagai pin yang diset bernilai *HIGH*, maka LED akan menyala, ketika pin diset bernilai *LOW* maka LED padam. LED Tersedia secara built-in pada papan Arduino Nano.
- 11) *Input Analog (A0-A7)* merupakan pin yang berfungsi sebagai pin yang dapat diukur/diatur dari mulai Ground sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan fungsi *analogReference()*.

c. Sumber Daya Arduino Nano

Arduino Nano dapat diaktifkan melalui koneksi USB Mini-B, atau melalui catu daya eksternal dengan tegangan belum teregulasi antara 6-20 Volt yang dihubungkan melalui pin 30 atau pin VIN, atau melalui catu daya eksternal dengan tegangan teregulasi 5 volt melalui pin 27 atau pin 5V. Sumber daya akan secara otomatis dipilih dari sumber tegangan yang lebih tinggi. Chip FTDI FT232L pada Arduino Nano akan aktif apabila memperoleh daya melalui USB, ketika Arduino Nano

diberikan daya dari luar (Non-USB) maka Chip FTDI tidak aktif dan pin 3.3V pun tidak tersedia (tidak mengeluarkan tegangan), sedangkan LED TX dan RX pun berkedip apabila pin digital 0 dan 1 berada pada posisi *HIGH*.

d. Memori

Chip ATmega328 pada Arduino Uno R3 memiliki memori 32 KB, dengan 0.5 KB dari memori tersebut telah digunakan untuk bootloader. Jumlah SRAM 2 KB, dan EEPROM 1 KB, yang dapat di baca-tulis dengan menggunakan EEPROM *library* saat melakukan pemrograman.

e. *Input dan Output (I/O)*

Seperti yang telah disebutkan sebelumnya, Arduino Nano memiliki 14 buah digital pin yang dapat digunakan sebagai *input* atau *output*, sengan menggunakan fungsi *pinMode()*, *digitalwrite()*, dan *digital(Read)*. Pin-pin tersebut ekerja pada tegangan 5V, dan setiap pin dapat menyediakan atau menerima arus 20mA, dan memiliki tahanan pull-up sekitar 20-50k ohm (secara default dalam posisi *disconnect*). Nilai *maximum* adalah 40mA, yang sebisa mungkin dihindari untuk menghindari kerusakan chip mikrokontroler.

Beberapa pin memiliki fungsi khusus :

- 1) Serial, terdiri dari 2 pin : pin 0 (RX) dan pin 1 (TX) yang digunakan untuk menerima (RX) dan mengirim (TX) data serial.
- 2) *External Interrupt*, yaitu pin 2 dan pin 3. Kedua pin tersebut dapat digunakan untuk mengaktifkan *interrupt*. Gunakan fungsi *attachInterrupt()*

- 3) PWM: Pin 3, 5, 6, 9, 10, dan 11 menyediakan *output* PWM 8-bit dengan menggunakan fungsi *analogWrite()*
- 4) SPI : Pin 10 (SS), 11 (MOSI), 12 (MISO), dan 13 (SCK) mendukung komunikasi SPI dengan menggunakan *SPI Library*
- 5) LED : Pin 13. Pada pin 13 terhubung built-in led yang dikendalikan oleh digital pin no 13.

Arduino Nano memiliki 8 buah *input* analog, yang diberi tanda dengan A0 hingga A7. Masing-masing pin analog tersebut memiliki resolusi 1024 bits (jadi bisa memiliki 1024 nilai). Secara default, pin-pin tersebut diukur dari ground ke 5V, namun bisa juga menggunakan pin REF dengan menggunakan fungsi *analogReference()*.

Pin Analog A6 dan A7 tidak bisa dijadikan sebagai pin digital, hanya sebagai analog. Beberapa pin lainnya pada board ini adalah :

- 1) I2C : Pin A4 (SDA) dan A5 (SCL). Pin ini mendukung komunikasi I2C (TWI) dengan menggunakan *Wire Library*.
- 2) AREF. Sebagai referensi tegangan untuk *input* analog.
- 3) *Reset*. Hubungkan ke *LOW* untuk melakukan *Reset* terhadap mikrokontroler. Bisaanya digunakan untuk dihubungkan dengan switch yang dijadikan tombol *Reset*.

f. Komunikasi

Arduino Nano 3.0 memiliki beberapa fasilitas untuk berkomunikasi dengan komputer, berkomunikasi dengan Arduino lainnya, atau dengan mikrokontroler lain nya. Chip Atmega328 menyediakan komunikasi serial UART TTL (5V) yang tersedia di pin 0 (RX) dan pin 1 (TX).

Sebuah *chip* FTDI yang terdapat pada board berfungsi menterjemahkan bentuk komunikasi ini melalui USB dan akan tampil sebagai Virtual Port di komputer.

Pada *Arduino Software (IDE)* terdapat monitor serial yang memudahkan data textual untuk dikirim menuju Arduino atau keluar dari Arduino. Lampu led TX dan RX akan menyala berkedip-kedip ketika ada data yang ditransmisikan melalui chip FTDI USB to Serial via kabel USB ke komputer. Untuk menggunakan komunikasi serial dari digital pin, gunakan *SoftwareSerial library*.

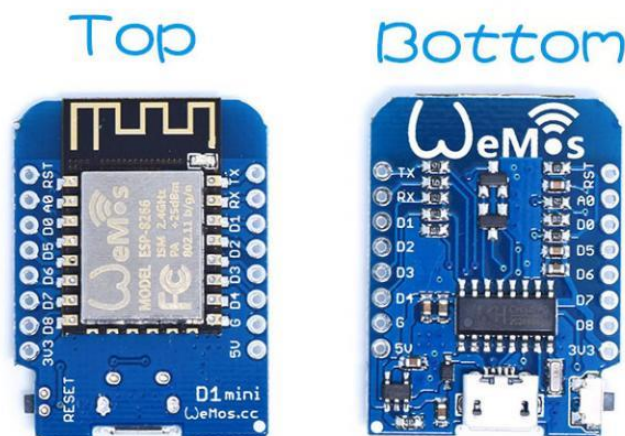
Chip ATmega328 juga mendukung komunikasi I2C (TWI) dan SPI. Di dalam *Arduino Software (IDE)* sudah termasuk *Wire Library* untuk memudahkan anda menggunakan bus I2C. Untuk menggunakan komunikasi SPI, gunakan *SPI library*.

g. *Reset Otomatis (software)*

Bisaanya, ketika anda melakukan pemrograman mikrokontroler, anda harus menekan tombol *Reset* sesaat sebelum melakukan upload program. Pada *Arduino Uno*, hal ini tidak lagi merepotkan anda. *Arduino Uno* telah dilengkapi dengan auto *Reset* yang dikendalikan oleh *software* pada komputer yang terkoneksi. Salah satu jalur *flow control* (DTR) dari ATmega16U pada *Arduino Uno R3* terhubung dengan jalur *Reset* pada ATmega328 melalui sebuah kapasitor 100nF. Ketika jalur tersebut diberi nilai *LOW*, mikrokontroler akan di *Reset*. Dengan demikian proses upload akan jauh lebih mudah dan anda tidak harus menekan tombol *Reset* pada saat yang tepat seperti bisaanya.

I. Mikrokontroler Wemos

Mikrokontroler Wemos adalah sebuah mikrokontroler pengembangan berbasis modul mikrokontroler ESP 8266. Mikrokontroler Wemos dibuat sebagai solusi dari mahalnya sebuah sistem *wireless* berbasis mikrokontroler lainnya. Dengan menggunakan mikrokontroler Wemos, biaya yang dikeluarkan untuk membangun sistem WiFi berbasis mikrokontroler sangat murah, hanya sepersepuluh dari biaya yang dikeluarkan apabila membangun sistem WiFi dengan menggunakan mikrokontroler Arduino Uno dan WiFi Shield. Penampakan hardware Wemos D1 dapat dilihat pada gambar 14.



Gambar 14. Mikrokontroler Wemos D1 mini

(Sumber : <https://www.wemos.cc/>)

Yang berbeda pada Mikrokontroler ini yaitu kemampuannya untuk menyediakan fasilitas konektivitas WiFi dengan mudah serta *memory* yang digunakan sangat besar yaitu 4 MB[2].

1. Mikrokontroler *Chipset* pada Mikrokontroler Wemos

Pada *Microcontroller* wemos memiliki 2 buah chipset yang digunakan sebagai otak kerja *platform* tersebut. Berapa chipset pada mikrokontroler ini adalah:

a. Chipset ESP8266

ESP8266 adalah sebuah *chip* mikrokontroler yang memiliki fitur WiFi yang mendukung stack TCP / IP. Diproduksi oleh produsen cina yang berbasis di shanghai, Espresif. Pada Agustus 2014 AI-Thinker membuat modul ESP-01 dengan menggunakan lisensi oleh Espresif. Modul kecil ini menggunakan mikrokontroler untuk terhubung dengan jaringan WiFi dan membuat koneksi TCP / IP hanya dengan menggunakan *command* yang sederhana seperti Hayes-gaya. Harga yang sangat rendah dan sangat sedikit komponen eksternal pada modul ini mengakibatkan sangat murah harga sebuah chip ini. Dengan clock 80 Mhz chip ini dibekali dengan 4MB Eksternal RAM, mendukung format IEEE 802.11 b/g/n sehingga tidak menyebabkan interference bagi yang lain. Mendukung enkripsi WEP, WPA sehingga menjadikan chipset ini sangat aman digunakan. Chipset ini memiliki 16 GPIO pin yang bekerja pada tegangan 3.3Volt, 1 pin ADC dengan resolusi 10 bit.

b. *Chipset* CH340

CH340 adalah sebuah *chipset* yang mengubah usb menjadi serial *interface*. Sebagai contohnya adalah aplikasi USB *converter* to IrDA atau aplikasi USB *converter* to Printer. Dalam mode *serial interface*, CH340 mengirimkan sinyal penghubung yang umum digunakan pada MODEM. CH340 digunakan untuk memperbesar *asynchronous serial interface*

komputer atau mengubah perangkat serial *interface* umum untuk berhubungan dengan bus USB secara langsung.

2. Spesifikasi Wemos D1 mini
 - a. 11 digital *input/output* pins
 - b. Interrupt/pwm/I2C/one-wire
 - c. 1 analog *input*(3.2V max *input*)
 - d. 16M bytes(128M bit) Flash
 - e. External antenna connector
 - f. Built-in ceramic antenna
 - g. New CP2104 USB-TO-UART IC
 - h. Same size as D1 mini, but more light

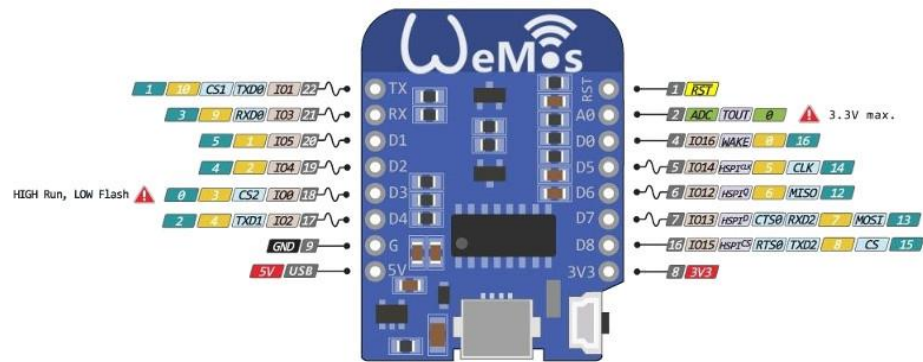
Tabel 11. Spesifikasi Wemos D1

Nama	Spesifikasi
Mikrokontroler	ESP-8266EX
<i>Operating Voltage</i>	3.3V
Digital I/O Pins	11
Analog <i>Input</i> Pins	1(Max <i>input</i> : 3.3V)
Clock Speed	80MHz/160MHz
Flash	16M bytes
Length	34.2mm
Width	25.6mm
Weight	2.5g

(Sumber : <https://www.wemos.cc/>)

3. Konfigurasi Pin D1 mini

Wemos D1 mini memiliki 16 Pin. Gambar 15 berikut menunjukkan konfigurasi pin Wemos D1 mini.



Gambar 15. Konfigurasi Pin Wemos D1 Mini

(Sumber : <https://www.wemos.cc/>)

Tabel 12. Konfigurasi Pin Wemos D1

Pin Wemos D1	Fungsi	ESP-8266 Pin
TX	TXD	TXD
RX	RXD	RXD
A0	Analog <i>Input</i> , max 3.3V <i>input</i>	A0
D0	IO	GPIO 16
D1	IO, SCL	GPIO 5
D2	IO, SDA	GPIO 4
D3	IO, 10k Pull-up	GPIO 0
D4	IO, 10k Pull-up, BUILTIN_LED	GPIO 2
D5	IO, SCK	GPIO 14
D6	IO, MISO	GPIO 12
D7	IO, MOSI	GPIO 13
D8	IO, 10k Pull-down, SS	GPIO 15
G	Ground	GND
5V	5V	
3V3	3.3V	3.3V
RST	<i>Reset</i>	RST

(Sumber : <https://www.wemos.cc/>)

J. Bahasa Pemrograman C

Dikembangkan pertama kali oleh *Dennis Ritchie* dan *Ken Thomson* pada tahun 1972, Bahasa C merupakan salah satu bahasa pemrograman yang paling

populer untuk mengembangkan program-program aplikasi yang berjalan pada sistem mikroprosesor. Karena kepopulerannya dan banyak digunakan, vendor vendor perangkat lunak kemudian mengembangkan *compiler C* sehingga menjadi beberapa varian, macam-macam varian tersebut yaitu: Turbo C, Borland C, Microsoft C, *Power C*, Zortech C, dan lain sebagainya. Untuk menjaga probabilitas, *compiler-compiler C* tersebut menerapkan ANSI C (*American National Standards Institute*) sebagai standar bakunya. Perbedaan antara *compiler-compiler* tersebut umumnya hanya pada pengembangan fungsi-fungsi pustaka serta fasilitas IDE (*Intergrated Development Environment*).

Apabila dibandingkan dengan bahasa tingkat tinggi lain, bahasa C relatif merupakan bahasa pemrograman yang sangat fleksibel dan tidak terlalu terikat dengan berbagai aturan yang sifatnya kaku. Salah satu hal yang membatasi penggunaan bahasa C dalam sebuah aplikasi adalah semata-mata kemampuan imajinasi pemrogramnya saja. Sebagai contoh, dalam bahasa program C kita dapat saja secara bebas menjumlahkan karakter huruf (misal 'D') dengan bilangan bulat (misal '4'), dimana hal ini tidak mungkin dapat dilakukan dengan menggunakan bahasa tingkat tinggi lainnya. Karena sifatnya ini, maka bahasa C seringkali dikategorikan sebagai bahasa tingkat menengah (*mid level language*).

Dalam kaitannya dengan pemrograman mikrokontroler, bahasa C sekarang mulai menggeser bahasa yang lebih dulu digunakan untuk pemrograman mikrokontroler yaitu bahasa *assembler*. Penggunaan bahasa C akan sangat efisien terutama untuk program mikrokontroler yang berukuran

relatif besar. Dibandingkan dengan bahasa *assembler*, penggunaan bahasa C dalam pemrograman memiliki beberapa kelebihan berikut: Mempercepat waktu pengembangan, bersifat modular dan terstruktur, namun kelemahannya adalah kode program hasil kompilasi akan relatif lebih besar dan sebagai konsekuensinya hal ini terkadang akan mengurangi kecepatan eksekusi.

Khusus pada mikrokontroler AVR, untuk mereduksi konsekuensi negatif diatas, Perusahaan Atmel merancang sedemikian rupa sehingga arsitektur AVR ini efisien dalam *mendekode* serta mengeksekusi instruksi-instruksi yang umum dibangkitkan oleh *compiler C*. Dalam kenyataannya, pengembangan arsitektur AVR ini tidak dilakukan sendiri oleh perusahaan Atmel tetapi ada kerja sama dengan salah satu vendor pemasok *compiler C* untuk mikrokontroler tersebut, yaitu

IAR C.

Tabel 13. Beberapa *compiler C* untuk mikrokontroler AVR

<i>Compiler C</i>	<i>Platform</i>	Keterangan
IARC C	-DOS -Windows	Komersil
Code VisionAVR	-Windows	Komersil
ImageCraft's C	-DOS -Windows -Linux	Komersil
AVR-GCC	-DOS -Windows	General Public Licence
C-AVR	-Windows	Komersil
Small C for AVR	-DOS	Komersil
GNU C for AVR	-Linux	General Public Licence
LCC-AVR	-Linux -Windows	Free
Dunfields AVR	-Windows	Komersil

1. Struktur Dasar Pemrograman Arduino menggunakan Bahasa C

a. Struktur

Struktur dasar arduino hanya terjadi dalam dua bagian:

```
Void setup()
{
  // Statement; di eksekusi satu kali
}
Void loop()
{
  // Statement; di eksekusi terus menerus
}
```

b. *Setup ()*

Fungsi *setup()* hanya dipanggil satu kali ketika program pertama kali di jalankan. Ini digunakan untuk mendefinisikan mode pin atau memulai komunikasi serial. Fungsi *setup()* harus di ikut sertakan dalam program, walaupun tidak ada statement yang di jalankan.

Contoh pemrograman yang menggunakan fungsi *setup* sebagai berikut:

```
Void setup ()
{
  pinMode(3,OUTPUT) ; // men-set pin 3 sebagai Output
  pinMode(6, INPUT); // men-set pin 6 sebagai Input
  pinMode(9, INPUT_PULLUP); //men-set pin 9 sebagai Input Pullup
  Serial.begin(9600);
}
```

Keterangan:

pinMode() = berfungsi untuk mengatur fungsi sebuah pin sebagai INPUT atau OUTPUT.

Serial.begin(9600) = digunakan untuk mengaktifkan fitur UART dan menginialisasinya.

c. *Loop()*

Setelah melakukan fungsi *setup()* maka secara langsung akan melakukan fungsi *loop()* secara berurutan dan melakukan instruksi-instruksi yang ada dalam fungsi *loop()*.

```
void loop()
{
  If (digitalread(6)==HIGH) // membaca input digital pin 6
  {
    digitalwrite(3, HIGH); // nyalakan pin 3
  }
  Else if (digitalread(9)==LOW) // membaca input dari pin digital 9
  {
    digitalwrite(3, LOW); //matikan pin 3
  }
}
```

Keterangan:

digitalwrite: Untuk memberikan nilai LOW dan HIGH pada sebuah pin *output*.

digitalread: Untuk membaca logika LOW dan HIGH.

d. Function

Function (fungsi) adalah blok pemrograman yang mempunyai nama dan mempunyai *statement* yang akan dieksekusi ketika *function* di panggil. Cara pendeklarasian *function*:

Type function Name (parameters)

```
{
  //statement;
}
```

Contoh:

```
float delayVal()
{
  Float Data; // membuat variabel "Data" bertipe float
  Data = analogRead(A0); // baca nilai sensor pada pin analog A0
  Data /= 4; // konversi data bit dari 0-1023 ke 0-255
}
```

Pada contoh diatas fungsi tersebut memiliki nilai balik "*float*". Jika tidak menghendaki adanya nilai balik maka tipe function harus "*void*".

e. *{ } Curly Braces*

Curly brace mendefinisikan awal dan akhir dari sebuah blok fungsi. *Curly brace* ini sangat penting untuk pemrograman pada Bahasa C. Jika pada sebuah blok fungsi program, awalan dan akhiran sebuah *statement* tidak diberi *Curly brace* maka program akan *error* ketika di *compile*.

f. *; Semicolon*

Semicolon harus diberikan pada setiap *statement* program yang kita buat, Karena ini merupakan pembatas dari setiap *statement* program yang dibuat.

g. */*...*/ Blok Comment*

Semua *statement* yang ditulis dalam *block comments* tidak akan dieksekusi dan tidak akan di *compile* sehingga tidak mempengaruhi besar program yang dibuat untuk di masukkan dalam *board Arduino*.

h. *// Line Comment*

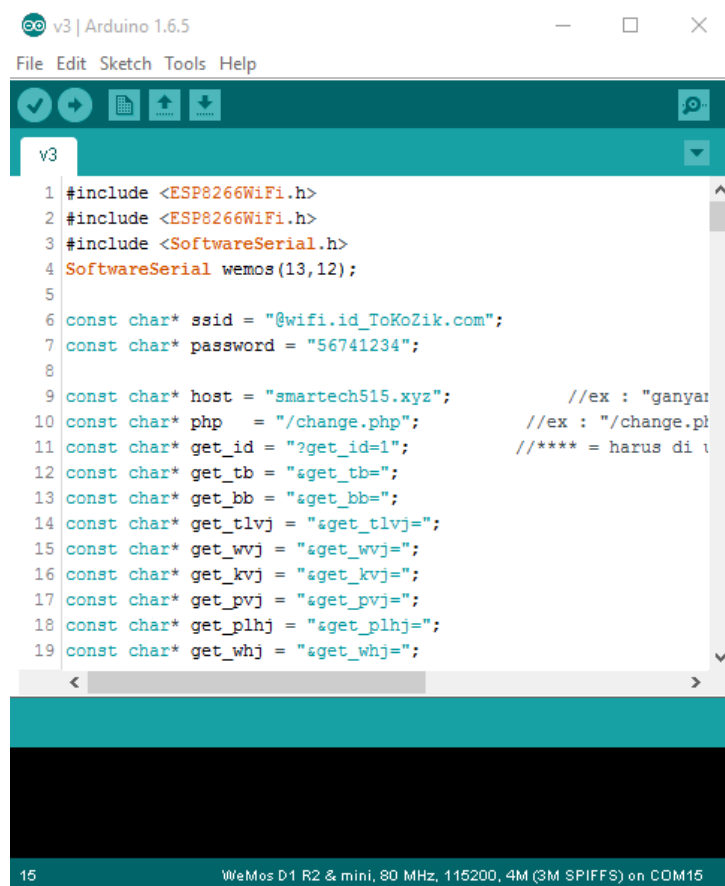
Sama halnya dengan *block comments*, *line comments* pun sama hanya saja yang dijadikan komentar adalah per satu baris.

K. Perangkat Lunak IDE

IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dinamakan melalui sintaks

pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan *library* C/C++ yang bisa disebut *Wiring* yang membuat operasi *input* dan *output* menjadi lebih mudah. Arduino IDE ini dikembangkan dari *software Processing* yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino. Tampilan dari Arduino IDE dapat dilihat pada gambar 16 berikut.



```

v3
1 #include <ESP8266WiFi.h>
2 #include <ESP8266WiFi.h>
3 #include <SoftwareSerial.h>
4 SoftwareSerial wemos(13,12);
5
6 const char* ssid = "@wifi.id_ToKoZik.com";
7 const char* password = "56741234";
8
9 const char* host = "smartech515.xyz"; //ex : "ganyar
10 const char* php = "/change.php"; //ex : "/change.ph
11 const char* get_id = "?get_id=1"; //**** = harus di
12 const char* get_tb = "%get_tb=";
13 const char* get_bb = "%get_bb=";
14 const char* get_tlvj = "%get_tlvj=";
15 const char* get_wvj = "%get_wvj=";
16 const char* get_kvj = "%get_kvj=";
17 const char* get_pvj = "%get_pvj=";
18 const char* get_plhj = "%get_plhj=";
19 const char* get_whj = "%get_whj=";

```

15 WeMos D1 R2 & mini, 80 MHz, 115200, 4M (3M SPIFFS) on COM15

Gambar 16. IDE Perangkat Lunak Arduino

(Sumber : Dokumentasi pribadi)

L. *Wireless*

Wireless merupakan Koneksi antar suatu perangkat dengan perangkat lainnya tanpa menggunakan kabel atau Metode untuk mengirimkan sinyal melalui suatu ruangan bukannya menggunakan kabel. Gelombang radio dan sinar infra merah bisaa digunakan untuk komunikasi nirkabel.

1. *Wireless Application Protocol*(WAP)

Wireless Application Protocol merupakan sebuah protocol pengembangan dari protocol *wireless* data yang telah ada. Phone.com menciptakan sebuah versi standart HTML (*HyperText Markup Language*)

Internet *protocol* yang didesain khusus untuk transfer informasi antar *mobile network* yang efisien. Terminal *wireless* dengan HDML (*Handheld Device Markup Language*) *microbrowser*, dan *Handheld Device Transport Protocol* (HDTP) dari Phone.com terhubung dengan UP. *Link Server Suite* yang seterusnya terhubung ke Internet atau intranet dimana informasi yang dibutuhkan berada. Teknologi inilah yang kemudian dikenal sebagai WAP.

Keterbatasan perangkat *Wireless Application Protocol* antara lain:

- a. Kemampuan *Central Processing Unit* (CPU) yang lebih rendah dibandingkan CPU yang digunakan pada perangkat *wired* (seperti komputer)
- b. Keterbatasan ukuran memori
- c. Penghematan penggunaan catu daya yang bisaanya menggunakan baterai
- d. Ukuran tampilan yang lebih kecil dan terbatas
- e. Perangkat masukan yang berbeda dengan perangkat bisaa

Desain dari informasi yang dikirimkan melalui WAP bisaanya menggunakan format WML, *Wireless Markup Language*. WML ini mirip HTML, hanya lebih spesifik untuk perangkat *wireless* yang memiliki keterbatasa seperti di atas.

2. *Wireless Bitmap*

Wireless Bitmap isingkat dengan WBMP. Format grafik yang terdapat dalam WAP. WBMP merupakan format yang mirip dengan format BMP. Gambar dengan standar format WBMP terbagi dalam dua bagian, yaitu:

- a. Bagian *header*, tempat untuk informasi karakteristik dari gambar, seperti tinggi, lebar dan tipe gambar.
- b. Bagian isi yang disebut *Type dependent*, merupakan bagian dari informasi gambar.

3. *Wireless Computing*

Wireless Computing merupakan proses komputerisasi yang dilakukan melalui media jaringan tanpa kabel.

4. *Wireless Fidelity*

Wireless Fidelity bisa disingkat dengan WiFi. WiFi merupakan sebuah teknologi yang memungkinkan sejumlah komputer terhubung dalam sebuah jaringan tanpa kabel alias *wireless LAN*.

5. *Wireless Internet Service Provider*

Disingkat dengan WISP. *Internet Access Provider* atau *Internet Service Provider* yang berusaha memberikan layanan sambungan nirkabel *broadband* dan sambungan untuk station bergerak kepada perusahaan pengguna.

6. *Wireless Markup Language*

Disingkat dengan WML. *Wireless Markup Language* merupakan Sebuah standar bahasa yang mirip HTML hanya dikhususkan kepada perangkat *wireless* seperti *handphone*.

7. *Wireless Entertainment Device*

Wireless Entertainment Device adalah PDA yang berukuran saku yang memiliki kemampuan konektivitas nirkabel dan dirancang untuk

online gaming, konsepnya ialah penggabungan dari telepon seluler dan perangkat seperti *Gameboy*.

8. *Wireless Card*

Wireless Card merupakan kartu yang digunakan untuk mendukung komputer bisa terhubung dalam suatu jaringan. Kartu ini bisaanya digunakan pada notebook yang disebut dengan PCMCIA (*Personal Computer Memory Card International Association*).

9. *Wireless LAN*

Wireless LAN adalah jaringan komputer yang terhubung melalui tanpa kabel. Local Area Network dari komputer dan peralatan lainnya yang berkomunikasi lewat sinyal radio atau gelombang cahaya. Sistem ini berguna apabila penyambungan lewat koneksi kabel atau serat optik cukup mahal atau untuk aplikasi koneksi bergerak.

10. *Wireless PAN*

Personal Area Network yang terhubung dengan media tanpa kabel. Teknologi yang digunakan pada *wireless PAN* ini adalah IrDA dan *Bluetooth*.

11. *Wireless Modem*

Wireless Modem merupakan modem yang digunakan untuk jaringan tanpa kabel.

12. *Infra Merah*

Gelombang cahaya infra merah. Gelombang ini dapat digunakan untuk proses transmisi data untuk jarak dekat. Standard *wireless networking* yang diluncurkan pada dasarnya adalah menggunakan

hubungan radio jarak dekat atau *short-range radio link* untuk pertukaran informasi, sehingga hubungan antar hp, mobile PC, PDA, dan lainnya dapat dilakukan tanpa gangguan kabel atau *wireless*.

13. *Bluetooth*

Tujuan dari peluncuran *Bluetooth* ini diantaranya adalah untuk mengganti spesifikasi IrDA dari Infra merah pada hp dan peralatan *mobile* lainnya. *Bluetooth* menyediakan transfer data 720 Kbps dalam range 40 feet. *Bluetooth* menggunakan gelombang radio yang omni direksional dan dapat menembus dinding. Ini berbeda dengan IrDa yang menggunakan teknologi pandang dan perlu satu sama lain agar bisa melakukan kontak.

14. 3G

3G atau *third generation* adalah istilah yang digunakan untuk sistem komunikasi *mobile* generasi selanjutnya. Sistem ini akan memberikan pelayanan yang lebih baik dari apa yang ada sekarang, yaitu pelayanan suara, text dan data. Keuntungan utama adalah sistem ini akan menawarkan pelayanan dengan kapabilitas *high-end*, yang mana termasuk peningkatan kapasitas, kualitas dan data *rate* dari apa yang ada sekarang. Juga akan dapat melakukan pemakaian serentak dari beberapa jasa pelayanan. Sistem 3rd *Generation* juga akan menjembatani celah yang ada antara dunia *wireless* dan dunia computer/internet.

15. *Service Set Identifier*

Nama dari suatu *wireless local area network*, digunakan pada semua perangkat nirkabel agar bisa untuk berkomunikasi satu sama lainnya.

16. Antena

Antena merupakan suatu alat yang digunakan untuk mengirim maupun untuk menerima suatu sinyal. Antena ini lebih ditujukan untuk jaringan tanpa kabel, seperti antenna televisi, antena handphone, antenna untuk WLAN (*Wireless Local Area Network*), *satellite dish*.

M. Website

Website merupakan kumpulan halaman web yang saling terhubung dan *filefilenya* saling terkait. Web terdiri dari *page* atau halaman, dan kumpulan halaman yang dinamakan *homepage*. *Homepage* berada pada posisi teratas, dengan halaman-halaman terkait berada di bawahnya. Bisaanya setiap halaman di bawah *homepage* disebut *child page*, yang berisi *hyperlink* ke halaman lain dalam web. (Gregorius, 2000, h:30)

Website awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep *hyperlink*, yang memudahkan surfer atau pengguna internet melakukan penelusuran informasi di internet. Informasi yang disajikan dengan web menggunakan konsep multimedia, informasi dapat disajikan dengan menggunakan banyak media, seperti teks, gambar, animasi, suara, atau film. Berikut adalah jenis-jenis script pada *website*:

1. HTML
2. *Cascading Style Sheets (CSS)*
3. *JavaScript*
4. PHP
5. PHP Untuk MySQL

N. *Rotary Encoder*

Rotary encoder adalah peralatan elektro-mekanik yang menggunakan sensor optic untuk menghasilkan rentetan gelombang pulsa – pulsa yang dapat diubah kedalam suatu gerakan, posisi atau arah. Contoh dari *rotary encoder* dapat dilihat pada gambar 17.



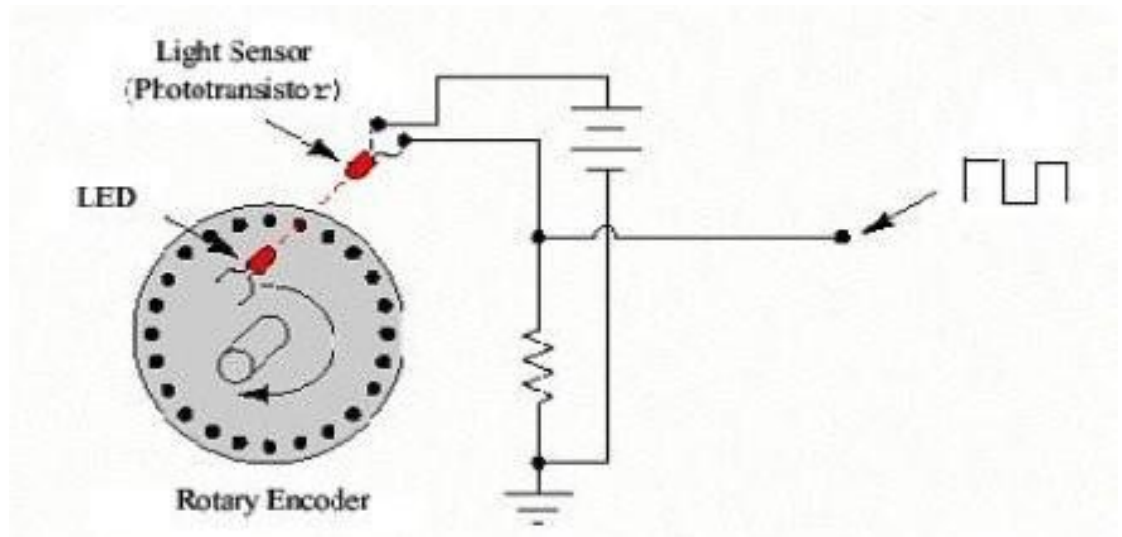
Gambar 17. *Rotary Encoder*

(Sumber : <https://www.alibaba.com/showroom/korea-encoder.html>)

Pada gambar 18 menunjukkan prinsip kerja secara umum dari *rotary encoder*, dimana sebuah piringan tipis dan LED yang ditempatkan sedemikian rupa sehingga cahayanya tetap terfokus pada piringan tersebut. Sebuah transistor aktif cahaya ditempatkan pada sisi lain dari piringan sehingga dapat mendeteksi cahaya dari LED.

Piringan tersebut ditempatkan pada poros (*shaft*) yang bergerak. Dimana pergerakan piringan tersebut sesuai dengan pergerakan poros (*shaft*) sehingga ketika poros (*shaft*) berputar, maka piringan ikut berputar. Ketika piringan yang di atasnya ditempatkan LED yang memancarkan cahaya yang terfokus

terhadap fototransistor akan berada dalam keadaan saturasi yang keluarannya akan berupa pulsa gelombang kotak.



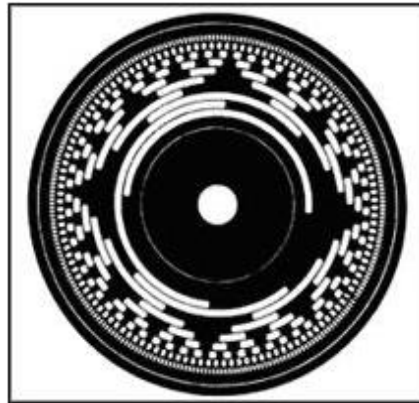
Gambar 18. Prinsip Kerja *Rotary Encoder*

(Sumber : <https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/>)

1. *Absolute Rotary Encoder*

Absolute encoder menggunakan piringan dan sinyal optik yang diatur sedemikian sehingga dapat menghasilkan kode digital untuk menyatakan sejumlah posisi tertentu dari poros yang dihubungkan padanya. Piringan yang digunakan untuk *absolut encoder* tersusun dari segmen-segmen cincin konsentris yang dimulai dari bagian tengah piringan ke arah tepi luar piringan yang jumlah segmennya selalu dua kali jumlah segmen cincin sebelumnya. Cincin pertama di bagian paling dalam memiliki satu segmen transparan dan satu segmen gelap, cincin kedua memiliki dua segmen transparan dan dua segmen gelap, dan seterusnya hingga cincin terluar.

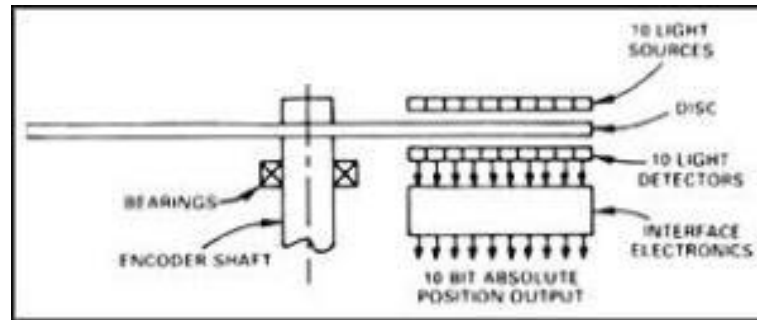
Sebagai contoh apabila absolut *encoder* memiliki 16 cincin konsentris maka cincin terluarnya akan memiliki 32767 segmen. Gambar 19 menunjukkan pola cincin pada piringan absolut *encoder* yang memiliki 16 cincin.



Gambar 19. Contoh susunan pola 16 cincin konsentris pada absolut *encoder*

(Sumber : <https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/>)

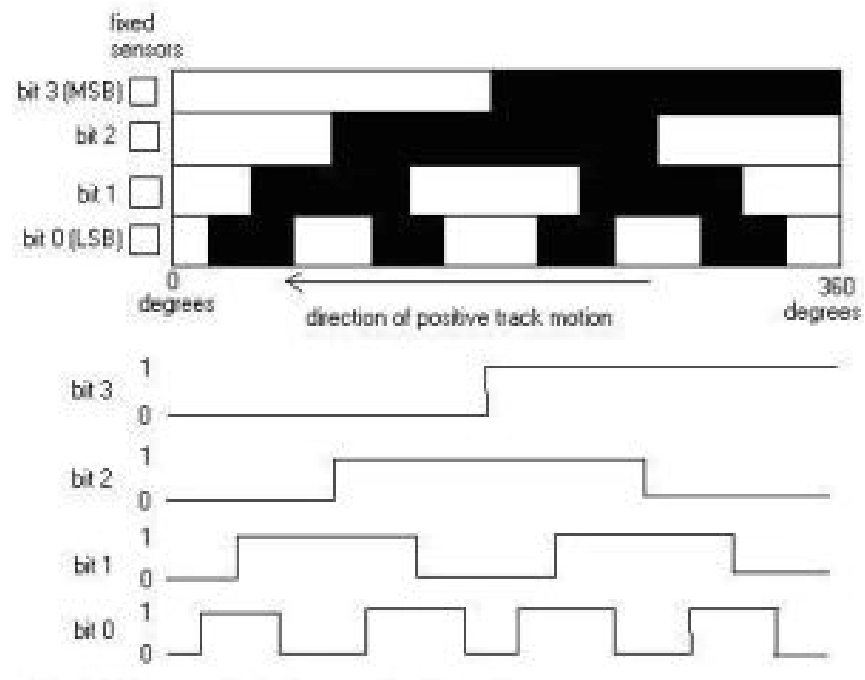
Karena setiap cincin pada piringan absolute *encoder* memiliki jumlah segmen kelipatan dua dari cincin sebelumnya, maka susunan ini akan membentuk suatu sistem biner. Untuk menghasilkan sistem biner pada susunan cincin maka diperlukan pasangan LED dan *photo-transistor* sebanyak jumlah cincin yang ada pada absolut *encoder* tersebut.



Gambar 20. Contoh piringan dengan 10 cincin dan 10 LED – photo-transistor untuk membentuk sistem biner 10 bit.

(Sumber : <https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/>)

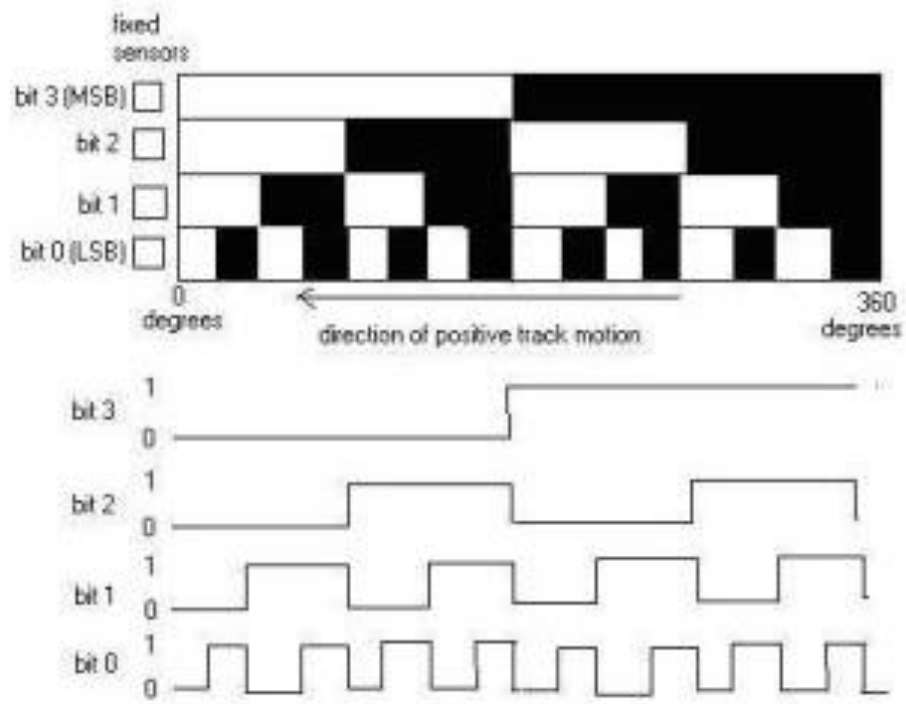
Sistem biner yang untuk menginterpretasi posisi yang diberikan oleh absolute *encoder* dapat menggunakan kode gray atau kode biner bisa, tergantung dari pola cincin yang digunakan. Untuk lebih jelas, kita lihat contoh absolut *encoder* yang hanya tersusun dari 4 buah cincin untuk membentuk kode 4 bit. Apabila *encoder* ini dihubungkan pada poros, maka photo-transistor akan mengeluarkan sinyal persegi sesuai dengan susunan cincin yang digunakan. Gambar 21 dan 22 menunjukkan contoh perbedaan diagram keluaran untuk absolute *encoder* tipe *gray code* dan tipe *binary code*.



Gambar 21. Contoh diagram keluaran absolut *encoder* 4-bit tipe gray code

(Sumber : <https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/>)

Dengan *absolute encoder* 4-bit ini maka kita akan mendapatkan 16 informasi posisi yang berbeda yang masing-masing dinyatakan dengan kode biner atau kode *gray* tertentu. Tabel 14 menyatakan posisi dan *output* biner yang bersesuaian untuk absolut *encoder* 4-bit. Dengan membaca *output* biner yang dihasilkan maka posisi dari poros yang kita ukur dapat kita ketahui untuk diteruskan ke rangkaian pengendali. Semakin banyak bit yang kita pakai maka posisi yang dapat kita peroleh akan semakin banyak.



Gambar 22. Contoh diagram keluaran absolut *encoder* 4-bit tipe binary code

(Sumber : <https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/>)

Tabel 14. *Output* biner dan posisi yang bersesuaian pada absolute *encoder* 4-bit

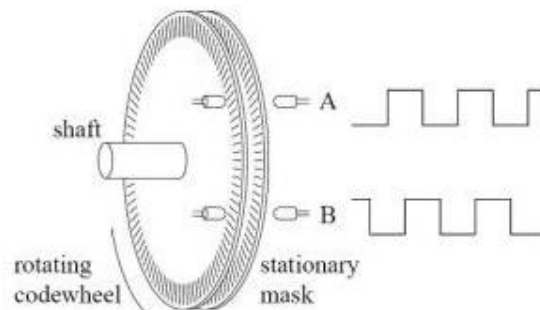
Desimal	Rentang Putaran	Kode Biner	Kode Gray
0	0 - 22.5	0000	0000
1	22.5 - 45	0001	0001
2	45 - 67.5	0010	0011
3	67.5 - 90	0011	0010
4	90 - 112.5	0100	0110
5	112.5 - 135	0101	0111
6	135 - 157.5	0110	0101
7	157.5 - 180	0111	0100
8	180 - 202.5	1000	1100
9	202.5 - 225	1001	1101
10	225 - 247.5	1010	1111
11	247.5 - 270	1011	1110

12	270 – 292.5	1100	1010
13	292.5 – 315	1101	1011
14	315 – 337.5	1110	1001
15	337.5 – 360	1111	1000

(Sumber : <https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/>)

2. Incremental Rotary Encoder

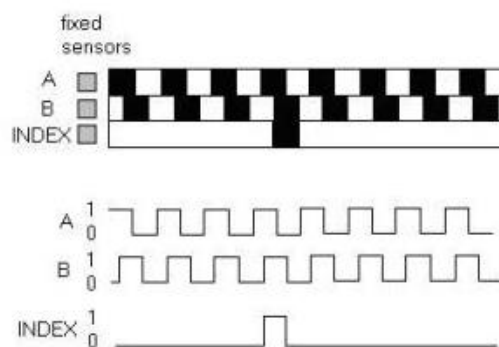
Incremental encoder terdiri dari *dual track* atau *single track* dan dua sensor yang disebut *channel A* dan *B* (Gambar 23). Ketika poros berputar, deretan pulsa akan muncul di masing-masing *channel* pada frekuensi yang proporsional dengan kecepatan putar sedangkan hubungan fasa antara *channel A* dan *B* menghasilkan arah putaran. Dengan menghitung jumlah pulsa yang terjadi terhadap resolusi piringan maka putaran dapat diukur. Untuk mengetahui arah putaran, dengan mengetahui *channel* mana yang *leading* terhadap *channel* satunya dapat kita tentukan arah putaran yang terjadi karena kedua *channel* tersebut akan selalu berbeda fasa seperempat putaran (*quadrature signal*). Seringkali terdapat *output channel* ketiga, disebut INDEX, yang menghasilkan satu pulsa per putaran berguna untuk menghitung jumlah putaran yang terjadi.



Gambar 23. susunan piringan untuk *incremental encoder*

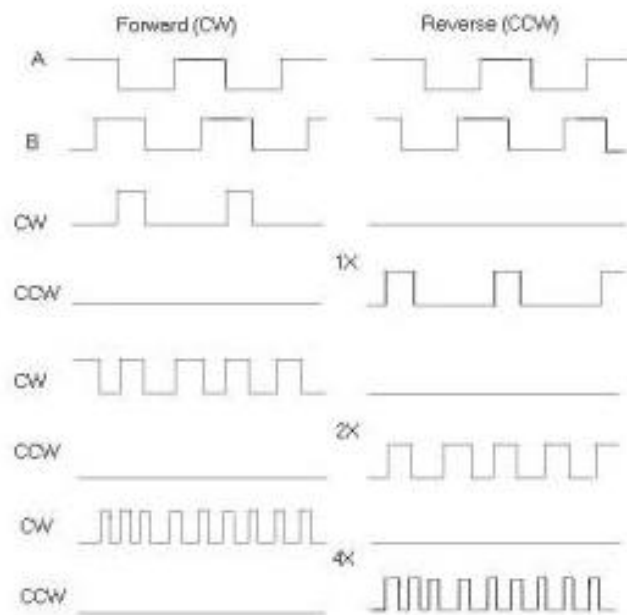
(Sumber : <https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/>)

Contoh pola diagram keluaran dari suatu incremental *encoder* ditunjukkan pada Gambar 24. Resolusi keluaran dari sinyal quadrature A dan B dapat dibuat beberapa macam, yaitu 1X, 2X dan 4X. Resolusi 1X hanya memberikan pulsa tunggal untuk setiap siklus salah satu sinyal A atau B, sedangkan resolusi 4X memberikan pulsa setiap transisi pada kedua sinyal A dan B menjadi empat kali resolusi 1X. Arah putaran dapat ditentukan melalui level salah satu sinyal selama transisi terhadap sinyal yang kedua. Pada contoh resolusi 1X, A = arah bawah dengan B = 1 menunjukkan arah putaran searah jarum jam, sebaliknya B = arah bawah dengan A = 1 menunjukkan arah berlawanan jarum jam.



Gambar 24. Contoh pola keluaran *incremental encoder*

(Sumber : <https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/>)



Gambar 25. *output* dan arah putaran pada resolusi yang berbeda-beda

(Sumber : <https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/>)

Pada *incremental encoder*, beberapa cara dapat digunakan untuk menentukan kecepatan yang diamati dari sinyal pulsa yang dihasilkan. Diantaranya adalah menggunakan *frequencymeter* dan *periodimeter*.

$$\omega_r = \frac{\alpha_f}{T}$$

Cara yang sederhana untuk menentukan kecepatan dapat dengan *frequencymeter*, yakni menghitung jumlah pulsa dari *encoder*, n , pada selang waktu yang tetap, T , yang merupakan periode loop kecepatan. Apabila α adalah sudut antara pulsa *encoder*, maka sudut putaran pada suatu periode adalah:

$$\alpha_f = n\alpha$$

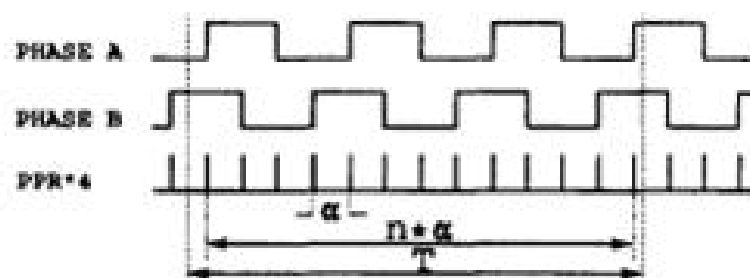
(2)

Sehingga kecepatan putar akan kita dapatkan sebagai:

$$\omega_r = \frac{\alpha_f}{T}$$

(3)

Kelemahan yang muncul pada cara ini adalah pada setiap periode sudut α_f yang didapat merupakan kelipatan integer dari α . Ini akan dapat menghasilkan *quantification error* pada kecepatan yang ingin diukur.



Gambar 26. Sinyal keluaran *encoder* untuk pengukuran kecepatan dengan *frequencymeter*

(Sumber : <https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/>)

Cara yang lain adalah dengan menggunakan periodimeter. Dengan cara ini kita akan mengukur kecepatan tidak lagi dengan menghitung jumlah pulsa *encoder* tetapi dengan menghitung clock frekuensi tinggi (HF Clock) untuk sebuah pulsa dari *encoder* yaitu mengukur periode pulsa dari *encoder*. Apabila α_p adalah sudut dari pulsa *encoder*, t adalah periode dari HF clock,

dan n adalah jumlah pulsa HF yang terhitung pada counter. Maka waktu untuk sebuah pulsa *encoder*, T_p , adalah:

$$T_p = nt$$

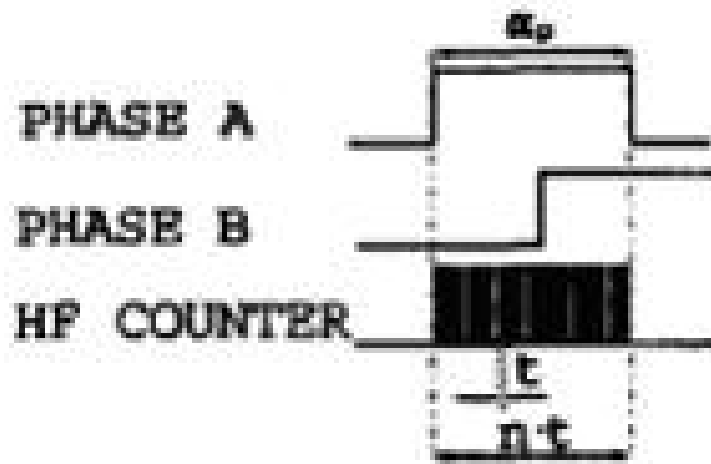
(4)

Sehingga kecepatan yang akan kita ukur dapat kita peroleh dengan:

$$\omega_r = \frac{\alpha_p}{T_p}$$

(5)

Seperti halnya pada frequencymeter, disini juga muncul *quantification error* karena waktu T_p akan selalu merupakan perkalian integer dengan t .



Gambar 27. Pengukuran kecepatan dengan menggunakan Periodimeter

(Sumber : <https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/>)

O. *Liquid crystal Display (LCD)*

Liquid crystal display atau sering disebut LCD merupakan suatu media penampilan data yang sangat efektif dan efisien dalam penggunaannya. Untuk menampilkan sebuah karakter pada layar LCD diperlukan beberapa rangkaian tambahan. Untuk lebih memudahkan para pengguna, maka beberapa perusahaan elektronik menciptakan modul LCD. Adapun bentuk fisik LCD 20x4 seperti pada Gambar 28 dibawah.



Gambar 28. LCD karakter 20 x 4

(Sumber : <http://www.gravitech.us/20chyelcdwib.html>)

LCD dibagi menjadi dua bagian yaitu bagian depan panel LCD yang terdiri dari banyak dot atau titik LCD dan mikrokontroler yang menempel pada bagian belakang panel LCD yang berfungsi untuk mengatur titik-titik LCD sehingga dapat menampilkan huruf, angka, dan simbol khusus yang dapat terbaca.

1. Fungsi Pin LCD

Modul LCD berukuran 20 karakter x 4 baris dengan fasilitas *BACKlighting* memiliki 16 pin yang terdiri dari 8 jalur data, 3 jalur kontrol

dan jalur-jalur catu daya, dengan fasilitas pin yang tersedia maka lcd 20 x 4 dapat digunakan secara maksimal untuk menampilkan data yang dikeluarkan oleh mikrokontroler, secara ringkas fungsi pin-pin pada LCD dituliskan pada Tabel 15.

Tabel 15. Fungsi-fungsi pin pada LCD 20x4

PIN NUMBER	SYMBOL	FUNCTION
1	V _{ss}	GND
2	V _{dd}	+ 3V or +5V
3	V _o	Contrast Adjustment
4	RS	H/L Register Select <i>Signal</i>
5	RW	H/L Read/Write <i>Signal</i>
6	E	H->L <i>Enable Signal</i>
7	DB0	H/L Data Bus <i>Line</i>
8	DB1	H/L Data Bus <i>Line</i>
9	DB2	H/L Data Bus <i>Line</i>
10	DB3	H/L Data Bus <i>Line</i>
11	DB4	H/L Data Bus <i>Line</i>
12	DB5	H/L Data Bus <i>Line</i>
13	DB6	H/L Data Bus <i>Line</i>
14	DB7	H/L Data Bus <i>Line</i>
15	A/V _{cc}	+4.2V for LED/Negative <i>Voltage Output</i>
16	K	<i>Power supply for B/L (0V)</i>

Secara umum pin-pin LCD diterangkan sebagai berikut :

a. Pin 1 dan 2

Merupakan sambungan catu daya, V_{ss} dan V_{dd}. Pin V_{dd} dihubungkan dengan tegangan positif catu daya, dan V_{ss} pada 0V atau ground. Meskipun data menentukan catu 5 Vdc (hanya pada beberapa mA), menyediakan 6V dan 4.5V yang keduanya bekerja dengan baik, bahkan 3V cukup untuk beberapa modul.

b. Pin 3

Pin 3 merupakan pin kontrol Vee, yang digunakan untuk mengatur kontras *display*. Idealnya pin ini dihubungkan dengan tegangan yang bisa dirubah untuk memungkinkan pengaturan terhadap tingkatan kontras *display* sesuai dengan kebutuhan, pin ini dapat dihubungkan dengan variable resistor sebagai pengatur kontras.

c. Pin 4

Pin 4 merupakan Register Select (RS), masukan yang pertama dari tiga command control *input*. Dengan membuat RS menjadi *high*, data karakter dapat ditransfer dari dan menuju modulnya.

d. Pin 5

Read/Write (R/W), untuk memfungsikan sebagai perintah *write* maka R/W *low* atau menulis karakter ke modul. R/W *high* untuk membaca data karakter atau informasi status dari register-nya.

e. Pin 6

Enable (E), *input* ini digunakan untuk transfer aktual dari perintah-perintah atau karakter antara modul dengan hubungan data. Ketika menulis ke *display*, data ditransfer hanya pada perpindahan *high* atau *low*. Tetapi ketika membaca dari *display*, data akan menjadi lebih cepat tersedia setelah perpindahan dari *low* ke *high* dan tetap tersedia hingga sinyal *low* lagi.

f. Pin 7 sampai 14

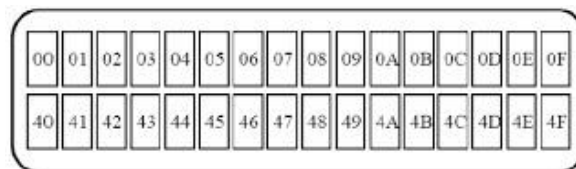
Pin 7 sampai 14 adalah delapan jalur data/data bus (D0 sampai D7) dimana data dapat ditransfer ke dan dari *display*.

g. Pin 16

Pin 16 dihubungkan kedalam tegangan 5 Volt untuk memberi tegangan dan menghidupkan lampu latar/*BACK* Light LCD.

2. Pengalamatan LCD

Pengalamatan LCD dimulai dengan menghidupkan modul LCD, karakter kursor pada LCD diposisikan pada awal baris pertama (alamat 00H). Masing-masing sewaktu sebuah karakter dimasukkan, kursor bergerak ke alamat selanjutnya 01H, 02H dan seterusnya. Sebuah alamat awal yang baru bergerak ke alamat selanjutnya, harus dimasukkan sebagai sebuah perintah. Dengan cara mengirimkan sebuah perintah Set *Display* Address, nilai 80H. Dengan dua *line* karakter, baris yang pertama dari karakter, baris pertama mulai pada alamat 00H dan baris ke dua pada alamat 40H. Hubungan antara tata letak alamat-alamat terlihat pada Gambar 29 dibawah ini.



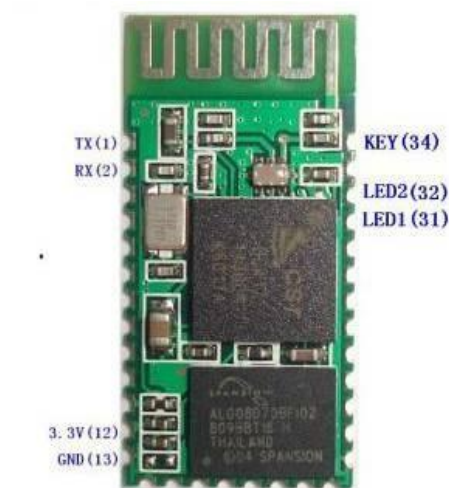
Gambar 29. Pengalamatan LCD 20 x 4

(Sumber : <http://www.gravitech.us/20chyelcdwib.html>)

P. *Bluetooth* HC-05

Bluetooth adalah protocol komunikasi *wireless* yang bekerja pada frekuensi radio 2.4 GHz untuk pertukaran data pada perangkat bergerak seperti PDA, laptop, HP, dan lain-lain. Salah satu hasil contoh modul *Bluetooth* yang paling banyak digunakan adalah tipe HC-05. Modul *Bluetooth* HC-05

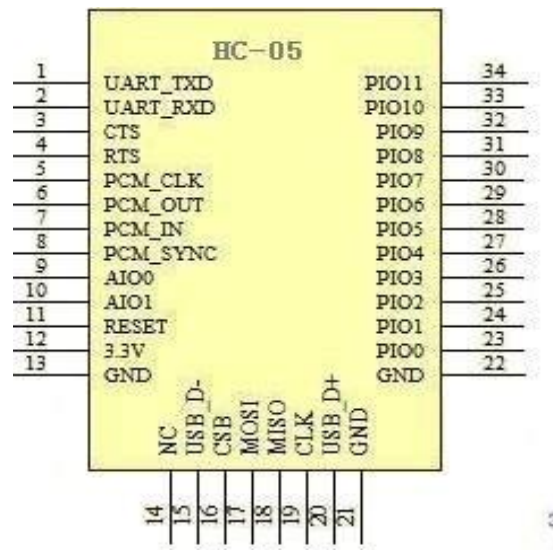
merupakan salah satu modul *Bluetooth* yang dapat ditemukan dipasaran dengan harga yang relatif murah. Modul *Bluetooth* HC-05 terdiri dari 6 pin konektor, yang setiap pin konektor memiliki fungsi yang berbeda-beda. Untuk gambar *module Bluetooth* dapat dilihat pada gambar 30. Dibawah ini:



Gambar 30. Modul *Bluetooth* HC-05

(Sumber: <https://mbed.org/users/edodm85/notebook/HC-05-Bluetooth>)

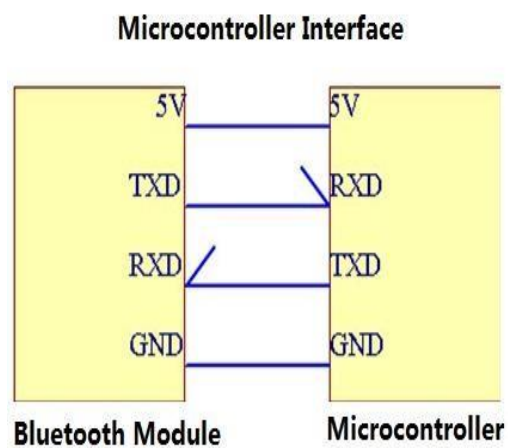
Modul *Bluetooth* HC-05 dengan *supply* tegangan sebesar 3,3 V ke pin 12 modul *Bluetooth* sebagai VCC. Pin 1 pada modul *Bluetooth* sebagai transmitter. kemudian pin 2 pada *Bluetooth* sebagai *receiver*. Berikut merupakan konfigurasi pin *Bluetooth* HC-05 ditunjukkan pada gambar 31. Dibawah ini:



Gambar 31. Konfigurasi Pin HC-05

(Sumber: <https://mbed.org/users/edodm85/notebook/HC-05-Bluetooth>)

Berikut ini merupakan *Bluetooth-to-Serial-Module* HC-05 dapat dilihat pada gambar 32. Dibawah ini:



Gambar 32. *Bluetooth-to-Serial-Module* HC-05

(Sumber: <http://tokoone.com/modul-Bluetooth-modul-serial>)

Konfigurasi pin modul *Bluetooth* HC-05 dapat dilihat pada tabel 16 berikut ini :

Tabel 16. Konfigurasi pin *Module Bluetooth HC-05*

No.	Nomor Pin	Nama	Fungsi
1.	Pin 1	Key	-
2.	Pin 2	VCC	Sumber tegangan 5V
3.	Pin 3	GND	Ground tegangan
4.	Pin 4	TXD	Mengirim data
5.	Pin 5	RXD	Menerima data
6.	Pin 6	STATE	-

(Sumber:<http://diytech.net/2012/03/07/dalam-beberapa-aplikasi-atau-disain-kadangkala-kitamemerlukan>)

Module Bluetooth HC-05 merupakan *module* yang bisa menjadi *slave* ataupun *master* hal ini dibuktikan dengan bisa memberikan notifikasi untuk melakukan *pairing* keperangkat lain, maupun perangkat lain tersebut yang melakukan *pairing* ke *module Bluetooth HC-05*. Untuk mengeset perangkat *Bluetooth* dibutuhkan perintah-perintah *AT Command* yang mana perintah *AT Command* tersebut akan direspon oleh perangkat *Bluetooth* jika *module Bluetooth* tidak dalam keadaan terhubung dengan perangkat lain. Keterangan *AT Command Module Bluetooth HC-05* dapat dilihat pada tabel 17 berikut:

Tabel 17. *AT Command Module Bluetooth HC-05*

No.	Perintah	Kirim	Terima	Keterangan
1.	Tes Komunikasi	AT	OK	-
2.	Ganti Nama <i>Bluetooth</i>	AT+NAME=(nama <i>Bluetooth</i>)	OK (nama <i>Bluetooth</i>)	-
3.	Ubah Password	AT+PSWD=(password)	OK (password)	Xxxx digit key
4.	Ubah Baudrate	AT+BAUD1 AT+BAUD2 AT+BAUD3 AT+BAUD4	OK 1200 OK 2400 OK 4800 OK 9600	1→1200 2→2400 3→4800 4→9600

		AT+BAUD5	OK 19200	5→19200
		AT+BAUD6	OK 38400	6→38400
				7→57600
				8→115200

(Sumber: <http://diytech.net/2013/10/09/mengenal-Bluetooth-modul-hc-05-1>)

Q. *Buzzer*

Buzzer Listrik adalah sebuah komponen elektronika yang dapat mengubah sinyal listrik menjadi getaran suara. Pada umumnya, *Buzzer* yang merupakan sebuah perangkat audio ini sering digunakan pada rangkaian antimaling, Alarm pada Jam Tangan, Bel Rumah, peringatan mundur pada Truk dan perangkat peringatan bahaya lainnya. Jenis *Buzzer* yang sering ditemukan dan digunakan adalah *Buzzer* yang berjenis *Piezoelectric*, hal ini dikarenakan *Buzzer Piezoelectric* memiliki berbagai kelebihan seperti lebih murah, relatif lebih ringan dan lebih mudah dalam menggabungkannya ke Rangkaian Elektronika lainnya. *Buzzer* yang termasuk dalam keluarga Transduser ini juga sering disebut dengan Beeper.

Efek *Piezoelectric* (*Piezoelectric Effect*) pertama kali ditemukan oleh dua orang fisikawan Perancis yang bernama Pierre Curie dan Jacques Curie pada tahun 1880. Penemuan tersebut kemudian dikembangkan oleh sebuah perusahaan Jepang menjadi Piezo Electric *Buzzer* dan mulai populer digunakan sejak 1970an. Untuk penampakan hardware dari *Buzzer* dapat dilihat pada gambar 33 berikut.

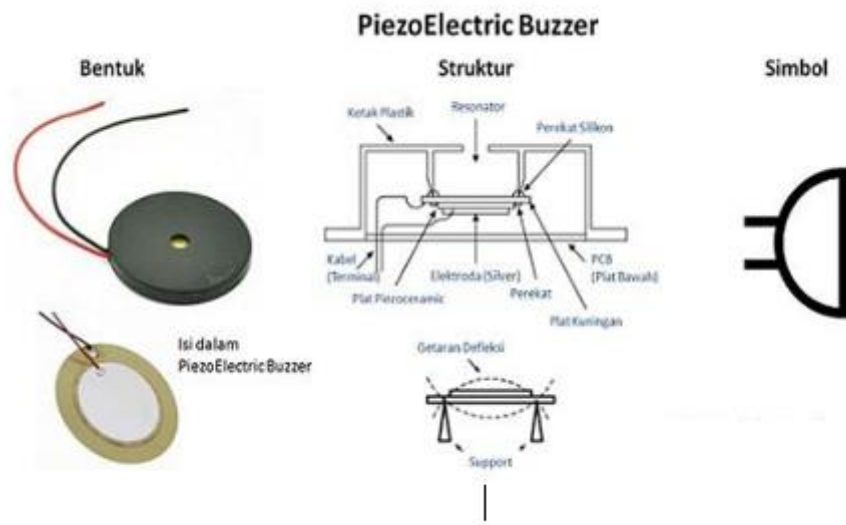


Gambar 33. *Buzzer*

(Sumber : <http://www.instructables.com/id/How-to-use-a-Buzzer-Arduino-Tutorial/>)

3. Cara Kerja *Piezoelectric Buzzer*

Seperti namanya, *Buzzer* adalah jenis *Buzzer* yang menggunakan efek *Piezoelectric* untuk menghasilkan suara atau bunyinya. Tegangan listrik yang diberikan ke bahan *Piezoelectric* akan menyebabkan gerakan mekanis, gerakan tersebut kemudian diubah menjadi suara atau bunyi yang dapat didengar oleh telinga manusia dengan menggunakan diafragma dan resonator. Gambar 34 menunjukkan bentuk dan struktur dasar dari sebuah *Piezoelectric Buzzer*.



Gambar 34. *Piezoelectric Buzzer*

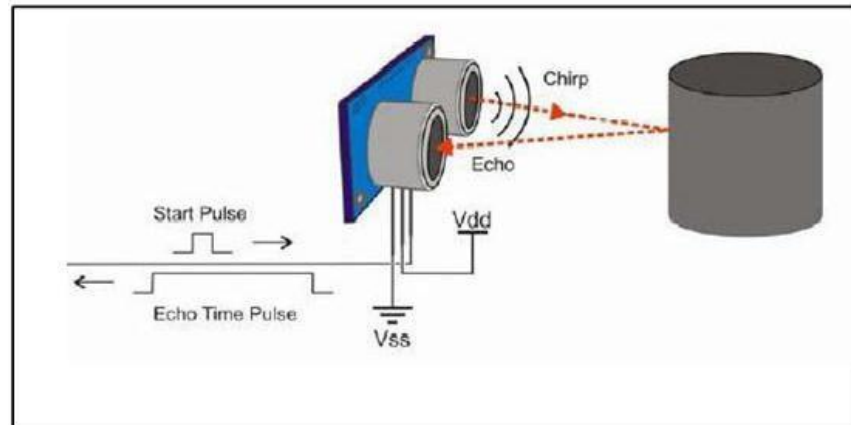
(Sumber : <http://www.robotics-university.com/2014/12/motor-dc-magnet-permanent-dcmp.html/>)

Jika dibandingkan dengan Speaker, Piezo *Buzzer* relatif lebih mudah untuk digerakan. Sebagai contoh, Piezo *Buzzer* dapat digerakan hanya dengan menggunakan *output* langsung dari sebuah IC TTL, hal ini sangat berbeda dengan Speaker yang harus menggunakan penguat khusus untuk menggerakan Speaker agar mendapatkan intensitas suara yang dapat didengar oleh manusia. Piezo *Buzzer* dapat bekerja dengan baik dalam menghasilkan frekuensi di kisaran 1 – 5 kHz hingga 100 kHz untuk aplikasi Ultrasound. Tegangan Operasional *Piezoelectric Buzzer* yang umum bisaanya berkisar diantara 3Volt hingga 24 Volt.

R. Sensor Ultrasonik

Sensor ultrasonik adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara dan digunakan untuk mendeteksi keberadaan suatu objek tertentu di depannya, frekuensi kerjanya pada daerah di atas gelombang suara dari 40 KHz hingga 400 KHz. Sensor ultrasonik terdiri dari dua unit, yaitu unit pemancar dan unit penerima. Struktur unit pemancar dan penerima sangatlah sederhana, sebuah kristal *piezoelectric* dihubungkan dengan mekanik jangkar dan hanya dihubungkan dengan diafragma penggetar. Tegangan bolak-balik yang memiliki frekuensi kerja 40 KHz – 400 KHz diberikan pada plat logam. Struktur atom dari kristal *piezoelectric* akan berkontraksi (mengikat), mengembang atau menyusut terhadap polaritas tegangan yang diberikan dan ini disebut dengan efek *piezoelectric*.

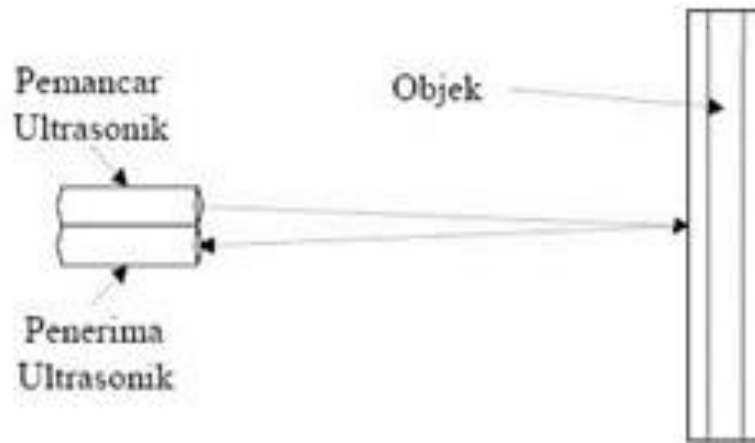
Kontraksi yang terjadi diteruskan ke diafragma penggetar sehingga terjadi gelombang ultrasonik yang dipancarkan ke udara (tempat sekitarnya). Pantulan gelombang ultrasonik akan terjadi bila ada objek tertentu dan pantulan gelombang ultrasonik akan diterima kembali oleh unit sensor penerima. Selanjutnya unit sensor penerima akan menyebabkan diafragma penggetar akan bergetar dan efek *piezoelectric* menghasilkan sebuah tegangan bolak-balik dengan frekuensi yang sama. Untuk lebih jelas tentang prinsip kerja dari sensor ultrasonik dapat dilihat prinsip dari sensor ultrasonik pada gambar 35 dibawah.



Gambar 35. Sensor Ultrasonik

(Sumber : en.wikipedia.org/wiki/Ultrasonik_transducer)

Besar amplitudo sinyal elektrik yang dihasilkan unit sensor penerima tergantung dari jauh dekatnya objek yang dideteksi serta kualitas dari sensor pemancar dan sensor penerima. Proses sensing yang dilakukan pada sensor ini menggunakan metode pantulan untuk menghitung jarak antara sensor dengan objek sasaran. Jarak antara sensor tersebut dihitung dengan cara mengalikan setengah waktu yang digunakan oleh sinyal ultrasonik dalam perjalanannya dari rangkaian pengirim sampai diterima oleh rangkaian penerima, dengan kecepatan rambat dari sinyal ultrasonik tersebut pada media rambat yang digunakannya, yaitu udara. Prinsip pantulan dari sensor ultrasonik ini dapat dilihat pada gambar 36.



Gambar 36. Prinsip Pemantulan Ultrasonik

(Sumber : www.gunturwicaksono.com/2012/08/sensor-ultrasonik-ping-parallax.html)

1. Sensor ultrasonik defantech (SRF 04 *ranger*)

SRF04 adalah sensor non-kontak pengukur jarak menggunakan ultrasonik. Prinsip kerja sensor ini adalah *transmitter* mengirimkan seberkas gelombang ultrasonik, lalu diukur waktu yang dibutuhkan hingga datangnya pantulan dari objek. Lamanya waktu ini sebanding dengan dua kali jarak sensor dengan objek, sehingga jarak sensor dengan objek dapat ditentukan persamaan :

$$s = \frac{v \times t}{2}$$

Keterangan :

s = jarak (meter)

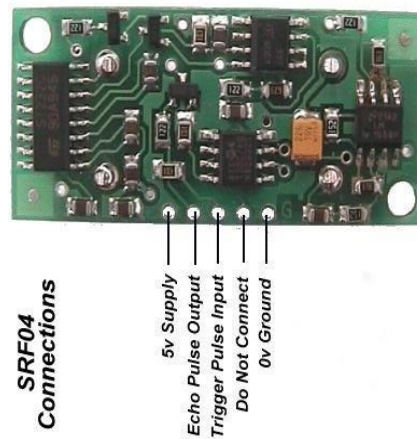
v = kecepatan suara (344 m/detik)

t = waktu tempuh (detik)

SRF04 dapat mengukur jarak dalam rentang antara 3 cm – 3 m dengan *output* panjang pulsa yang sebanding dengan jarak objek. Sensor ini hanya memerlukan 2 pin I/O untuk berkomunikasi dengan mikrokontroler, yaitu *TRIGGER* dan *ECHO*. Untuk mengaktifkan SRF04 mikrokontroler mengirimkan pulsa positif melalui pin *TRIGGER* minimal 10 μ s, selanjutnya SRF04 akan mengirimkan pulsa positif melalui pin *ECHO* selama 100 μ s hingga 18 ms, yang sebanding dengan jarak objek.

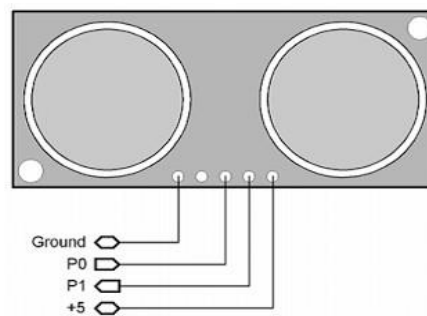
Spesifikasi dari sensor ultrasonik SRF04 adalah sebagai berikut :

- a. Dimensi : 24mm (P) x 20mm (L) x 17mm (T).
- b. Tegangan : 5VDC
- c. Konsumsi Arus : 30 mA (rata-rata), 50 mA (max)
- d. Frekuensi Suara : 40 kHz
- e. Sensitivitas : Mampu mendeteksi objek dengan diameter 3 cm pada jarak > 2 m
- f. *Input Trigger* : 10 ms min. Pulsa Level TTL
: Sinyal Level TTL Positif, Lebar berbanding dengan jarak yang terdeteksi



Gambar 37. Koneksi pada Sensor Ultrasonik SRF04

(Sumber : www.robot-electronics.co.uk/htm/srf04tech.html)



Gambar 38. Konfigurasi Pin SRF 04

(Sumber : www.robot-electronics.co.uk/htm/srf04tech.html)

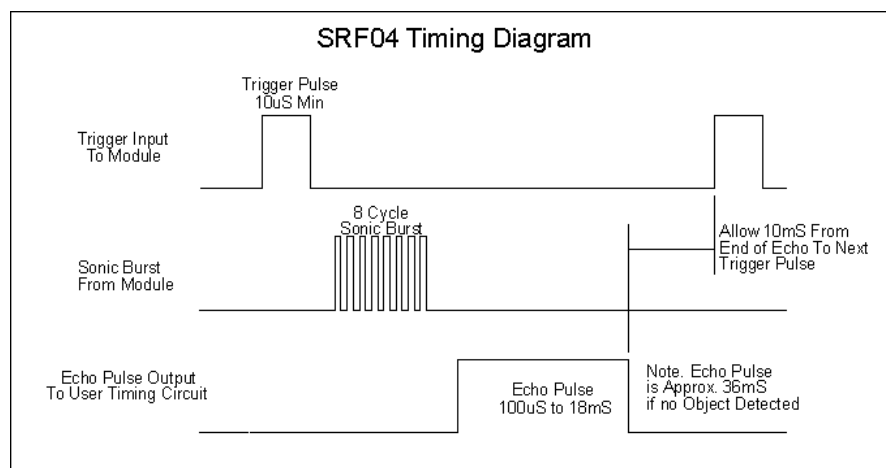
Konfigurasi SRF 04 seperti pada gambar 37. Sensor ini memiliki 5 pin, tetapi hanya 4 pin saja yang digunakan yaitu :

- a. *5V supply*
- b. *EchoPulseOutput*
- c. *Trigger Pulse Input*
- d. *Do Not Connect*

e. 0V Ground

Dari gambar 38 diatas sudah jelas bahwa sensor ini membutuhkan tegangan supply 5VDC yang dihubungkan ke pin 5v *Supply* dan *ground* yang di hubungkan ke pin 0v ground. Pin *echo pulse* berfungsi sebagai pin *output* yang nantinya dihubungkan ke controller sehingga kontroler dapat membaca pulsa yang dihasilkan sensor dan pin *trigger pulse* adalah pin *input* yang nantinya dihubungkan ke kontroler untuk mendapatkan pulsa dari kontroler.

Prinsip kerja SRF04 adalah *transmitter* memancarkan seberkas sinyal ultrasonik (40KHz) yang berbentuk pulsa, kemudian jika di depan SRF04 ada objek padat maka *receiver* akan menerima pantulan sinyal ultrasonik tersebut. *Receiver* akan membaca lebar pulsa (dalam bentuk PWM) yang dipantulkan objek dan selisih waktu pemancaran. Dengan pengukuran tersebut, jarak objek di depan sensor dapat diketahui. Untuk lebih jelasnya, perhatikan gambar 39 di bawah ini :



Gambar 39. Timing Diagram Sensor Utrasonik SRF04

(Sumber : www.robot-electronics.co.uk/htm/srf04tech.html)

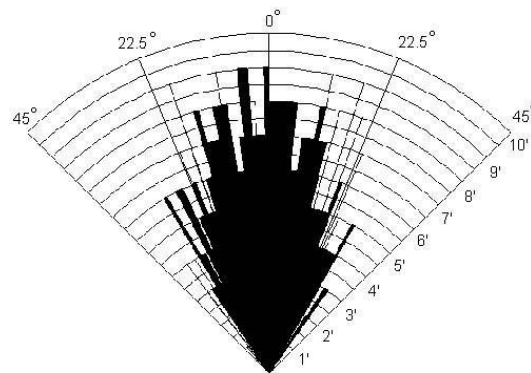
Pin *trigger* dan *echo* dihubungkan ke mikrokontroler. Untuk memulai pengukuran jarak, mikro akan mengeluarkan *output high* pada pin *trigger* selama minimal 10 μ S, sinyal *high* yang masuk tadi akan membuat SRF04 ini mengeluarkan suara ultrasonik. Kemudian ketika bunyi yang dipantulkan kembali ke sensor SRF04, bunyi tadi akan diterima dan membuat keluaran sinyal *high* pada pin *echo* yang kemudian menjadi *inputan* pada mikrokontroler. SRF04 akan memberikan pulsa 100 μ s - 18ms pada *outputnya* tergantung pada informasi jarak pantulan objek yang diterima. Lamanya sinyal *high* dari *echo* inilah yang digunakan untuk menghitung jarak antara sensor SRF04 dengan benda yang memantulkan bunyi yang berada di depan sensor ini.

Untuk menghitung lamanya sinyal *high* yang diterima mikrokontroler dari pin *echo*, maka digunakan fasilitas *timer* yang ada pada masing-masing mikrokontroler. Ketika ada perubahan dari *low* ke *high* dari pin *echo* maka akan mengaktifkan *timer*, dan ketika ada perubahan dari *high* ke *low* dari pin *echo* maka akan mematikan *timer*. Setelah itu yang diperlukan adalah mengkonversi nilai *timer* dari yang satuannya dalam detik, menjadi ke dalam satuan jarak (*inch/cm*) dengan menggunakan rumus berikut :

$$\text{Jarak (inch)} = \text{waktu hasil pengukuran (uS)} / 148$$

$$\text{Jarak (cm)} = \text{waktu hasil pengukuran (uS)} / 58$$

Gambar 40 ini adalah hasil data perbandingan antara sudut pantulan dan jarak pada sensor ultrasonik SRF04 :



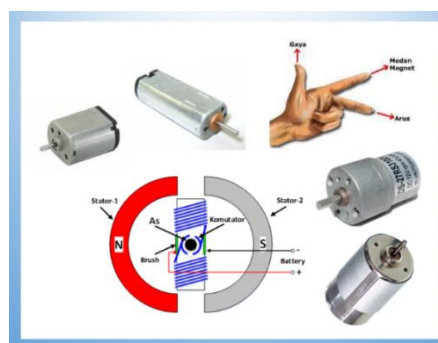
Gambar 40. Perbandingan Sudut Pantul SRF04

(Sumber : www.robot-electronics.co.uk/htm/srf04tech.html)

S. Motor DC

1. Prinsip Kerja

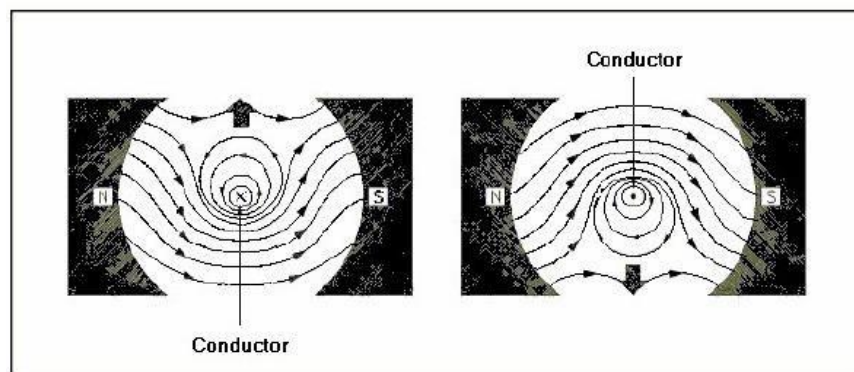
Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet. Arah medan magnet dapat ditentukan dengan kaidah tangan kiri. Ibu jari tangan menunjukkan arah aliran arus listrik sedangkan jari-jari yang lain menunjukkan arah medan magnet yang timbul, seperti yang ditunjukkan oleh gambar 41 berikut ini



Gambar 41. Kaidah Tangan Kiri

(Sumber : <http://www.robotics-university.com/2014/12/motor-dc-magnet-permanent-dcmp.html>)

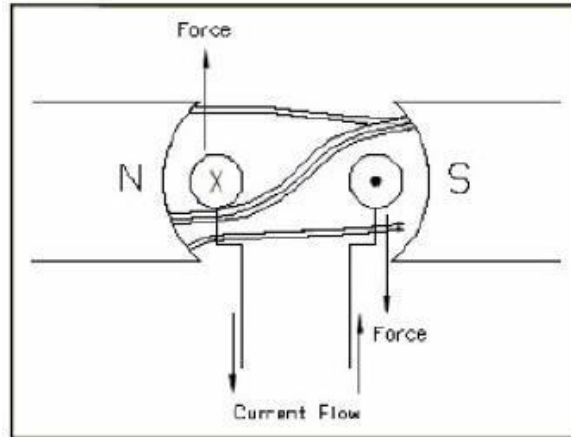
Jika suatu konduktor yang dialiri arus listrik ditempatkan dalam sebuah medan magnet, kombinasi medan magnet akan ditunjukkan oleh gambar 2. Arah aliran arus listrik dalam konduktor ditunjukkan dengan tanda “x” atau “.”. Tanda “x” menunjukkan arah arus listrik mengalir menjauhi pembaca gambar, tanda “.” menunjukkan arah arus listrik mengalir mendekati pembaca gambar 42.



Gambar 42. Konduktor Berarus Listrik Dalam Medan Magnet

(Sumber : <http://elektronika-dasar.web.id/>)

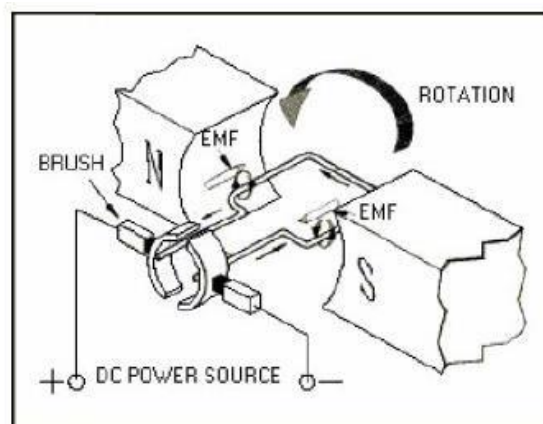
Pada sebuah motor dc, konduktor dibentuk menjadi sebuah loop sehingga ada dua bagian konduktor yang berada didalam medan magnet pada saat yang sama, seperti diperlihatkan pada gambar 43.



Gambar 43. Bergeraknya Sebuah Motor

(Sumber : <http://elektronika-dasar.web.id/>)

Konfigurasi konduktor seperti ini akan menghasilkan distorsi pada medan magnet utama dan menghasilkan gaya dorong pada masing-masing konduktor. Pada saat konduktor di tempatkan pada rotor, gaya dorong yang timbul akan menyebabkan rotor berputar searah dengan jarum jam, seperti diperlihatkan pada gambar 44.



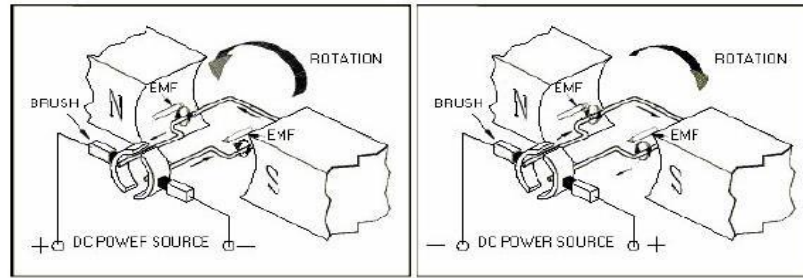
Gambar 44. Konstruksi Dasar Motor DC

(Sumber : <http://elektronika-dasar.web.id/>)

Pada gambar 44 diatas tampak sebuah konstruksi dasar motor dc, pada gambar diatas terlihat bahwa pada saat terminal motor diberi tegangan dc, maka arus elektron akan mengalir melalui konduktor dari terminal negatif menuju ke terminal positif. Karena konduktor berada diantara medan magnet, maka akan timbul medan magnet juga pada konduktor. Arah garis gaya medan magnet yang dihasilkan oleh magnet permanen adalah dari kutub utara menuju ke selatan.

Sementara pada konduktor yang dekat dengan kutub selatan, arah garis gaya magnet disisi sebelah bawah searah dengan garis gaya magnet permanen sedangkan di sisi sebelah atas arah garis gaya magnet berlawanan arah dengan garis gaya magnet permanen. Ini menyebabkan medan magnet disisi sebelah bawah lebih rapat daripada sisi sebelah atas. Dengan demikian konduktor akan terdorong ke arah atas. Sementara pada konduktor yang dekat dengan kutub utara, arah garis gaya magnet disisi sebelah atas searah dengan garis gaya magnet permanen sedangkan di sisi sebelah bawah arah garis gaya magnet berlawanan arah dengan garis gaya magnet permanen. Ini menyebabkan medan magnet disisi sebelah atas lebih rapat daripada sisi sebelah bawah. Dengan demikian konduktor akan terdorong ke arah bawah. Pada akhirnya konduktor akan membentuk gerakan berputar berlawanan dengan jarum jam seperti terlihat pada gambar 44.

2. Membalik Arah Putaran Motor



Gambar 45. Arah Putaran Motor DC

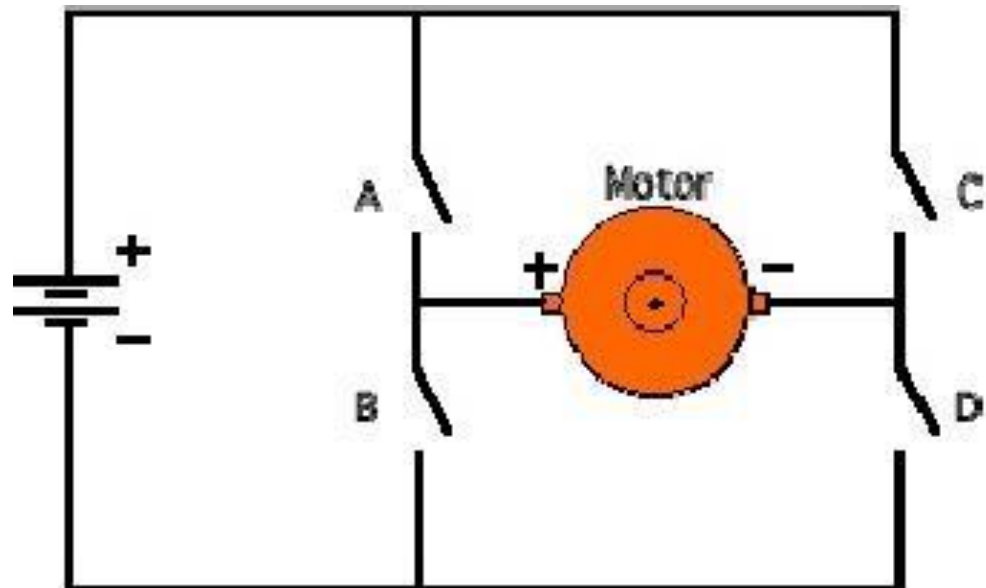
(Sumber : <http://elektronika-dasar.web.id/>)

Dari gambar 45 diatas, untuk membalik arah putaran motor dc, maka polaritas tegangan pada terminal motor harus dibalik.

T. *Driver Motor DC H-Bridge*

H-Bridge secara harfiah adalah jembatan H atau jembatan yang membentuk huruf H. Fungsinya adalah untuk mengontrol motor DC. *H-bridge* bisa untuk mengontrol motor putar kanan, putar kiri dan bisa untuk mengontrol PWM (*Pulse Width Modulation*).

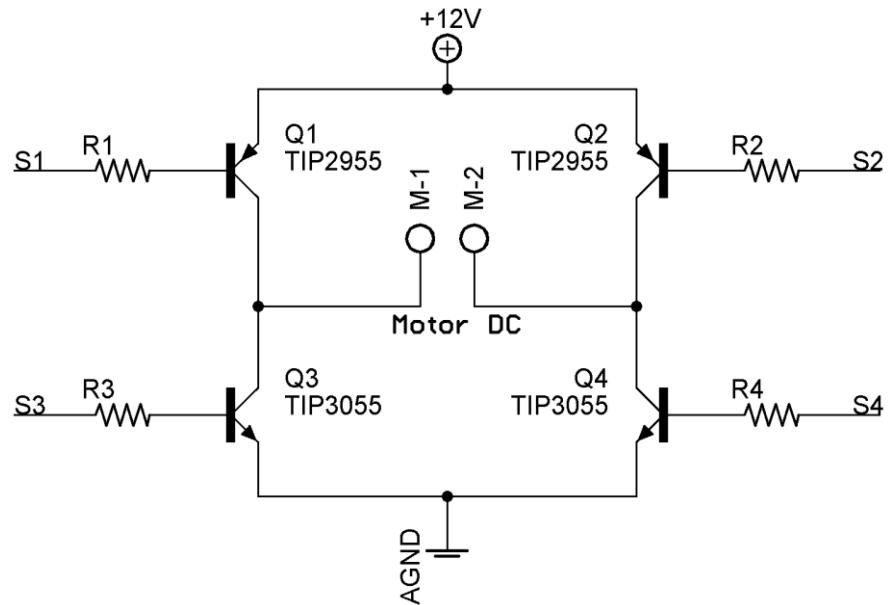
Driver H-bridge bisa di rangkai menggunakan kombinasi saklar, kombinasi Transistor, maupun kombinasi relay. Selain itu, di pasaran, tersedia IC khusus *HBridge*, yakni IC L298 dan L293. Untuk mengatur putaran motor dapat menggunakan konfigurasi *H-Bridge*, seperti tampak pada gambar 46 berikut ini.



Gambar 46. Rangkaian Dasar H-Bridge

(Sumber : <http://www.robotics-university.com/2015/01/driver-motor-dcmp-h-bridge-menggunakan-mosfet.html/>)

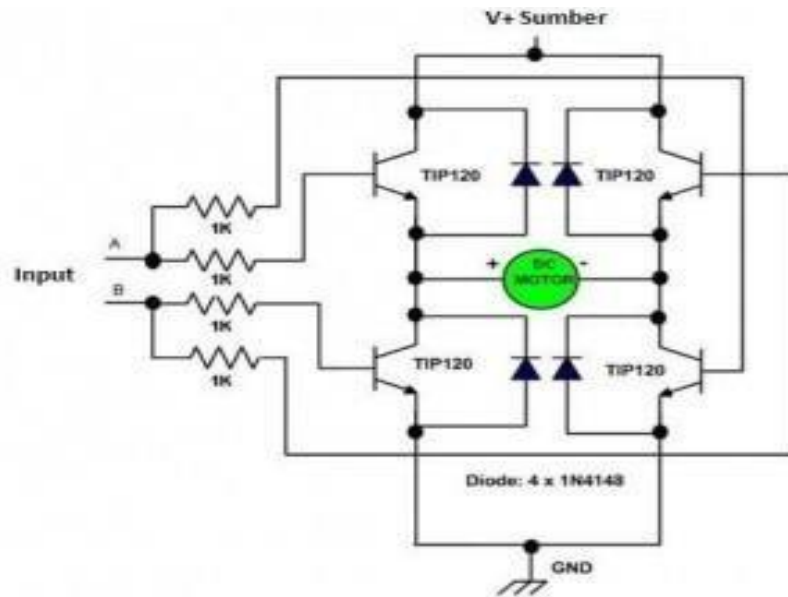
Dengan konfigurasi ini kita akan dapat menjalankan motor DC, membalik arah putaran atau menghentikan dari kontrol logika TTL. Untuk mendapatkan arah putaran pertama kita menutup saklar SA dan SD. Sedangkan untuk membalik arah putaran kita buka saklar SA dan SD, dan menutup saklar SB dan SC. Pada rangkaian sesungguhnya saklar – saklar tersebut diganti dengan transistor sehingga bisa mudah dikontrol oleh MCU. Gambar 47 menunjukkan skema dari driver motor H-*bride* dengan menggunakan transistor.



Gambar 47. H-Bridge dengan saklar Transistor

(Sumber : <http://www.robotics-university.com/2015/01/driver-motor-dcmp-h-bridge-menggunakan-mosfet.html/>)

Rangkaian *driver* motor DC H-Bridge transistor ini dapat mengendalikan arah putaran motor DC dalam 2 arah dan dapat dikontrol dengan metode PWM (*Pulse Width Modulation*) maupun metode sinyal logika dasar TTL (*High*) dan (*Low*). Untuk pengendalian motor DC dengan metode PWM maka dengan rangkaian *driver* motor DC ini kecepatan putaran motor DC dapat dikendalikan dengan baik. Apabila menggunakan metode logika TTL 0 dan 1 maka rangkaian ini hanya dapat mengendalikan arah putaran motor DC saja dengan kecepatan putaran motor DC maksimum. Rangkaian *driver* motor DC H-Bridge ini menggunakan rangkaian jembatan transistor 4 unit dengan proteksi impuls tegangan induksi motor DC berupa dioda yang dipasang paralel dengan masing-masing transistor secara reverse bisa. Gambar 48 menunjukkan rangkaian dari driver motor H-bridge dengan menggunakan transistor.



Gambar 48. Rangkaian *Driver Motor DC H-Bridge Transistor*

(Sumber : <http://elektronika-dasar.web.id/driver-motor-dc-h-bridge-transistor/>)

Proses mengendalikan motor DC menggunakan rangkaian *driver motor DC HBridge* diatas dapat diuraikan dalam beberapa bagian sebagai berikut: *Driver Motor DC* dengan metode logika TTL (0 dan 1) atau *High* dan *Low* hanya dapat mengendalikan arah putar motor DC dalam 2 arah tanpa pengendalian kecepatan putaran (kepatan maksimum). untuk mengendalikan motor DC dalam 2 arah dengan rangkaian *driver motor dc h-bridge* diatas konfigurasi kontrol pada jalur *input* adalah dengan memberikan *input* berupa logika TTL ke jalur *input* A dan B. Untuk mengendalikan arah putar searah jarum jam adalah dengan memberikan logika TTL 1 (*high*) pada jalur *input* A dan logika TTL 0 (*low*) pada jalur *input* B. Untuk mengendalikan arah putar berlawanan arah jarum jam adalah dengan memberikan logika TTL 1 (*high*) pada jalur *input* B dan logika TTL 0 (*low*) pada jalur *input* A.

U. Motor Servo

Motor servo adalah sebuah motor DC dengan sistem umpan balik tertutup di mana posisi rotor-nya akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri dari sebuah motor DC, serangkaian *gear*, potensiometer, dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel *motor servo*. Gambar 49 menunjukkan bentuk hardware dari motor servo.



Gambar 49. Motor Servo

(Sumber : <http://zoniaelektro.net/motor-servo/>)

1. Keunggulan dari penggunaan motor servo adalah :
 - a. Tidak bergetar dan tidak ber-resonansi saat beroperasi.
 - b. Daya yang dihasilkan sebanding dengan ukuran dan berat motor.
 - c. Penggunaan arus listrik sebanding dengan beban yang diberikan.
 - d. Resolusi dan akurasi dapat diubah dengan hanya mengganti *encoder* yang dipakai.
 - e. Tidak berisik saat beroperasi dengan kecepatan tinggi.

2. Jenis-jenis motor servo:

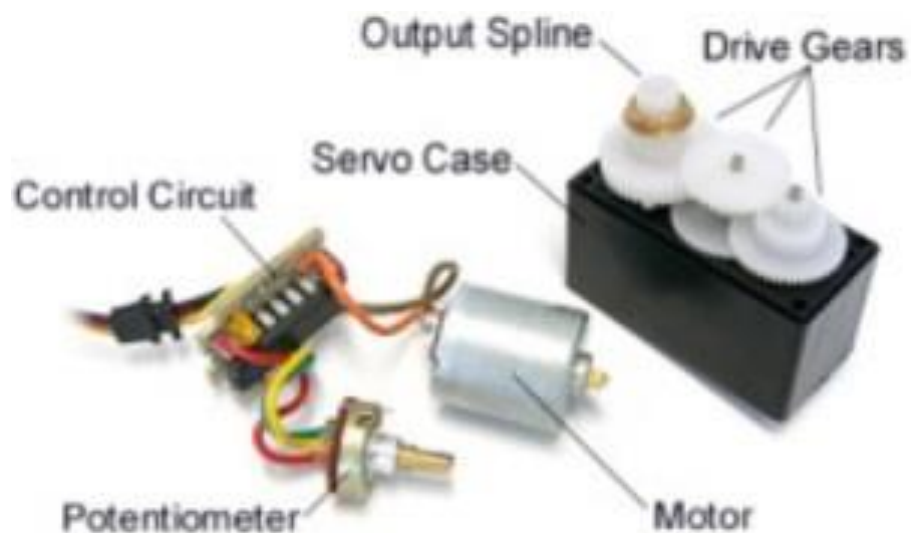
a. Motor Servo *Standart* 180°

Motor servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180°.

b. Motor Servo *Continous*

Motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu).

Motor servo dapat dimanfaatkan pada pembuatan robot, salah satunya sebagai penggerak kaki robot. Motor servo dipilih sebagai penggerak pada kaki robot karena motor servo memiliki tenaga atau torsi yang besar, sehingga dapat menggerakkan kaki robot dengan beban yang cukup berat. Pada umumnya motor servo yang digunakan sebagai penggerak pada robot adalah motor servo 180°.

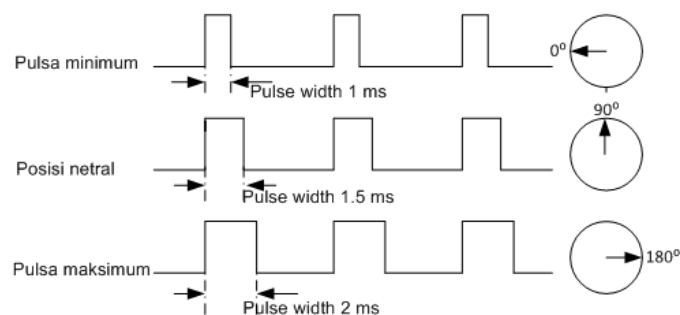


Gambar 50. Komponen Penyusun Motor Servo

(Sumber : <http://zoniaelektro.net/motor-servo/>)

Gambar 50 menunjukkan komponen penyusun dari motor servo. Tiap komponen pada motor servo diatas masing-masing memiliki fungsi sebagai *controler*, *driver*, *sensor*, *gearbox* dan aktuator. Pada gambar diatas terlihat beberapa bagian komponen motor servo. Motor pada sebuah motor servo adalah motor DC yang dikendalikan oleh bagian *controler*, kemudian komponen yang berfungsi sebagai sensor adalah *potensiometer* yang terhubung pada sistem *gearbox* pada motor servo.

Untuk menjalankan atau mengendalikan motor servo berbeda dengan motor DC. Karena untuk mengedalikan motor servo perlu diberikan sumber tegangan dan sinyal kontrol. Besarnya sumber tegangan tergantung dari spesifikasi motor servo yang digunakan. Sedangkan untuk mengendalikan putaran motor servo dilakukan dengan mengirimkan pulsa kontrol dengan frekuensi 50 Hz dengan periode 20 ms dan *duty cycle* yang berbeda. Dimana untuk menggerakkan motor servo sebesar 90° diperlukan pulsa dengan ton *duty cycle* pulsa positif 1,5ms dan unjtuk bergerak sebesar 180° diperlukan lebar pulsa 2ms. Gambar 51 menunjukkan bentuk pulsa kontrol motor servo dimaksud.



Gambar 51. Sinyal Pulsa untuk mengendalikan Motor Servo

(Sumber : <http://zoniaelektro.net/motor-servo/>)

V. Sensor Berat

1. *Loadcell*

Loadcell merupakan sensor timbangan yang bekerja secara mekanis, dimana *Loadcell* menggunakan prinsip tekanan yang memanfaatkan *strain gauge* sebagai pengindera (sensor). *Strain gauge* adalah sebuah *transduser* pasif yang merubah suatu pergeseran mekanis menjadi perubahan tahanan. Perubahan ini kemudiandiukur dengan jembatan *Wheatsone* dimana tegangan keluaran dijadikan referensi beban yang diterima *Loadcell*. Untuk penampakan hardware dari sensor *Loadcell* dapat dilihat pada gambar 52.



Gambar 52. Sensor *Loadcell*

(Sumber : [http:// www.dx.com/p/diy-50kg-body-load-cell-weighing-sensor-resistance-strain-half-bridge-original-for-diy-arduino-etc-423485#.WQWxytqGPIU](http://www.dx.com/p/diy-50kg-body-load-cell-weighing-sensor-resistance-strain-half-bridge-original-for-diy-arduino-etc-423485#.WQWxytqGPIU))

Prinsip kerja dari *Loadcell* adalah, ketika bagian lain yang lebih *elastic* mendapat tekanan, maka pada sisi lain akan mengalami perubahan regangan

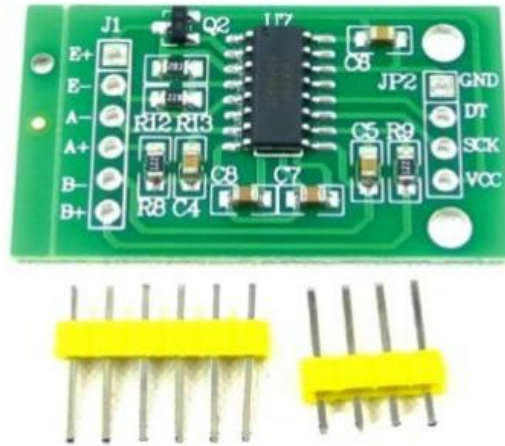
yang sesuai dengan yang dihasilkan oleh *strainingauge*, hal ini terjadi karena ada gaya yang seakan melawan pada sisi lainnya. Perubahan nilai resistansi yang diakibatkan oleh perubahan gaya diubah menjadi nilai tegangan oleh rangkaian pengukuran yang ada. Dan berat dari objek yang diukur dapat diketahui dengan mengukur besarnya nilai tegangan yang timbul. Sel beban (*Loadcell*) terdiri dari satu buah strain gauge atau lebih, yang ditempelkan pada batang atau cincin logam. Sel beban dikalibrasikan oleh pabrikan yang bersangkutan. Piranti ini dirancang untuk mengukur gaya tekanan mekanis, gaya pemampatan (kompresi), atau gaya puntir yang bekerja pada sebuah objek. Ketika batang atau cincin logam piranti ini berada dibawah tekanan, tegangan yang timbul pada terminal-terminalnya yang dapat dijadikan rujukan untuk mengukur besarnya gaya.

2. IC HX711

HX711 *24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales* hx711 adalah sebuah komponen terintegrasi dari perusahaan "AVIA SEMICONDUCTOR" . HX711 presisi 24-bit *analog-to-digital converter* (ADC) yang di desain untuk sensor timbangan digital (*weight scales*) dan industrial *control* aplikasi yang terkoneksi dengan sensor jembatan (*bridge sensor*).

HX711 adalah modul timbangan, yang memiliki prinsip kerja mengkonversi perubahan yang terukur dalam perubahan resistansi dan mengkonversinya ke dalam besaran tegangan melalui rangkaian yang ada. Modul melakukan komunikasi dengan komputer/mikrokontroler melalui

TTL232. Untuk penampakan hardware dari IC HX711 dapat dilihat pada gambar 53.



Gambar 53. IC HX711

(Sumber :

https://www.makefab.com/index.php?route=product/product&product_id=174)

Struktur yang sederhana, mudah dalam penggunaan, hasil yang stabil dan reliable, memiliki sensitivitas tinggi, dan mampu mengukur perubahan dengan cepat. Fitur- fitur dari IC HX711 adalah sebagai berikut:

- a. *Differential input voltage: $\pm 40\text{mV}$ (Full-scale differential input voltage is $\pm 40\text{mV}$)*
- b. *Data accuracy: 24 bit (24 bit A / D converter chip.)*
- c. *Refresh frequency: 80 Hz*
- d. *Operating Voltage : 5V DC*
- e. *Size: 38mm*21mm*10mm*

W. Sensor Switch

1. Proximity Sensor

Sensor *proximity* adalah suatu jenis sensor yang akan aktif bila ada benda atau objek tertentu didekatkan, yang memiliki sifat kerja secara tidak langsung membutuhkan kontak langsung dengan objeknya. Terdapat dua macam sensor *proximity* yang dibedakan berdasarkan jenis objeknya yaitu:

a. Sensor *Proximity Induktive*

- 1) Untuk mendeteksi benda objek yang dapat menghasilkan “*eddy current*” bila dikenai medan magnet. Objek yang seperti ini adalah benda logam pada umumnya.
- 2) Prinsip kerjanya seperti *transformator*, kondisi pada kumparan sekunder akan mempengaruhi kondisi kumparan primer. Kumparan primernya pada *proximity* adalah ‘*detecting coil*’ yang diletakkan pada permukaan deteksi, sedang objek seperti kumparan sekunder yang terminalnya terhubung singkat, sehingga bila *detecting coil* diisolasikan, bila objek didekatkan maka seolah seperti beban yang ditambah pada kumparan sekunder dan akan mengecilkan amplitude osilasi pada sisi primer. Untuk penampakan hardware dari sensor *proximity induktif* dapat dilihat pada gambar 54.



Gambar 54. *Proximity* Induktif

(Sumber : <http://wonderfulengineering.com/10-best-proximity-sensor-modules-for-electronics-project/>)

b. Sensor *Proximity Capacitive*

- 1) Untuk mendeteksi benda objek yang mudah terpolarisasi muatan elektrostatis di dalamnya bila dikenai medan listrik statis. Objek seperti ini adalah isolator dengan konstanta dielektrikum tinggi dan konduktor.
- 2) Prinsip kerjanya memanfaatkan efek polarisasi muatan. Bila dua penghantar yang saling berhadapan dan tak saling bersentuhan dihubungkan pada kutub yang berlawanan dari sebuah sumber tegangan, maka pada permukaan penghantar yang berhadapan tersebut akan terkonsentrasi muatan listrik statis yang berlawanan jenis dan sama besar. Bila jaraknya kemudian diubah (misal didekatkan) maka kerapatan muatannya akan berubah (misal membesar). Energi untuk menambah kerapatan muatan diambil dari sumber tegangan tersebut, sehingga perubahan kerapatan muatan

menyebabkan perubahan arus dari sumber. Untuk penampakan hardware dari sensor *proximity kapasitif* dapat dilihat pada gambar 55.



Gambar 55. *Proximity* Kapasitif

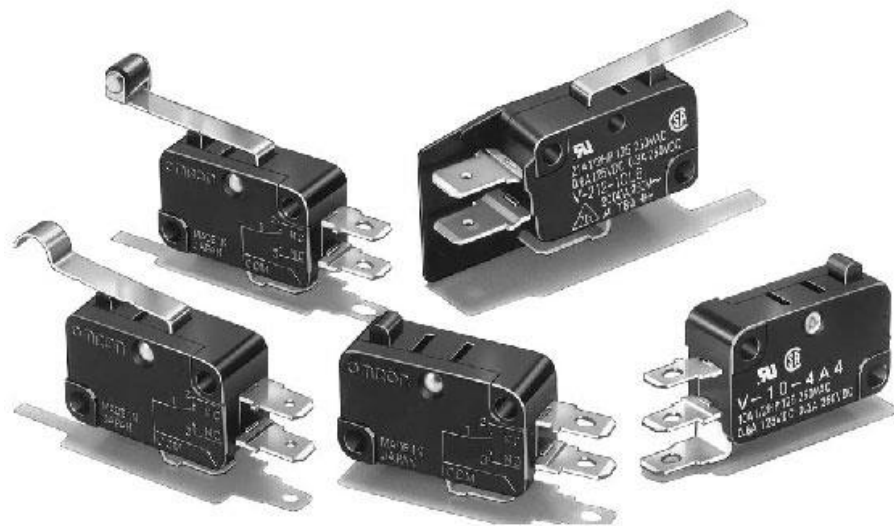
(Sumber : <http://wonderfulengineering.com/10-best-proximity-sensor-modules-for-electronics-project/>)

2. *Limit switch*

Saklar merupakan komponen listrik yang mengontrol arus listrik dengan cara memutus atau menghubungkan serta dapat digunakan untuk merubah atau memindahkan arus listrik secara mekanik. Kontruksi saklar sangat sederhana, yaitu menghubungkan dua permukaan dengan bantuan batang handel. Saklar diisolasi untuk menahan tegangan dari arus listrik, selain itu saklar memiliki kontak-kontak tertutup sehingga aliran listrik dapat mengalir dinamakan *normally close* (NC). Sedangkan jika kontak terbuka sehingga aliran listrik mengalir dinamakan terminal *normally open* (NO). Tidak semua saklar memiliki terminal NC dan NO, Saklar ON/OFF bisaanya digunakan untuk menghidupkan suatu mesin atau proses pada

suatu industri. Sedangkan *Switch* adalah salah satu peralatan yang dapat membuka atau menutup, dan oleh karenanya dalam dunia elektronika akan dapat digunakan untuk mengatur apakah suatu arus hendak dialirkan atau tidak.

Push button adalah suatu switch dimana tekanan harus tetap diberikan bila ingin switch tersebut “on”. Sebuah *limit switch* adalah suatu *push button* switch yang dipasang pada suatu kedudukan, dimana switch ini akan diaktifkan oleh kontak fisik dengan benda yang bergerak. Untuk penampakan hardware dari sensor *limit switch* dapat dilihat pada gambar 56.



Gambar 56. *Limit switch*

(Sumber : <http://gururu.org/guru-berbagi/pengendali-lampu-pijar-menggunakan-limit-switch/>)

Ada dua istilah teknik dalam *limit switch*:

a. *Normally Open*

Kondisi dimana *limit switch* / sensor pada saat *switch* belum aktif (belum mendapat gaya tekan dari luar) adalah “OFF” atau kontak berada pada kondisi terbuka.

b. *Normally Cosed*

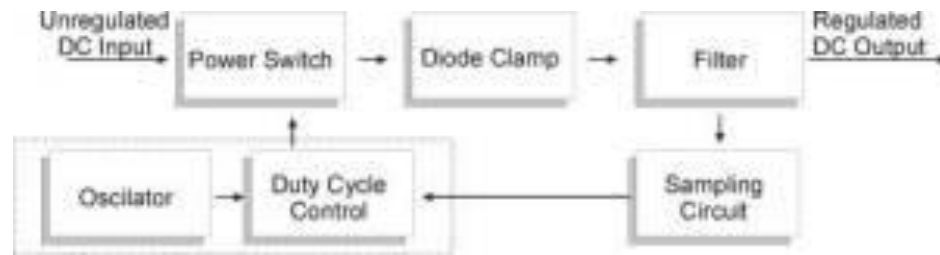
Kondisi dimana *limit switch* / sensor pada saat *switch* belum aktif (belum mendapat gaya tekan dari luar) adalah “ON” atau kontak berada pada posisi terbuka.

X. Catu Daya

1. *Switching Power Supply Adaptor*

Switching power supllly merupakan sebuah desain *power supply* dengan efisiensi daya yang baik. Saat ini perlatan elektronika yang menggunakan adaptor semakin banyak dan semakin beraneka ragam. Mulai dari peralatan elektronik yang murah seperti radio sampai dengan *handphone*. *Power supply* dengan regulasi *switching* ini lebih dikenal sebagai *power supply switching*. Kelebihan *power supply switching* adalah efisiensi daya yang besar sampai sekitar 83% jika dibandingkan dengan *power supply* dengan regulasi bisaa yang menggunakan LM78xx.

Efisiensi yang rendah pada regulator LM78xx dikarenakan kelebihan tegangan *input regulator* akan dirubah menjadi panas sehingga sebagian besar daya *input* akan hilang karena dirubah menjadi panas tersebut. Bagaimanapun juga semua regulator harus mendapatkan tegangan *input* yang lebih tinggi daripada tegangan regulasi *output* untuk mendapatkan tegangan yang teregulasi. Gambar 57 menunjukkan skema blok diagram dari *switching regulator*.



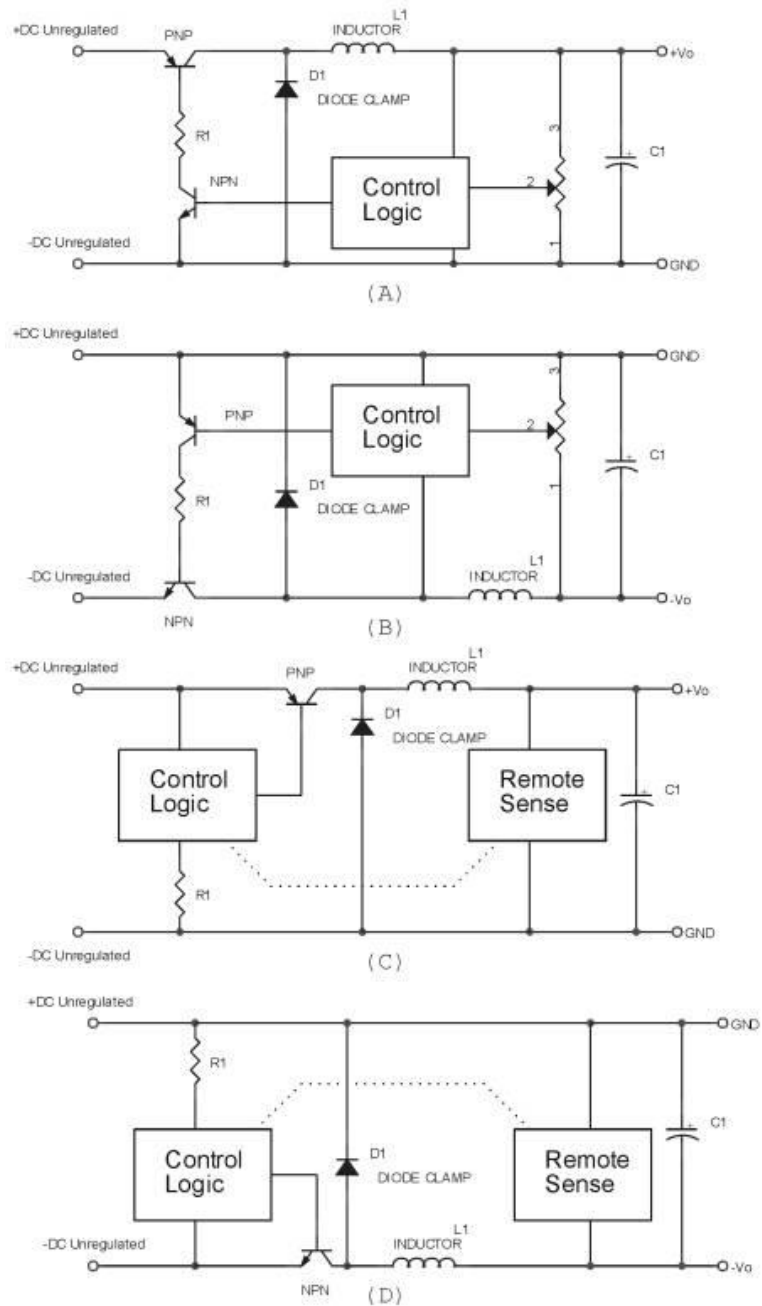
Gambar 57. Blok Diagram *Switching Regulator*

(Sumber : <http://zoniaelektro.net/switching-power-supply/>)

Ada beberapa variasi dari rangkaian *regulator switching*. Perbedaannya adalah pada posisi transistor *switching*-nya. Variasi *regulator switching* tersebut dapat dilihat pada gambar dibawah.

Transistor seri merupakan transistor yang diseri antara tegangan sumber (+DC Unregulated) dan tegangan output regulasi (+Vo). Untuk rangkaian a dan b pada gambar 58 dibawah cocok untuk rangkaian kontrol tegangan teregulasi pada industri karena rangkaian kontrolnya terpisah/terisolasi dengan transistor serinya. Bisaanya antara rangkaian kontrol dengan transistor serinya dipisahkan dengan menggunakan optoisolator (MOCxx atau 4N3x).

Pada rangkaian a dan b pada gambar 58 dibawah, rangkaian kontrolnya mendapatkan tegangan dari *output* tegangan teregulasi sehingga rangkaian tidak akan '*start*' jika tidak diberi tegangan awal. Sedangkan pada rangkaian 2c dan 2d rangkaian kontrolnya mendapatkan tegangan dari +DC *Unregulated* sehingga akan tetap bekerja walaupun terjadi kerusakan/kesalahan pada *Remote Sense* atau Induktor yang menyebabkan tegangan *output* regulasi menjadi nol. Gambar 58 menunjukkan beberapa variasi dari *Switching Regulator* Pada *Switching Power Supply*.



Gambar 58. Variasi *Switching Regulator* Pada *Switching Power Supply*

(Sumber : <http://zoniaelektro.net/switching-power-supply/>)

a. *Filter Input* dan *Penyearah Input Switching Power Supply*

Penyearah input dan filter input terdiri dari penyearah bridge (*full wave rectifier*) dan sebuah filter kapasitor. Untuk meningkatkan efisiensi

dari regulasi maka resistor seri tidak digunakan. Perlu diperhatikan dalam memilih dioda *bridge* yang digunakan karena terdapat arus ‘*surge*’ yang besarnya sampai kira-kira 12A. Arus ‘*surge*’ merupakan arus pengisian kapasitor pada saat rangkaian regulator ini dihidupkan pertama kali. Arus ‘*surge*’ ini menjadi besar karena tidak terdapat resistor seri. Rangkaian penyearah dan filter input ini akan menghasilkan tegangan DC yang tidak teregulasi.

b. *Output Filter Switching Power Supply*

Rangkaian *filter output* tidak terlalu rumit. Rangkaian *filter output* hanya terdiri dari induktor (L) dan kapasitor (C). Nilai induktor dan nilai kapasitor yang digunakan dapat ditentukan dengan menggunakan persamaan :

$$L = \frac{(V_{in} - V_o)V_o}{[(0.05 \times V_{in})(I_o \times f)} \quad C = \frac{(V_{in} - V_o)V_o}{[(2L \times f^2)(V_{in} \times v_o)}$$

Dimana :

v_o = tegangan *ripple* yang diinginkan.

V_o = tegangan regulasi *output*.

V_{in} = tegangan DC tak teregulasi.

f = frekuensi *switching*.

Sebuah rangkaian regulator yang baik harus mempunyai tegangan *ripple* harus sekecil mungkin. Tegangan *ripple* harus dalam level puluhan mV bahkan lebih kecil. Untuk nilai kapasitor yang digunakan biasanya menggunakan 2 kali nilai yang didapatkan dari persamaan di atas karena Faktor disipasi dari kapasitor elektrolit untuk frekuensi tinggi tidak

terlalu baik. Dapat juga digunakan kapasitor tantalum dengan nilai sedikit di atas nilai yang dihasilkan oleh persamaan di atas. Selain itu *filter output* juga berfungsi sebagai *filter* adanya tegangan spike yang ditimbulkan oleh *switching transistor* (kondisi terburuk) agar tidak sampai ke peralatan elektronik (beban).

Sehingga di dalam mendisain sebuah *regulator switching* diperlukan parameter-parameter :

- 1) Tegangan *input* tak teregulasi.
- 2) Tegangan *output* teregulasi yang diinginkan.
- 3) Frekuensi kerja dari *switching transistor*.
- 4) Arus output dari *regulator switching*.
- 5) Tegangan *ripple output* teregulasi..

Selain *bandwidth* dari *transistor switching*, arus kolektor (I_c) dan tegangan kolektor-emitor (V_{CE}) juga perlu diperhatikan dalam proses disain regulator *switching* ini. Arus kolektor (I_c) akan mempengaruhi besarnya arus *output* yang dapat disuplai oleh regulator *switching* dalam kondisi normal. Sedangkan tegangan kolektor-emitor (V_{CE}) akan mempengaruhi tegangan *input* (tegangan DC tak teregulasi) yang dapat diterima oleh *transistor switching* tersebut.

2. Baterai Lipo

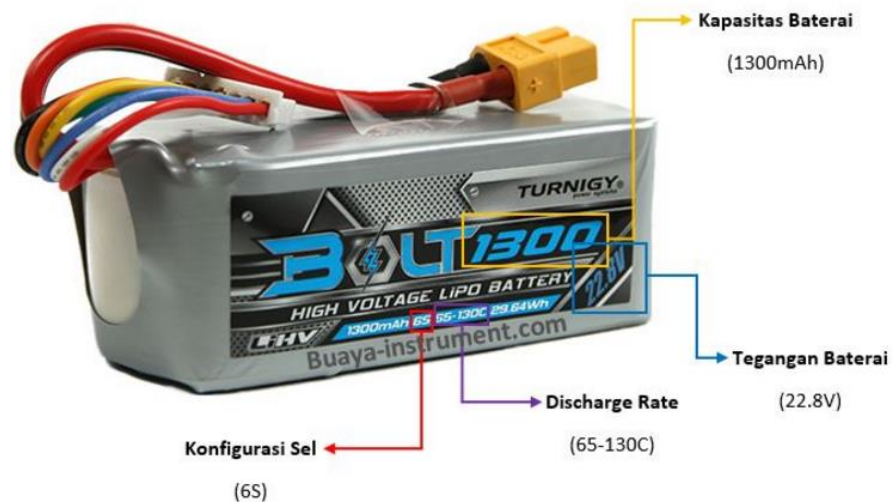
Baterai Lithium Polimer atau bisa disebut dengan *LiPo* adalah salah satu jenis baterai yang sering digunakan dalam dunia RC. Baterai ini merupakan Baterai tercanggih dan paling maju dalam dunia Baterai saat ini.

Keunggulan utamanya adalah *Ratio Power to Weight* nya yang memungkinkan baterai dicetak sesuai dengan keinginan.

Baterai Lipo didasarkan pada *Lithium Polymer* kimia yang memungkinkan baterai ini memiliki kepadatan energi yang sangat tinggi dibandingkan dengan jenis lain dari baterai. Sebuah baterai dengan kepadatan energi yang lebih tinggi akan mampu menahan lebih banyak energi dibandingkan dengan baterai lain dari berat yang sama, itu sebabnya baterai Lipo bisaanya digunakan untuk RC pesawat dan drone.

Baterai LiPo tidak menggunakan cairan sebagai elektrolit melainkan menggunakan elektrolit polimer kering yang berbentuk seperti lapisan plastik film tipis. Lapisan film ini disusun berlapis-lapis diantara anoda dan katoda yang mengakibatkan pertukaran ion. Dengan metode ini baterai LiPo dapat dibuat dalam berbagai bentuk dan ukuran. Diluar dari kelebihan arsitektur baterai LiPo, terdapat juga kekurangan-nya, yaitu lemahnya aliran pertukaran ion yang terjadi melalui elektrolit polimer kering. Hal ini menyebabkan penurunan pada charging dan discharging rate.

Ada jenis lain dari baterai seperti *Nickel Cadmium* (NiCad) atau *Nickel-Metal Hydride* (Nimh), namun ini dilihat sebagai teknologi baterai yang lebih tua dan tidak benar-benar digunakan lagi karena mereka sangat berat dan tidak mampu menampung banyak energi dibandingkan dengan LiPo. Gambar 59 menunjukkan penampakan dari baterai lipo..



Gambar 59. Baterai Lipo

(Sumber : <http://buaya-instrument.com/blog-buaya-instrument/cara-menggunakan-baterai-lipo>)

a. Sel (*Cells*)

Baterai dibangun dari sel-sel persegi panjang yang dihubungkan bersama untuk membentuk baterai. Sebuah sel yang dapat dianggap sebagai baterai itu sendiri, memiliki tegangan 3.7 volt per sel . Dengan menghubungkan lebih dari satu dalam seri, tegangan dapat meningkat menjadi 7.4V untuk baterai 2 sel, 11.1V untuk baterai 3 cell dan sebagainya. Dengan menghubungkan lagi baterai secara paralel kapasitas dapat ditingkatkan.

Sering kali kita melihat label di baterai yang disimbolkan dengan "S". Disini "S" berarti sel yang dimiliki sebuah paket baterai (*battery pack*). Sementara bilangan yang berada didepan simbol menandakan jumlah sel dan bisaanya berkisar antar 2-6S (meskipun kadang ada yang mencapai 10S). Kadang anda juga pernah melihat angka seperti 3S2P,

yang berarti baterai sebagai 3 sel (3S) dihubungkan secara seri, dan ada 2 sel terhubung secara paralel (2P), memberikan sejumlah total 6 individu sel di baterai. Berikut adalah beberapa contoh notasi baterai LiPo.

- 1) 3.7 volt battery = 1 cell x 3.7 volts
- 2) 7.4 volt battery = 2 cells x 3.7 volts (2S)
- 3) 11.1 volt battery = 3 cells x 3.7 volts (3S)
- 4) 14.8 volt battery = 4 cells x 3.7 volts (4S)
- 5) 18.5 volt battery = 5 cells x 3.7 volts (5S)
- 6) 22.2 volt battery = 6 cells x 3.7 volts (6S)

b. Tegangan (*Voltage*)

Setiap sel dalam baterai Lipo Anda memiliki tegangan minimum 3.6V dan tegangan maksimum untuk sel lipo adalah 4.2V. Jika tegangan dari setiap sel dalam baterai Lipo Anda melampaui kisaran ini, dapat menyebabkan bahan kimia menjadi tidak stabil yang dapat menyebabkan Lipo cepat gembung, kebakaran dan memicu ledakkan. Jadi untuk alasan ini pengisi daya baterai Lipo dirancang untuk memastikan Anda hanya men-charger setiap sel untuk 4.2V.

c. Kapasitas (*Capacity*)

Kapasitas baterai menunjukkan seberapa banyak energi yang dapat disimpan oleh sebuah baterai dan diindikasikan dalam miliampere hours (mAh). Semakin besar jumlah mAh, maka kapasitas baterai yang dimiliki semakin besar, sehingga dapat menjalankan beban lebih lama, atau dengan kata lain seberapa banyak beban yang dapat diberikan kepada

sebuah baterai selama 1 jam, dimana setelah 1 jam baterai akan benar-benar habis.

d. *Discharge Rate*

Discharge Rate adalah spesifikasi yang sangat penting untuk memastikan ketika membeli baterai. Jumlah ini, juga dikenal sebagai C rating baterai. Discharge rate bisa disimbolkan dengan "C" merupakan notasi yang menyatakan seberapa cepat sebuah baterai untuk dapat dikosongkan (discharge) secara aman. Sesuai dengan penjelasan diatas bahwa energi listrik pada baterai LiPo berasal dari pertukaran ion dari anoda ke katoda. Semakin cepat pertukaran ion yang dapat terjadi maka berarti semakin nilai dari "C". Sebuah baterai dengan discharge rate 15C berarti baterai tersebut dapat di discharge 15 kali dari kapasitas baterai sebenarnya. begitu juga 20C berarti 20 kali dan sebagainya.

Y. Media Penyimpanan

1. *Micro SD Card*

Micro SD adalah kartu memori non-volatile yang dikembangkan oleh SD Card Association yang digunakan dalam perangkat portabel. Keluarga *micro SD* yang lain terbagi menjadi SDSC yang kapasitas maksimum resminya sekitar 2GB, meskipun beberapa ada yang sampai 4GB. SDHC (*High Capacity*) memiliki kapasitas dari 4GB sampai 32GB. Gambar 60 menunjukkan bentuk dari *micro SD* dan konfigurasiya.



Gambar 60. Bentuk fisik micro SD dan pin mapping

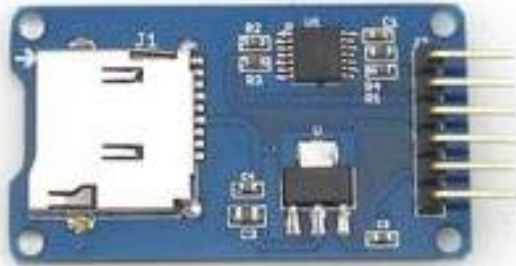
(Sumber : <http://www.indo-ware.com/produk-2735-micro-sd-card-modul-spi-antarmuka-mini-card-reader-tf.html>)

Layaknya kartu *flash* lainnya, *Micro SD* sudah terformat dengan sistem *file* sebagai FAT16, SDHC sebagai FAT32, sedangkan SDXC sebagai ExFAT. Dimanapun FAT16 dan FAT32 memungkinkan untuk dapat diakses melalui semua perangkat *host* pembaca SD. Pemeliharaan FAT standar dapat digunakan untuk memperbaiki atau mengambil data yang rusak dan beberapa dapat memulihkan *file* yang dihapus. Namun karena teknologi ini muncul sebagai *drive removable hard* maka bisa diformat ulang untuk setiap sistem *file* yang didukung oleh sistem operasi. Juga, bisa ditanamkan sistem operasi seperti *USB Live* yang bisa memulihkan host komputer dari *Flash Media Reader*.

2. *Micro SD Card Adapter*

Modul (*Micro SD Card Adapter*) adalah modul pembaca kartu *Micro SD*, melalui sistem *file* dan SPI antarmuka *driver*, MCU untuk melengkapi sistem *file* untuk membaca dan menulis kartu *Micro SD*. Pengguna Arduino dapat menggunakan Arduino IDE dilengkapi dengan kartu SD untuk

menyelesaikan inisialisasi kartu perpustakaan dan membaca menulis. Gambar 61 menunjukkan bentuk dari *micro SD adapter*.



Gambar 61. *Micro SD Card Adapter*

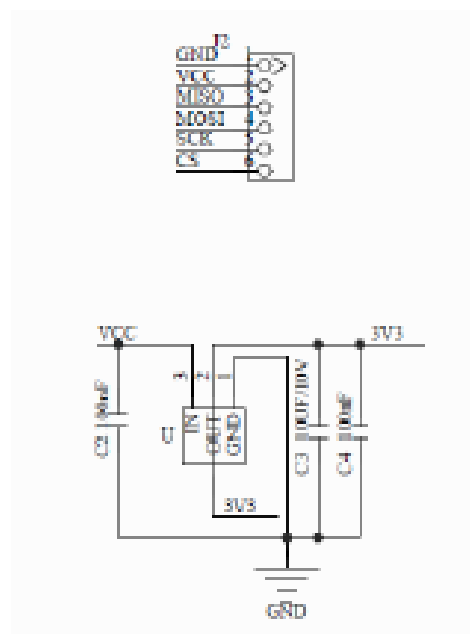
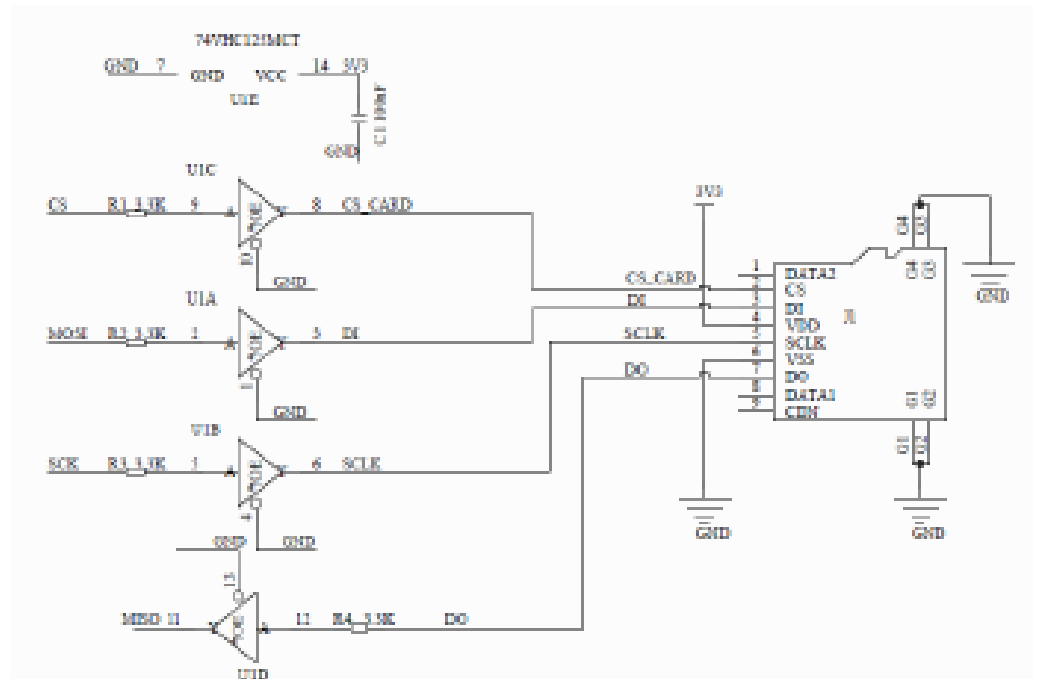
(Sumber : <http://www.indo-ware.com/produk-2735-micro-sd-card-modul-spi-antarmuka-mini-card-reader-tf.html>)

Fitur modul adalah sebagai berikut:

- a. Mendukung kartu Micro SD, kartu Micro SDHC (kartu kecepatan tinggi)
- b. tingkat konversi papan sirkuit yang antarmuka level untuk 5V atau 3.3V
- c. *power supply* adalah 4.5V ~ 5.5V, *regulator* tegangan 3.3V papan sirkuit
- d. adalah komunikasi antarmuka SPI antarmuka standar
- e. empat (4) M2 lubang sekrup posisi untuk kemudahan instalasi
- f. *Control Interface* sebanyak enam pin (GND, VCC, MISO, MOSI, SCK, CS), GND ke *ground*, VCC adalah *power supply*, MISO, MOSI, SCK adalah SPI bus, CS adalah chip pilih pin sinyal.

- g. 3.3V rangkaian *regulator* tegangan. Keluaran *regulator* LDO adalah chip yang mempunyai tegangan tingkat konverter 3.3V, untuk mensuplai *Micro SD Card*.
- h. *Micro SD Card* konektor
- i. Lubang *Positioning* : empat (4) M2 sekrup posisi diameter lubang adalah 2.2mm, posisi modul mudah untuk di pasang, untuk mencapai kombinasi antar-modul.

Gambar 62 menunjukkan skema *micro SD card Adapter*.



Gambar 62. Skema *micro SD Card Adapter*

(Sumber : <http://www.indo-ware.com/produk-2735-micro-sd-card-modul-spi-antarmuka-mini-card-reader-tf.html>)

BAB III

KONSEP RANCANGAN

Alat pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* pada Atlet menggunakan metode rancang bangun. Secara urut metode tersebut adalah analisa kebutuhan yang diperlukan. Kebutuhan tersebut kemudian diidentifikasi untuk mendapatkan komponen secara spesifik, selanjutnya dilakukan perancangan perangkat keras dan perangkat lunak, dilanjutkan dengan pembuatan serta pengujian alat.

A. Analisa Kebutuhan

Perancangan alat pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* pada Atlet ini guna menunjang prestasi atlet. Tahap ini dibutuhkan untuk menganalisa kebutuhan *Hardware*, *Software*, dan Sistem. Adapun Analisa kebutuhan sebagai berikut:

1. Dibutuhkan rangka dengan bahan yang ringan, kuat, tahan lama. Serta sesuai dengan mekanisme kerja alat.
2. Dibutuhkan box sebagai wadah kontrol dari sistem yang terintegrasi.
3. Dibutuhkan beberapa sensor yang dapat diterapkan pada mekanisme pengukuran *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction*.
4. Dibutuhkan beberapa komponen pengendali untuk mengendalikan sistem pengukuran *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction*.
5. Dibutuhkan Catu Daya untuk mensuplai tegangan pada seluruh sistem yang terintegrasi.
6. Dibutuhkan media penampil untuk menampilkan kondisi saat sistem sedang berjalan.

7. Dibutuhkan rangkaian driver pengendali putaran pada mekanisme penggerak motor DC.
8. Dibutuhkan media komunikasi data Serial secara *wireless*.
9. Dibutuhkan media penyimpanan untuk menyimpan data hasil pengukuran.
10. Dibutuhkan media upload data ke *website* secara *online*.
11. Dibutuhkan *server website* dan *database website*.

B. Identifikasi Kebutuhan

Berdasarkan Analisa kebutuhan di atas, maka diperoleh beberapa kebutuhan, terdiri komponen utama dan komponen habis pakai. Komponen utama adalah seperti yang tertera pada tabel 18.

Tabel 18. Identifikasi Kebutuhan

No	Nama Komponen	Spesifikasi	Jumlah	Keterangan
1.	Sensor Berat	<i>Loadcell</i>	4 buah	<i>Input</i>
2.	Sensor Jarak	Ultrasonik SRF04	5 buah	<i>Input</i>
		SHARP GP2Y0A21YK0F	1 buah	<i>Input</i>
3.	Sensor <i>Infrared</i>	<i>Proximity</i>	2 buah	<i>Input</i>
		<i>Opto Coupler</i>	1 buah	<i>Input</i>
4.	Sensor <i>Encoder</i>	<i>Autonics Rotary Encoder E40H8- 3600-3-N-24</i>	1 buah	<i>Input</i>
5.	Sensor Warna	Led dan LDR	1 pasang	<i>Input</i>
6.	Switch	<i>Limit switch Dekko</i>	7 buah	<i>Input</i>
7.	Modul Micro SD	Modul <i>Micro SD Adapter</i>	1 buah	<i>Input</i>
8.	Mikrokontroler	Arduino Nano 328P	2 buah	Proses
		Arduino Mega 2560	1 buah	Proses
		Wemos D1	1 buah	Proses
9.	Motor DC	Motor DC <i>Gearbox</i> 12 V 2 A 400 rpm	2 buah	<i>Output</i>
		<i>Power window</i> 12 V 5 A	1 buah	<i>Output</i>
10.	LCD <i>Display</i>	LCD 16x2	1 buah	<i>Output</i>
		LCD 20x4	1 buah	<i>Output</i>

11.	<i>Driver Motor</i>	H-Bridge Mosfet	1 buah	<i>Output</i>
		H-Bridge Relay	1 buah	<i>Output</i>
12.	Motor Servo	Tower-Pro 180 derajat	1 buah	<i>Output</i>
13.	LED Flash		1 buah	<i>Output</i>
14.	Micro SD	Sandisk 2 Gb	1 buah	<i>Data logger</i>
15.	Bluetooth	HC-05	4 buah	Komunikasi
16.	Converter	Modul LCD to I2C	1 buah	Komunikasi
17.	Baterai	Lipo 3 cell 2200 mAh	1 buah	Catu Daya
		Lipo 2 cell 850 mAh	1 buah	Catu Daya
18.	<i>Power supply</i>	12V 5A Switching	1 buah	Catu Daya
		12V 2A Switching	1 buah	Catu Daya
19.	<i>Step Down</i>	XL4015 DC to DC Step Down Module 5A 75W	3 buah	<i>Step Down</i>
20.	Router	Sierra 4G Wireless Modem Wifi	1 buah	Jaringan Internet

Adapun penjelasan identifikasinya adalah sebagai berikut:

1. Mikrokontroler

a. Arduino Mega (*Master Controller*)

Data dari *Slave 1*, *Slave 2*, dan semua sistem yang terintegrasi akan diolah dalam mikrokontroler *Master* ini dan akan menjadi pusat kendali dari semua perangkat yang sudah terintegrasi.

b. Arduino Nano 1 (*Slave 1 Controller*)

Slave 1 Controller disini difungsikan untuk mengolah data dari sistem pengukuran ketinggian loncatan (*Vertikal jump*) dengan menggunakan sensor ultrasonik dan *proximity* yang nantinya di proses menggunakan mikrokontroler Arduino Nano dan membagikan data hasil pengukuran ke *Master Controller*.

c. Arduino Nano 2 (*Slave 2 Controller*)

Slave 2 Controller disini difungsikan untuk mengolah data dari sistem pengukuran kejauhan loncatan (*Standing board jump*) dengan menggunakan seperangkat *mekanikal range measurement sistem* yang

nantinya di proses menggunakan mikrokontroler Arduino Nano dan membagikan data hasil pengukuran ke *Master Controller*.

d. Wemos D1 (*Slave 3 controller*)

Wemos berfungsi sebagai perangkat yang digunakan untuk membagi data ke jaringan internet. Mikrokontroler ini akan menerima data dari *Master Controller* berupa data data paket yang berisi data-data hasil semua pengukuran. Setelah mendapatkan data maka mikrokontroler ini akan mengirim data dari *Master controller* untuk di unggah dalam *database* secara *online*.

2. Sensor Berat (*Loadcell*)

Sensor berat pada alat ini memanfaatkan dari timbangan digital yang sudah tersedia di pasaran dan memodifikasinya dengan hanya mengambil bagian mekanik dan sensornya saja, lalu mengganti sistem kontrol bawaan timbangan dengan IC HX711 sebagai rangkaian penguat sinyal analog yang didapatkan dari timbangan yang sudah dimodifikasi ini untuk disalurkan ke bagian proses utama yaitu Arduino Mega. Timbangan ini sudah terintegrasi dengan 4 buah sensor *Loadcell* 50 kg.

3. Sensor Jarak (Ultrasonik)

a. Ultrasonik 1 digunakan sebagai pengukur tinggi badan dalam keadaan darurat saja 'Jika sistem pengukuran tinggi badan sudah diluar jangkauan bawahnya'.

b. Ultrasonik 2,3,4 Sensor ini digunakan sebagai pendeteksi keadaan seseorang ketika alat akan dioperasikan 'Apakah sudah ada orang apa belum', Karena alat ini menggunakan sistem kendali otomatis.

- c. Ultrasonik 5 Sensor yang satu ini digunakan pada rangkaian *Slave 1 Controller* sebagai pendeteksi tinggi lompatan (*Vertikal jump*).
4. Sensor *Infrared (Proximity)*
 - a. *Proximity*
 - 1) *Proximity 1* Digunakan pada rangkaian *Slave 1 Controller* sebagai pendeteksi mulai dan berhentinya lompatan (*Vertikal jump*).
 - 2) *Proximity 2* Digunakan pada rangkaian *Slave 2 Controller* sebagai pendeteksi mulai dan berhentinya lompatan (*Standing board jump*).
 - b. Sensor *Infrared (Optocoupler)*

Sensor digunakan pada rangkaian *Slave 2 Controller* sebagai penghitung putaran roda mekanik yang berjalan pada as lurus untuk kemudian dikonversikan sebagai satuan jarak (Cm) untuk mengukur panjang lompatan.
 5. Sensor *Encoder (Autonics Rotary Encoder E40H8-3600-3-N-24)*

Sensor ini digunakan pada rangkaian utama *Master Controller* sebagai penghitung jumlah putaran mekanik gir dan rantai yang berfungsi untuk menggerakkan mekanik pengukur tinggi badan, kemudian dikonversikan sebagai satuan jarak (Cm) untuk mengukur tinggi badan.
 6. Sensor warna (LED dan LDR)

Sensor ini digunakan sebagai pendeteksi batas depan ujung depan dan ujung belakang rel pengukur *standing board jump* dengan cara mendeteksi garis putih yang berada pada ujung depan dan belakang rel, kemudian

dihubungkan dengan *Slave 2 Controller* untuk di proses menjadi data analog.

7. *Limit switch*

- a. *Limit switch* 1 dan 2 digunakan sebagai batas untuk menghentikan putaran motor DC pada sistem pengukur tinggi badan menggunakan tiang pengukur yang digerakkan menggunakan motor DC *power window*.
- b. *Limit switch* 3 digunakan sebagai batas ketika tiang pengukur tinggi badan menyentuh kepala.
- c. *Limit switch* 4,5,6,7 digunakan untuk media pengukuran *Whole Body Reaction*.

8. Motor DC

- a. Motor DC (*Gearbox* 12V 2A 400 rpm) Digunakan pada rangkaian *Slave 2 Controller* sebagai sistem mekanik penggerak dudukan sensor pada as lurus pengukur panjang lompatan (*Standing board jump*).
- b. Motor DC (*Power window* 12V 5A) Digunakan pada rangkaian utama *Master Controller* sebagai sistem mekanik penggerak naik dan turunnya tiang pengukur tinggi badan.

9. Motor Servo

Digunakan pada rangkaian *Master Controller* sebagai dudukan dan penggerak sensor Ultrasonik 2,3,4 untuk mendeteksi keberadaan orang.

10. LCD (*Liquid crystal Display*)

- a. LCD 1 (20x4) Digunakan pada rangkaian utama *Master Controller* sebagai penampil keadaan sistem server utama yang sedang berjalan.

- b. LCD 2 (16x2) Digunakan pada rangkaian *Slave 2 Controller* sebagai penampil keadaan sistem pada *Slave 2*.

11. *Driver Motor*

- a. *Driver Motor (H-bridge Mosfet)* Digunakan pada rangkaian utama *Master Controller* sebagai pengendali putaran motor DC *power window* yang berfungsi sebagai motor penggerak sistem mekanik naik dan turunnya tiang pengukur tinggi badan. Alasan menggunakan Mosfet sebagai *Driver motor* karena *mosfet* ini mempunyai arus kerja yang lumayan besar dan cukup untuk mensuplai arus dari motor *power window*.
- b. *Driver Motor (H-bridge Relay)* Digunakan pada rangkaian *Slave 2 Controller* sebagai pengatur putaran dari motor DC (*Gearbox 400 rpm*) yang berfungsi sebagai sistem mekanik penggerak roda pada as lurus pengukur panjang lompatan (*Standing board jump*). Alasan menggunakan *driver* ini karena untuk menggerakkan sistem mekanik ini tidak memerlukan pengaturan kecepatan, melainkan hanya memerlukan pengaturan arah putaran roda penggeraknya.

12. Modul *Micro SD Adapter*

Digunakan pada rangkaian utama *Master Controller* sebagai interface untuk mengakses *micro SD*.

13. *Micro SD Card*

Digunakan pada rangkaian utama *Master Controller* sebagai media penyimpanan data-data yang diperoleh dari semua sistem yang telah terintegrasi pada *Master Controller* ini.

14. LED *Flash*

Digunakan pada rangkaian utama *Master* controller sebagai tanda dimulainya pengukuran *Whole Body Reaction*, sistem berjalan otomatis.

15. *Converter LCD to I2C*

Digunakan pada rangkaian *Slave 2 Controller* sebagai interface penghubung antara Arduino Nano dengan LCD melalui komunikasi Serial I2C. Menggunakan modul ini dikarenakan pin pada Arduino Nano sudah penuh oleh sensor dan yang lainnya.

16. Baterai

Digunakan pada rangkaian *Slave 1 Controller* dan *Slave 2 Controller* sebagai rangkaian catu daya penyedia tegangan, diambil dari 2 buah *lithium polymer battery 7,8 Volt 800 miliampere dan 11,3 Volt 2200 miliampere* sebagai penyedia tegangan rangkaian *Slave 1* dan *Slave 2*, dalam hal ini menggunakan modul UBEC.

17. *Power supply*

Power supply atau dikenal sebagai Catu Daya hanya digunakan pada rangkaian utama *Master Controller*, Karena pada sistem ini membutuhkan jumlah arus yang lumayan besar. Dalam hal ini menggunakan komponen *power supply switching* dikarenakan bentuk rangkaiannya yang cukup praktis dan efisien tempat.

- a. *Switching 12V 5 A* sebagai sumber tegangan *12 Volt 5 Ampere* hanya untuk mensuplai tegangan *driver* motor untuk menggerakkan motor DC *power window 12V 5A*,

- b. *Switching* 12V 2A digunakan untuk mensuplai tegangan kontroller beserta sensor-sensor yang terintegrasi pada rangkaian *Master Controller*.

18. *Step Down* (XL4015 DC to DC *Step Down* Module 5A 75W)

- a. *Step Down* 1 digunakan pada rangkaian *Master Controller* untuk menurunkan tegangan 12V 2A menjadi 5V 2A untuk mensuplai mikrokontroler dan juga sensor-sensor yang terintegrasi.
- b. *Step Down* 2 digunakan pada rangkaian *Slave* Kontroller 1 untuk menurunkan tegangan baterai Lipo 2 cell dari 7,8V menjadi 5V agar bisa digunakan untuk mensuplai Arduino Nano, sensor Ultrasonik, dan sensor *Proximity*.
- c. *Step Down* 3 digunakan pada rangkaian *Slave* Kontroller 1 untuk menurunkan tegangan baterai Lipo 2 cell dari 7,8V menjadi 5V agar bisa digunakan untuk mensuplai Arduino Nano, sensor Ultrasonik, dan sensor *Proximity*.

19. Modem Wifi

Digunakan untuk mengkoneksikan antara Wemos dengan jaringan internet, supaya bisa mengupload data ke *website* yang sudah dibuat secara *online*.

20. Modul *Bluetooth*

- a. *Bluetooth* HC-05 (1&2)

Digunakan untuk menghubungkan antara rangkaian utama *Master Controller* dengan *Slave 1 Controller* secara *serial wireless*.

- b. *Bluetooth* HC-05 (3&4)

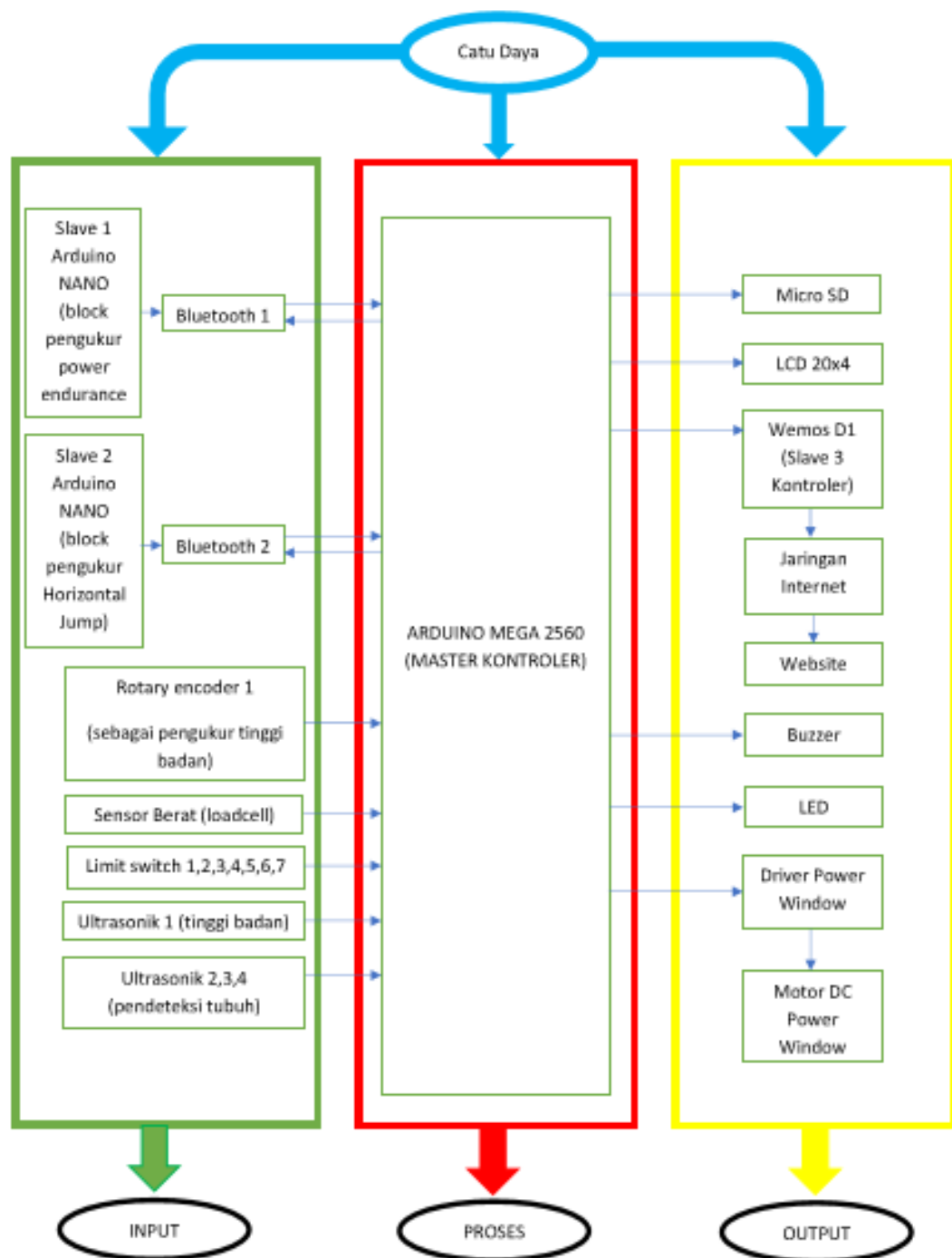
Digunakan untuk menghubungkan antara rangkaian utama *Master Controller* dengan *Slave 2 Controller* secara *serial wireless*.

C. Blok Diagram Sistem

Alat pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* pada Atlet pada tugas akhir ini melibatkan perangkat dari lima pengukuran secara langsung yaitu pengukuran tinggi badan, berat badan, tinggi lompatan, panjang lompatan, dan kecepatan reaksi tubuh. Pengukuran massa tubuh menggunakan timbangan yang sudah dimodifikasi dengan mengganti kontrol utama dari timbangan menggunakan IC HX711 sebagai penguat sinyal analog keluaran dari *Loadcell* untuk kemudian di hubungkan dengan rangkaian utama *Master Controller* Arduino Mega. Pengukuran tinggi badan menggunakan mekanik tiang penyangga setinggi 210 cm dengan cara menggerakkan tiang turun dari posisi default (210 cm) sejajar dengan tubuh seseorang yang akan diukur dan berhenti ketika menyentuh ujung kepala karena ada *limit switch*, kemudian menghitung selisih dari tinggi tiang saat pengukuran dan tinggi tiang default maka akan diketahui tinggi tubuh. Pengukuran tinggi lompatan (*vertical jump*) dan *Power Endurance* dilakukan menggunakan sensor ultrasonik dengan cara membuat rangkaian *Slave 1 Controller* yang terpisah dari rangkaian utama *Master Controller* untuk digunakan sebagai marker pada tubuh seseorang yang melakukan pengukuran dengan cara mengukur panjang gelombang ultrasonik pada saat objek melompat, kemudian diolah menjadi satuan jarak (Cm). Pengukuran panjang lompatan dilakukan pada *Slave 2 Controller* menggunakan mekanik rel seperti

prinsip kerja rumah *catridge* pada printer kertas yang bergerak dari kiri ke kanan, tapi pada sistem ini yang membedakan adalah kalau rumah *catridge* pada printer berfungsi untuk mencetak gambar dengan tinta, kalau pada alat ini difungsikan sebagai pengukur panjang lompatan karena mekanik seperti ini bisa mengukur panjang sesuai rel yang ada, dengan menggunakan sensor *Optocoupler* untuk menghitung jumlah putaran dari roda penggerak menjadi satuan jarak (Cm). Pengukuran kecepatan reaksi tubuh menggunakan sensor *proximity* dengan cara kerja mendeteksi kaki objek pada bidang tumpuan, saat pengukuran dimulai Led Flash akan menyala selama 0,2 detik dan kaki objek akan bergerak menghindari dari tumpuan, seketika itu juga dari saat Led Flash menyala dan kaki objek menghindari dari tumpuan maka akan diketahui berapa lama kecepatan reaksi seseorang itu.

Semua data yang diperoleh dari hasil pengukuran akan di tampilkan pada LCD dan di unggah pada *database website* secara *online* menggunakan perangkat Wemos D1 sebagai *Slave 3 controller*, kemudian dapat ditampilkan pada *website*. Blok diagram sistem dari alat pengukur *power tungkai*, *power endurance*, dan *whole body reaction* pada atlet dapat dilihat pada gambar 63.



Gambar 63. Blok Diagram utama dari alat pengukur *power* tungkai,

Power Endurance, dan *Whole Body Reaction* pada atlet

D. Perancangan Rangkaian

Pada alat pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* pada Atlet ini terdapat tiga buah rangkaian sistem yang terpisah yaitu rangkaian sistem Arduino Mega sebagai sistem utama (*Master Controller*) pengendali semua sensor, aktuator, *Slave 1 Controller*, *Slave 2 Controller*, dan Wemos D1 sebagai (*Slave 3 controller*). Rangkaian Arduino Nano 1 sebagai marker pengukur tinggi lompatan (*Slave 1 Controller*). Rangkaian Arduino Nano 2 sebagai sistem mekanik rel pengukur panjang lompatan (*Slave 2 Controller*).

1. Blok rangkaian Aduino Mega (*Master Controller*)

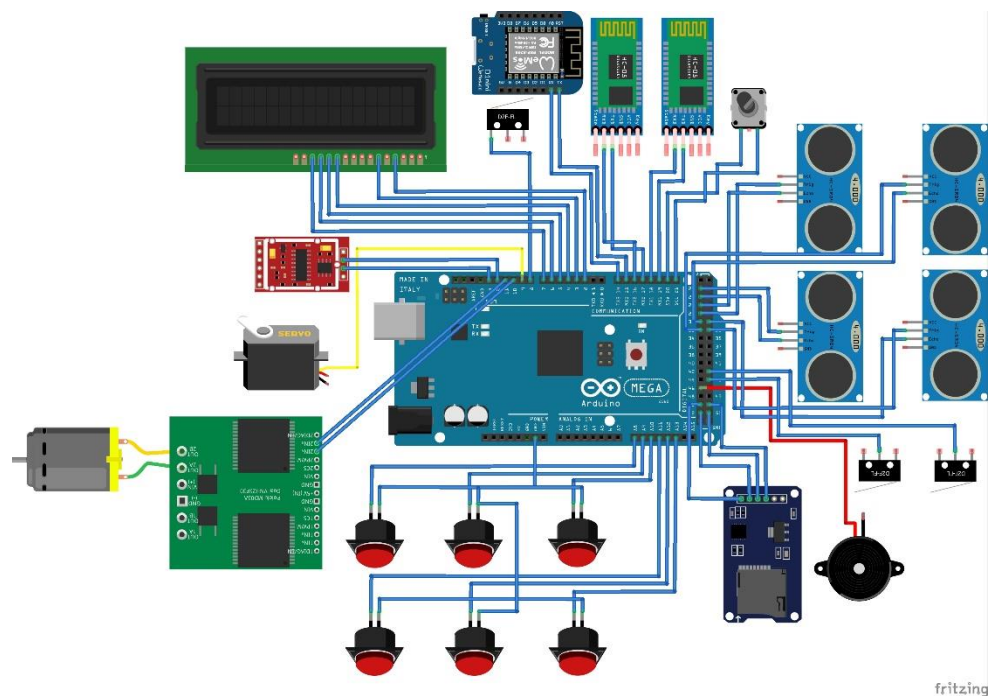
Rangkaian *Master Controller* adalah pusat dari semua sistem yang terintegrasi, mulai dari *Slave 1* kontrolle, *Slave 2 Controller*, wemos D1, sensor-sensor, dan aktuator. Dari sinilah nanti semua proses pengambilan data akan dimulai dan disimpan, untuk kemudian di tampilkan pada *website* menggunakan jaringan internet. Untuk gambar Rangkaian *Master Controller* dapat dilihat pada Gambar 64. Komponen-komponen penyusunnya terdiri dari:

- a. 4 buah sensor ultrasonik SRF04 yang memiliki 4 buah pin (VCC, *Trigger*, *Echo*, GND) dengan konfigurasi pin Arduino Ultrasonik 1 (*trigger*=D22, *echo*=D24), Ultrasonik 2 (*trigger*=D28, *echo*=26), Ultrasonik 3 (*trigger*=32, *echo*=30), Ultrasonik 4 (*trigger*=33, *echo*=31). Pin Vcc keempat Ultrasonik terhubung dengan 5V supply dan pin GND keempat Ultrasonik terhubung dengan GND supply.

- b. 5 buah *button* menu dan 1 buah *button Reset* dengan konfigurasi pin *button* 1 UP (A8), *button* 2 DOWN (A9), *button* 3 OK (A10), *button* 4 SAVE (A11), *button* 5 BACK (A12), dan *button* 6 RESET. Tombol *button* ini dikondisikan sebagai aktif LOW.
- c. 2 buah modul *Bluetooth* HC-05 yang berfungsi modul *Bluetooth* HC-05 (1) sebagai interface ke *Slave 1 Controller* dan modul *Bluetooth* HC-05 (2) sebagai interface ke *Slave 2 Controller*. Modul ini memiliki 4 buah pin (KEY, GND, VCC, RX, TX, EN) dengan konfigurasi pada pin Arduino *Bluetooth* 1 (RX=TX1, TX=RX1), *Bluetooth* 2 (RX=TX2, TX=RX2). Pin Vcc dari kedua modul terhubung dengan 5V supply, dan GND kedua modul terhubung dengan GND supply.
- d. LCD 20x4 dengan konfigurasi pin(RS=D2, E=D3, D4=D4, D5=D5, D6=D6, D7=D7) pin RW terhubung dengan GND Bersama dengan VSS dan Anoda, pin VDD terhubung dengan GND bersama dengan Katoda, dan pin V0 terhubung dengan trimpot 10 k.
- e. Wemos D1 sebagai interface ke jaringan internet dengan konfigurasi (RX wemos = TX3 Arduino, TX wemos = RX3 Arduino) pin GND terhubung dengan GND supply dan pin Vcc terhubung dengan 5V supply.
- f. *Rotary Encoder* sebagai penghitung jumlah putaran motor saat menggerakkan tiang pengukur dan kemudian mengubahnya menjadi satuan jarak (Cm). Sensor ini memiliki pin (VCC, GND, OUT A, OUT B, SWITCH) terhubung dengan Arduino dengan konfigurasi (OUT A

- = pin Interrupt 0(D20), OUT B = pin Interrupt 1(D21)) pin Vcc terhubung dengan 5V supply, dan pin GND terhubung dengan GND supply.
- g. Modul IC HX711 sebagai penguat sinyal Analog pada sensor *Loadcell* memiliki pin (VCC, GND, OUT A, OUT B) terhubung dengan Arduino dengan konfigurasi (OUT A = D12, OUT B = D13) pin VCC terhubung dengan 5V supply, dan pin GND terhubung dengan GND supply.
- h. *Driver* motor mosfet *H-Bridge* untuk mengontrol kecepatan motor menggunakan PWM. *Driver* ini memiliki pin (Vin 12V, GND, IN A, IN B). Terhubung dengan Arduino dengan konfigurasi (IN A= D11, IN B=D10) Vin terhubung dengan 12V supply, dan GND terhubung dengan GND supply.
- i. Motor Servo sebagai pengendali arah dari Ultrasonik 1,2,3 untuk mendeteksi orang. Mempunyai pin (Data, VCC, GND) terhubung dengan Arduino dengan konfigurasi Data = D9 (PWM), VCC terhubung dengan 5V supply, dan GND terhubung dengan GND supply.
- j. Modul *micro SD* sebagai media interface untuk mengakses *micro SD* card. Modul ini memiliki 6 buah pin yaitu (Vcc, Miso, Mosi, SCK, CS, GND). Konfigurasi dengan Arduino (Miso=D54, Mosi=D51, SCK=D52, CS=D53), Vcc terhubung dengan 5V supply, dan GND terhubung dengan GND supply.
- k. 3 buah *Limit switch* masing masing switch berfungsi: *Limit switch 1* sebagai batas bawah tiang pengukur tinggi badan, *Limit switch 2*

sebagai batas atas tiang pengukur tinggi badan, dan *Limit switch 3* sebagai batas saat ujung tiang mengenai kepala. *Limit switch* ini dikondisikan Aktif *LOW* dan Normaly Open. Konfigurasi pada Arduino adalah (Switch 1 = D43, Switch 2 = D45, Switch 3 = D8) dan pin Com dihubungkan dengan GND supply.

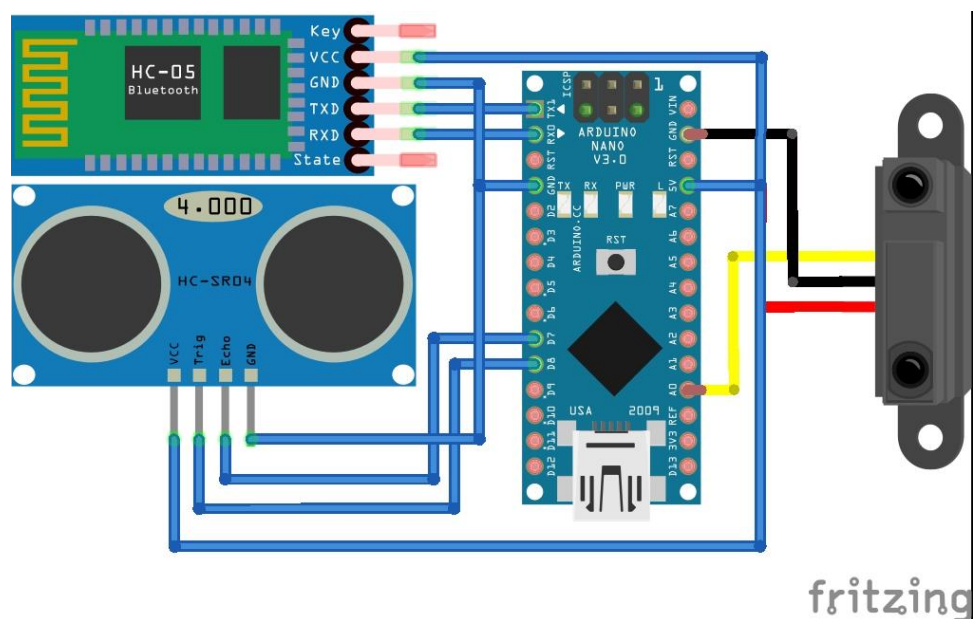


Gambar 64. Blok rangkaian Arduino Mega (*Master Controller*)

2. Blok Rangkaian Arduino Nano 1 (*Slave 1 Controller*)

Rangkaian *Slave 1 Controller* adalah pengatur sistem pengukuran Tinggi Lompatan. Mikrokontroler yang digunakan adalah Arduino NANO V2.8. Untuk gambar Rangkaian *Slave 1 Controller* dapat dilihat pada Gambar 65. Komponen penyusun dan kegunaannya adalah sebagai berikut:

- a. Modul *Bluetooth* HC-05 sebagai sarana penghubung dengan *Master Controller* untuk menerima perintah pengukuran dan mengirimkan hasil pengukuran lewat komunikasi *Serial Wireless*. Konfigurasi dengan Arduino NANO adalah (RX=TX, TX=RX). Vcc terhubung dengan 5V Arduino dan GND terhubung dengan GND Arduino.
- b. Sensor Ultrasonik SRF04 sebagai pengukur tinggi lompatan. Konfigurasi dengan Arduino NANO adalah (*trigger*=D7, *echo*=D8), Vcc terhubung dengan 5V Arduino, dan GND terhubung dengan GND Arduino.
- c. Sensor *Proximity* sebagai pendeteksi kondisi pada saat diam dan melompat. Memiliki pin (Data=kuning, GND=hitam, Vcc=merah), konfigurasi dengan Arduino NANO adalah pin Data terhubung dengan A0, Vcc terhubung dengan 5V Arduino, GND terhubung dengan GND Arduino.



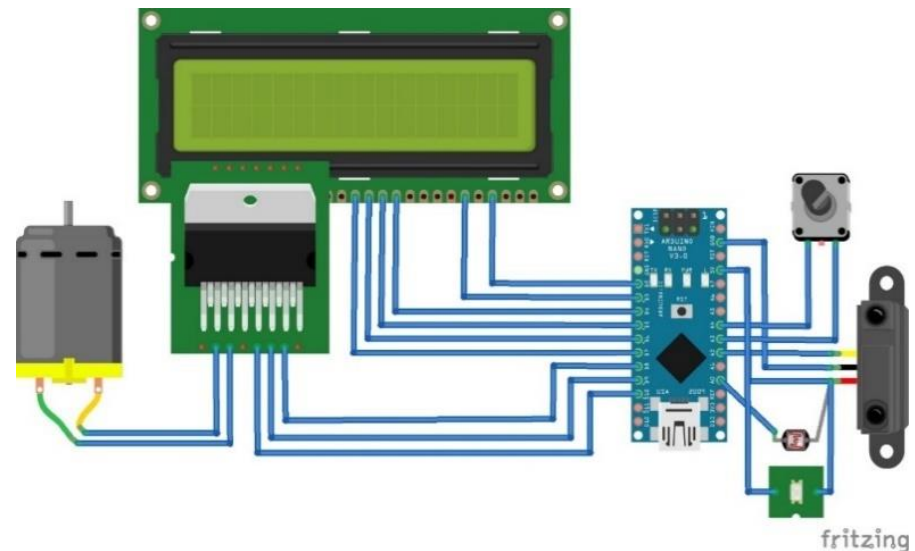
Gambar 65. Blok rangkaian Nano 1 (*Slave 1 Controller*)

3. Blok Rangkaian Arduino Nano 2 (*Slave 2 Controller*)

Rangkaian *Slave 2 Controller* berfungsi sebagai pengatur sistem pengukuran panjang lompatan. Mikrokontroler yang digunakan adalah Arduino NANO V3.0. Untuk gambar Rangkaian *Slave 2 Controller* dapat dilihat pada Gambar 66. Adapun komponen penyusun dan kegunaannya adalah sebagai berikut:

- a. LCD 16x2 dengan konfigurasi pin(RS=D2, E=D3, D4=D4, D5=D5, D6=D6, D7=D7) pin RW terhubung dengan GND Bersama dengan VSS dan Anoda, pin VDD terhubung dengan GND bersama dengan Katoda, dan pin V0 terhubung dengan trimpot 10 k.
- b. *Rotary Encoder* sebagai penghitung jumlah putaran motor saat bergerak pada rel pengukur dan kemudian mengubahnya menjadi satuan jarak (Cm). Sensor ini memiliki pin (VCC, GND, OUT A, OUT B, SWITCH) terhubung dengan Arduino dengan konfigurasi (OUT A = pin Interrupt 0(D2), OUT B = pin Interrupt 1(D3)) pin Vcc terhubung dengan 5V supply, dan pin GND terhubung dengan GND supply.
- c. Sensor *Proximity* sebagai pendeteksi kaki pada saat selesai melompat. Memiliki pin (Data=kuning, GND=hitam, Vcc=merah), konfigurasi dengan Arduino NANO adalah pin Data terhubung dengan A2, Vcc terhubung dengan 5V Arduino, GND terhubung dengan GND Arduino.
- d. Sensor Warna menggunakan LDR dan LED sebagai pendeteksi batas awal dan batas akhir pada ujung rel, ditandai dengan garis berwarna

putih. Memiliki pin (Data, GND, Vcc), konfigurasi dengan Arduino NANO adalah pin Data terhubung dengan A0, Vcc terhubung dengan 5V Arduino, GND terhubung dengan GND Arduino.



Gambar 66. Blok rangkaian Nano 2 (*Slave 2 Controller*)

E. Langkah Pembuatan Alat

Langkah pembuatan alat pada proyek akhir ini terdiri dari pembuatan mekanik, pembuatan elektronik, pemasangan elektronik pada mekanik, pembuatan *software*, dan uji coba rangkaian.

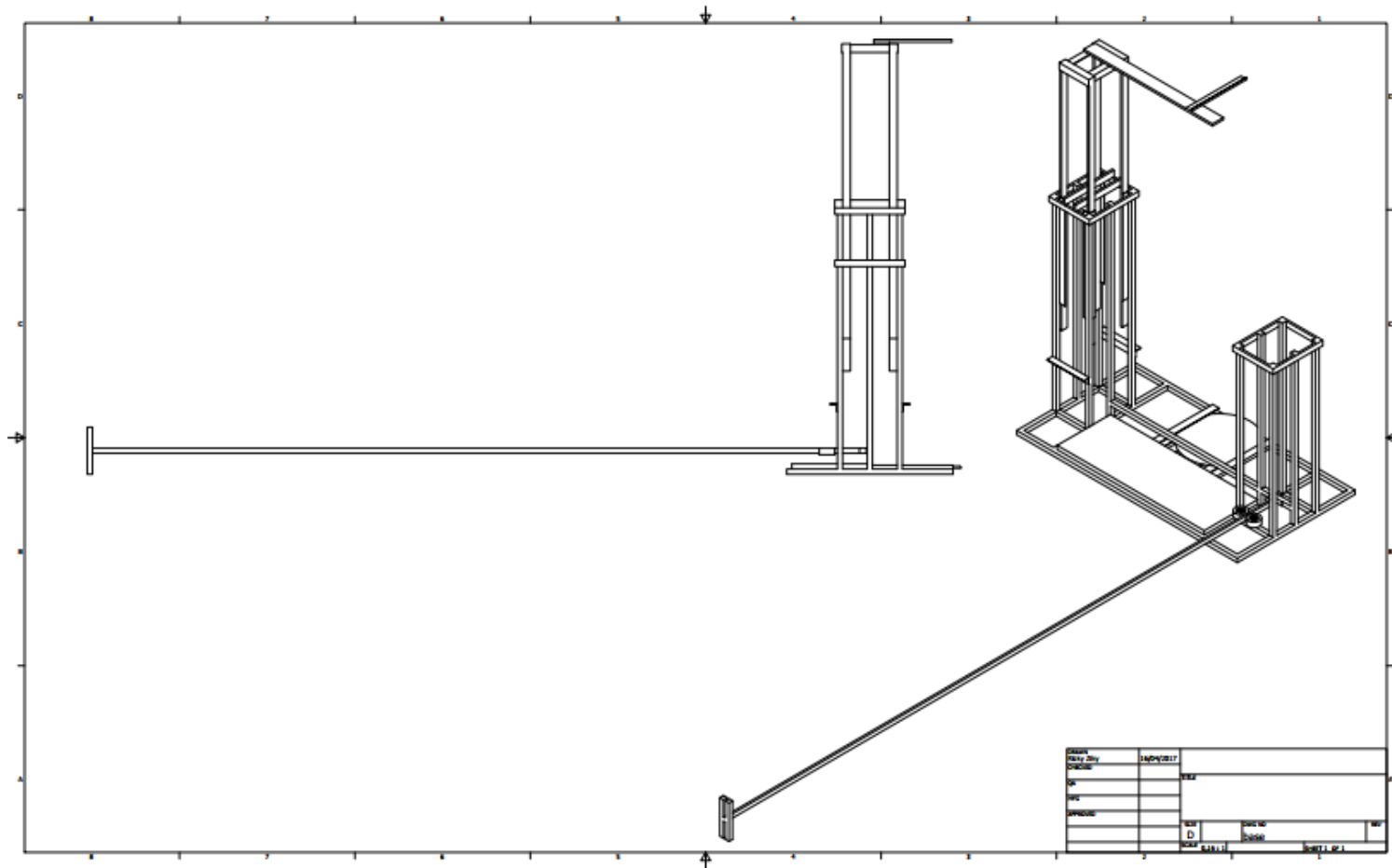
1. Pembuatan Mekanik

a. Perencanaan desain mekanik

Desain mekanik ini dibuat menggunakan *software* Inventor, karena *software* ini sangat cocok untuk membuat skema desain baik 3D maupun 2D secara kompleks dan detail. Perencanaan Ukuran Mekanik

Rangka adalah dengan Panjang 120 Cm, Lebar 64 Cm, Tinggi 225 Cm.

Penampakan desain dapat dilihat pada gambar 67.



Gambar 67. Sketsa gambar mekanik rangka

b. Pengerjaan fabrikasi mekanik

Proses pembuatan mekanik alat dilakukan di kontrakan dengan peralatan seadanya. Bagian mekanik rangka digunakan untuk memasang modul rangkaian. Proses pembuatan mekanik dilakukan dari desain mekanik, pemotongan aluminium sebagai bahan dasar mekanik, plat dan penyesuaian perakitan. Tabel 19 menunjukkan list dari komponen mekanik.

Tabel 19. Daftar komponen mekanik

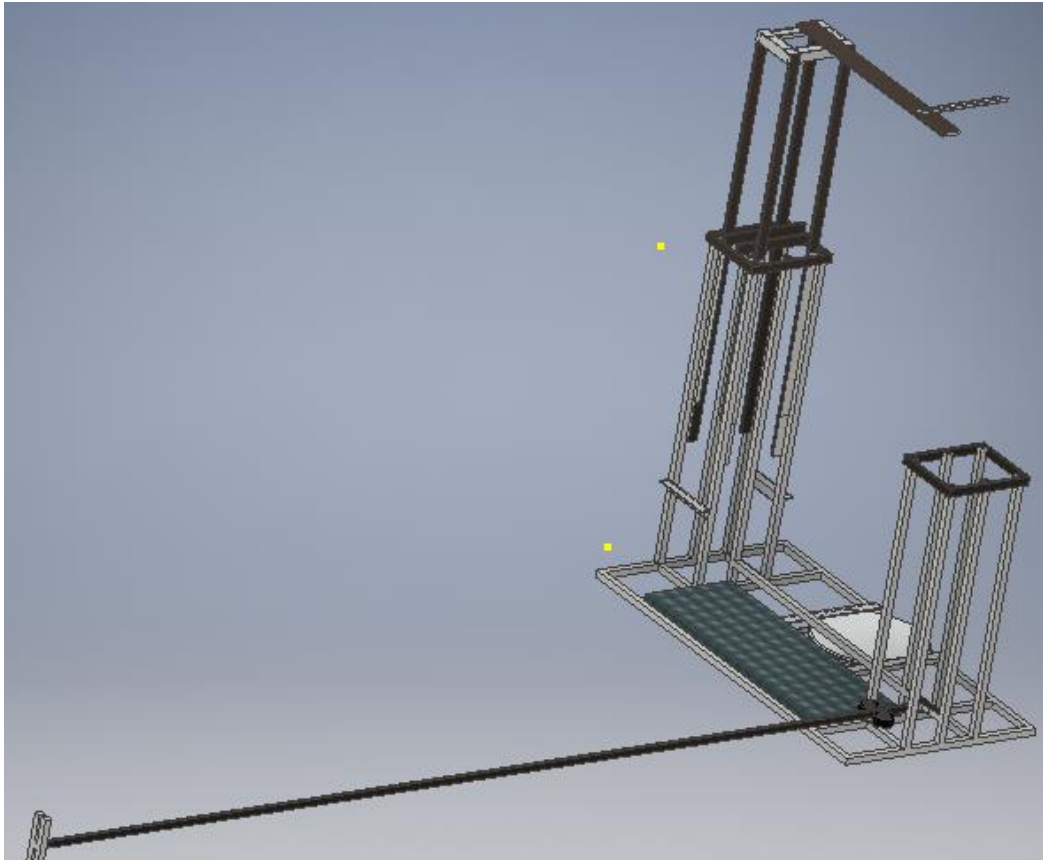
No.	Nama	Jumlah
1.	Alumunium Kotak $\frac{3}{4}$ inch	6 batang (6 meter)
2.	Alumunium Persegi $\frac{3}{4}$ inch	4 batang (6 meter)
3.	Alumunium Siku (<i>Spligot</i> 2 mm)	6 meter
4.	Alumunium Siku (<i>Spligot</i> 5 mm)	4 meter
5.	Paku rivet	1 pack
6.	Mur, Baut, Ring	Secukupnya
7.	Rantai Sepeda mini	2 buah
8.	Gear sepeda mini (6 cm)	1 buah
9.	Gear sepeda mini (8 cm)	1 buah
10.	Akrilik 3 mm	20 cm x 15 cm
11.	Roda mobil RC	2 buah
12.	Pir (Pegas) kampas rem motor	2 buah
13.	Box kotak 15 cm x 10 cm	2 buah
14.	Box kotak 20 cm x 15 cm	2 buah

Langkah-langkah pengerjaan mekanik meliputi:

- 1) Memotong Alumunium kotak sebagai rangka dan base sesuai ukuran yang sudah tertera pada desain yang telah dibuat sebelumnya.
- 2) Memotong Alumunium persegi sebagai tiang pengukur tinggi badan, sesuai dengan ukuran yang sudah tertera pada desain.

- 3) Memotong Alumunium siku (*spligot*) dengan lebar 2 cm secukupnya untuk difungsikan sebagai sambungan antar aluminium.
- 4) Merakit Alumunium kotak berbentuk persegi untuk dijadikan base rangka dengan dimensi ukuran 120 cm x 64 cm.
- 5) Merakit tiang penyangga box kontroller dengan dimensi ukuran 25 cm x 20 cm x 75 cm.
- 6) Merakit tiang penyangga mekanik *slider* dengan dimensi ukuran 25 cm x 20 cm x 100 cm
- 7) Merakit tiang *slider* sebagai pengukur tinggi badan dengan dimensi 21 cm x 16 cm x 125 cm.
- 8) Menyatukan semua bagian bagian sebelumnya sesuai dengan petunjuk desain.
- 9) Merakit mekanik *slider* horizontal pada rel sepanjang 3 meter sebagai pengukur panjang lompatan.

Dari prsoes pengerjaan mekanik yang telah dilakukan didapatkan hasil bentuk base dan rangka seperti gambar 68 berikut.

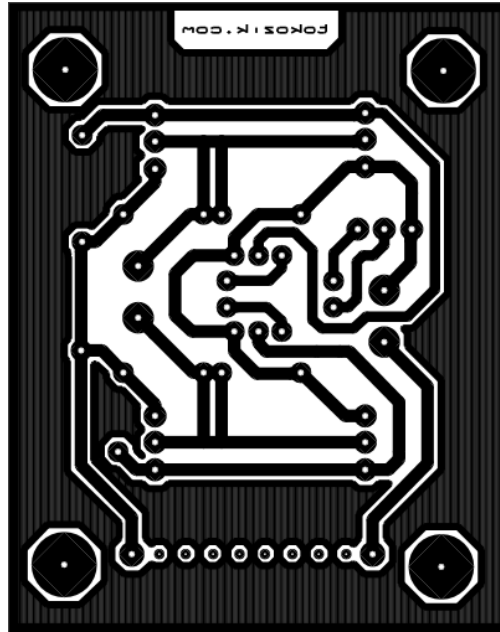


Gambar 68. Gambar 3D mekanik rangka

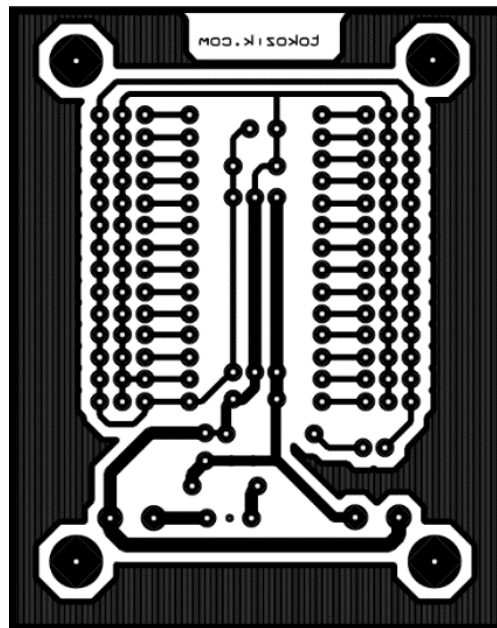
2. Pembuatan Elektronik

a. Pembuatan Layout PCB

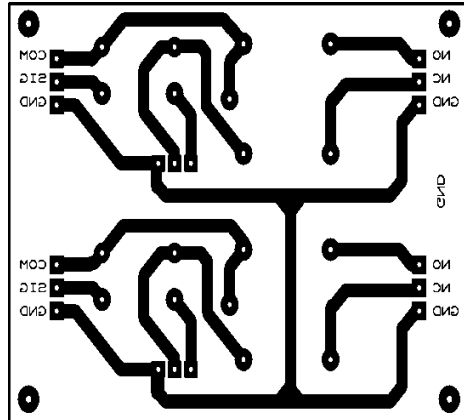
Langkah awal pembuatan Desain Elektronik adalah menggambar *layout* rangkaian PCB dengan perangkat lunak. Perangkat lunak yang digunakan yaitu Proteus. Hasil penggambaran *layout* PCB dapat dilihat pada Gambar 69, 70, 71, 72 berikut.



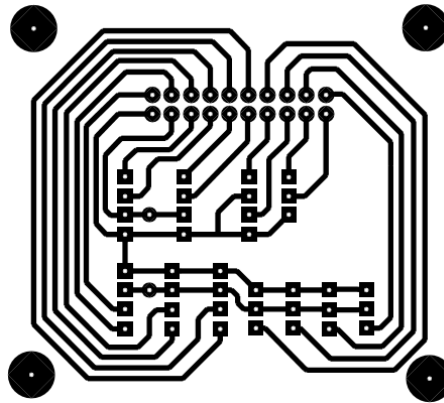
Gambar 69. *Layout Driver Motor Mosfet H-Bridge single channel*



Gambar 70. *Layout Shield Arduino NANO*



Gambar 71. Layout *Driver Motor Relay H-Bridge single channel*



Gambar 72. Layout Hub sensor

b. Penyablonan PCB

Setelah *layout* selesai dibuat maka langkah selanjutnya yaitu penyablonan, atau sering disebut pembuatan PCB. Proses penyablonan dilakukan dengan cara sebagai berikut:

- 1) Mencetak *layout* PCB yang telah dibuat pada kertas glossi.
- 2) Desain *layout* yang sudah dicetak pada kertas glossi dipindahkan atau disablonkan ke PCB dengan cara disetrika dengan panas sedang sampai dirasa gambar sudah terpindah semua.

- 3) Setelah gambar *layout* menempel semua pada PCB maka tunggu sebentar sampai dingin, kemudian bersihkan kertas yang menempel pada PCB dengan menggunakan air bersih

c. Pelarutan dan Pengeboran PCB

Proses melarutkan PCB atau sering disebut juga pembuatan jalur pada PCB. Langkah ini merupakan lanjutan dari proses sebelumnya, pada proses ini PCB yang sudah di sablon kemudian dilarutkan menggunakan cairan kimia HCl (*Asam Chloride*) sampai jalur rangkaian terbentuk semua. Kemudian setelah jalur PCB terbuat, mengambil PCB dari cairan HCl dan membersihkan dengan air sampai sisa tinta sablon bersih. Langkah selanjutnya yaitu pengeboran PCB, PCB di bor sesuai dengan titik titik yang telah ditentukan.

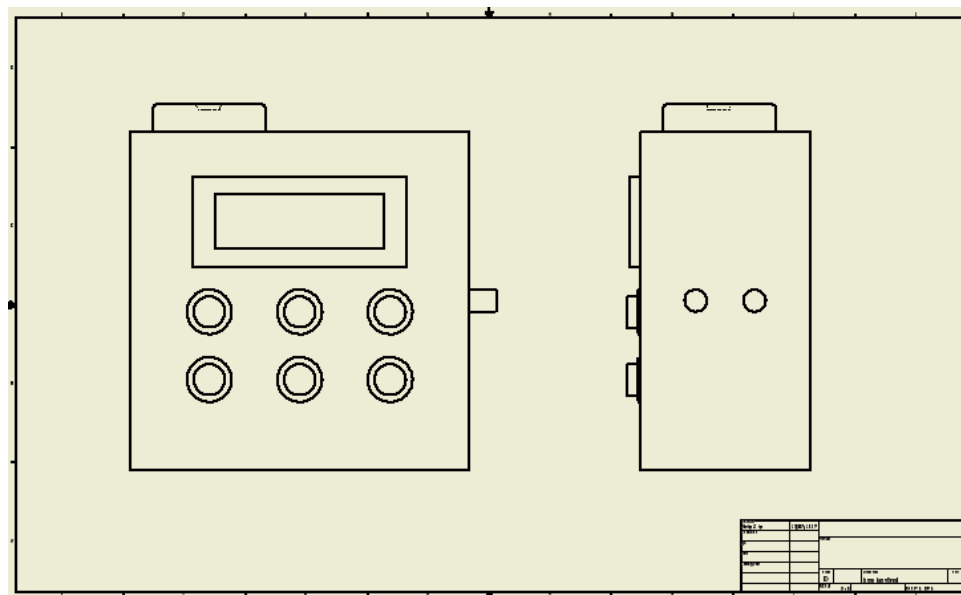
d. Pemasangan Komponen

Langkah selanjutnya yaitu pemasangan seluruh komponen pada PCB, dengan urutan:

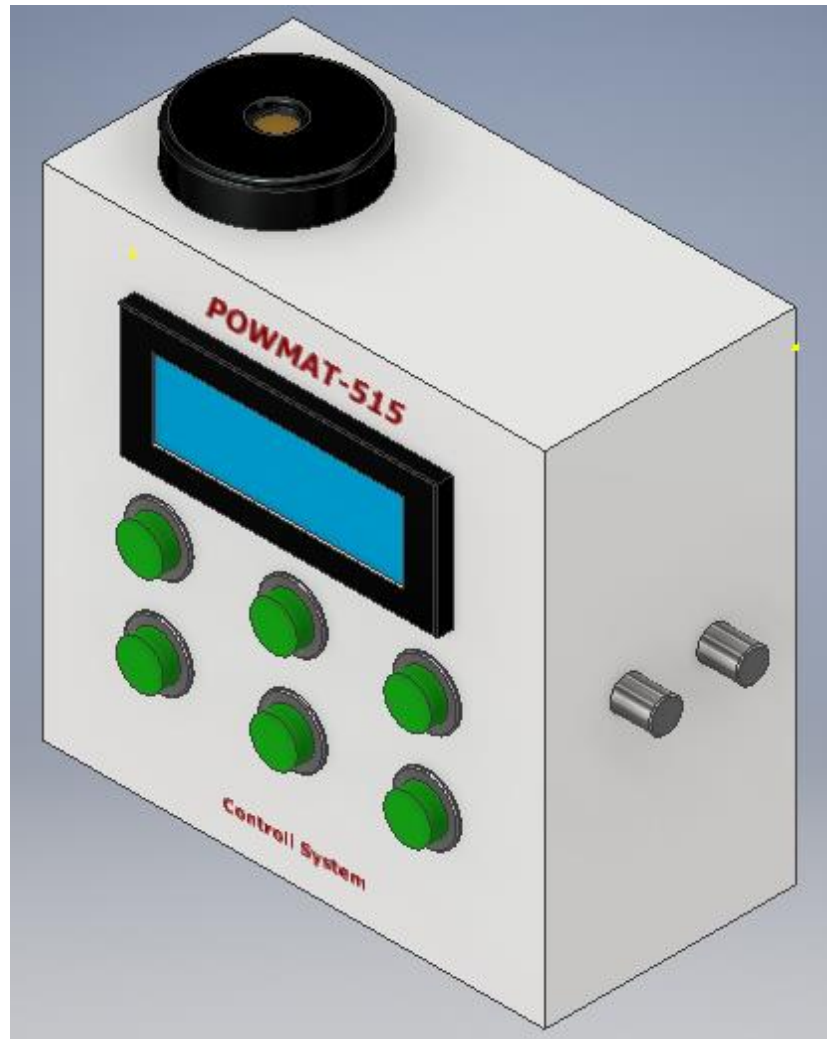
- 1) Menyiapkan seluruh komponen yang dibutuhkan.
- 2) Memasang komponen dari komponen yang memiliki ukuran yang paling kecil. Terlebih pemasangan jumper didahulukan.
- 3) Menyolder kaki komponen sampai semua komponen terpasang, dan kematangan timah selalu dijaga.
- 4) Menguji rangkaian apakah sudah dapat bekerja dengan baik atau belum.

3. Pembuatan *Box Control Sistem*

Pembuatan *box control sistem* dilakukak dengan tahapan yang pertama yaitu membuat desain box menggunakan aplikasi Inventor 2017. Dimensi ukuran box adalah 15 cm x 15 cm x 7,5 cm. hasil desain dapat dilihat pada gambar 74. Setelah desain box sudah jadi tinggal di *convert* ke dalam bentuk format .cdr, kemudian di cetak dengan menggunakan bahan akrilik dengan tebal 3 mm berwarna putih susu. Desain 2D alat dapaat dilihat pada gambar 73, sedangkan desain 3D alat dapat dilihat pada gambar 74.



Gambar 73. Gambar 2D Desain Box Kontrol Sistem

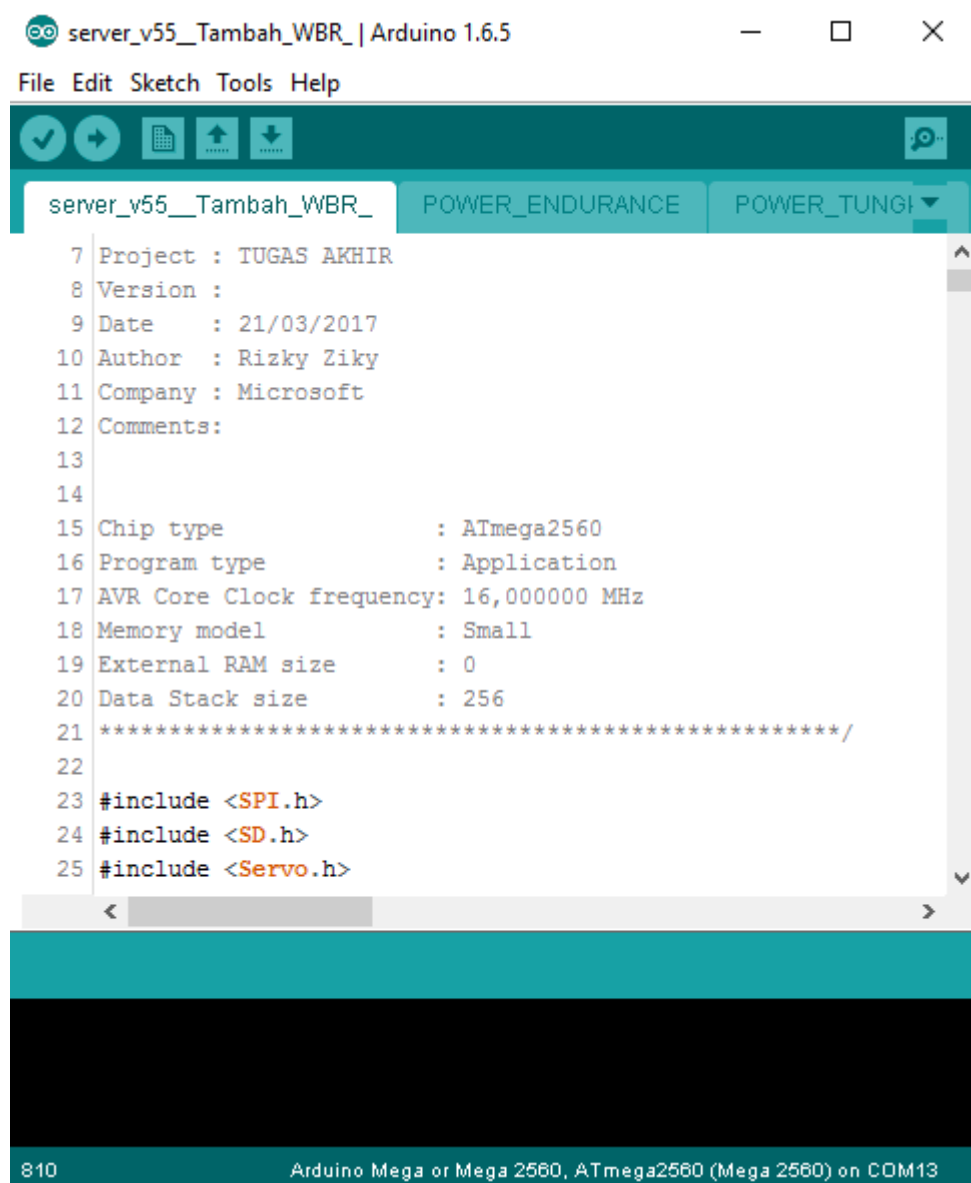


Gambar 74. Gambar 3D Desain Box Kontrol Sistem

F. Perangkat Lunak

1. Arduino IDE

Arduino merupakan *board open source* yang mempunyai aplikasi *software* sendiri bernama Arduino IDE yang digunakan untuk meng-*compile* kode program. Perkembangan aplikasi selalu mengalami *update*. Arduino IDE juga dilengkapi dengan *library* yang memudahkan kita dalam proses pembuatan sketsa program. Gambar 75 merupakan tampilan dari Arduino IDE.



Gambar 75. Tampilan Perangkat Lunak dari Arduino IDE

Untuk melakukan upload program pada *hardware* Arduino, maka diperlukan sedikit *setting* yang meliputi pemilihan jenis *chip* mikrokontroler Arduino yang akan digunakan (dipastikan sesuai dengan *hardware*), *setting port* (COM), dan setelah semua di *setting* proses selanjutnya adalah meng-*compile* kode program untuk memastikan masih ada program yang *error* atau tidak. Jika proses *compile* sudah selesai, proses selanjutnya adalah

meng-*upload* program ke *hardware* dengan cara Ctrl+U atau meng klik tombol berbentuk anak panah ke kanan di bar atas. Setelah proses *upload* selesai maka akan ada tulisan *done uploading* di bar status.

2. Text Editor (Notepad++)

Notepad++ adalah program aplikasi pengembang yang berguna untuk mengedit teks dan skrip kode pemrograman. Versi terbaru program ini adalah *Notepad++* v5.9, yang dirilis pada tanggal 06 April 2012. *Software Notepad++* dibuat dan dikembangkan oleh Tim *Notepad++*.

Software ini memiliki kelebihan pada peningkatan kemampuan sebuah program *text editor*, lebih dari sekedar program *Notepad* bawaan Windows. *Notepad++* bisa mengenal *tag* dan kode dalam berbagai bahasa pemrograman. Fitur pencarian tingkat lanjut dan pengeditan teks yang tersedia juga cukup ampuh, sangat membantu tugas seorang programmer atau *developer* dalam menyelesaikan skrip kode programnya. Program *Notepad++* banyak di aplikasikan dan digunakan oleh kalangan pengguna komputer di bidang pemrograman aplikasi *desktop* dan *web*. Gambar 76 merupakan tampilan dari *Notepad++*.

```

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <title>SMARTTECH 515 Home</title>
5 <meta charset="utf-8" />
6 <meta name="viewport" content="width=device-width, initial-sc
7 <!--[if lte IE 8]><script src="assets/js/ie/html5shiv.js"></s
8 <link rel="stylesheet" href="assets/css/main.css" />
9 <!--[if lte IE 8]><link rel="stylesheet" href="assets/css/ie8
10 </head>
11 <body class="homepage">
12 <div id="page-wrapper">
13
14 <!-- Header -->
15 <div id="header-wrapper" class="wrapper">
16 <div id="header">
17
18 <!-- Logo -->
19 <div id="logo">
20 <h1><a href="index.html">SMARTTECH 515
21 <p>Sport Technology By ToKoZik</p>
22 </div>
23
24 <!-- Nav -->
25 <nav id="nav">
26 <ul>
27 <li class="current"><a href="inde
28 <li>
29 <a href="#">Profil</a>
30 <ul>
31 <li><a href="#">Staff Kam

```

Gambar 76. Tampilan dari Perangkat Lunak Notepad++

3. Algoritma Sistem

Algoritma sistem di dalam pembuatan alat ini menjelaskan tentang alur semua dari proses sistem yang berjalan secara umum. Adapun algoritmanya adalah sebagai berikut:

- a. *Start*
- b. *Booting*
- c. Tampilan Stanby
- d. Tekan tombol *OK*

- e. Masuk menu
- f. Pilih menu pengukuran
- g. Menu pengukuran Tinggi & Berat
 - 1) Tekan tombol *OK*
 - 2) Mulai ukur Tinggi & Berat
 - 3) Tampil hasil pengukuran di LCD
 - 4) Simpan data
 - 5) Tekan tombol *BACK*
 - 6) Kirim data ke Internet
- h. Menu pengukuran *Power*
 - 1) Tekan tombol *BACK* untuk memilih mode [MANU] atau [AUTO]
 - 2) Tekan tombol *OK*
 - 3) Pilih pengukuran *power Vertical Jump* atau *Horizontal Jump*
 - 4) Tekan tombol *OK*
 - 5) Pilih Jenis Kelamin
 - 6) Tekan tombol *OK*
 - 7) Pilih Kategori umur
 - 8) Tekan tombol *OK*
 - 9) Mulai pengukuran *Power*
 - 10) Tampil hasil pengukuran di LCD
 - 11) Simpan data
 - 12) Tekan tombol *BACK*
 - 13) Kirim data ke Internet

- i. Menu pengukuran *Whole Body Reaction*
 - 1) Tekan tombol *OK*
 - 2) Pilih Jenis kelamin
 - 3) Tekan tombol *OK*
 - 4) Pilih metode pengukuran [*AUDIO*] atau [*VISUAL*] dengan cara tekan tombol *UP* atau *DOWN*
 - 5) Tekan tombol *OK*
 - 6) Mulai pengukuran *Whole Body Reaction*
 - 7) Tampil hasil pengukuran di LCD
 - 8) Simpan data
 - 9) Tekan tombol *BACK*
 - 10) Kirim data ke Internet
- j. Menu pengukuran *Power Endurance*
 - 1) Tekan tombol *OK*
 - 2) Mulai pengukuran *Power Endurance*
 - 3) Tampil hasil pengukuran di LCD
 - 4) Simpan data
 - 5) Tekan tombol *BACK*
 - 6) Kirim data ke Internet
- k. Menu Turunkan Tiang
 - 1) Tekan tombol *OK*
 - 2) Tiang Turun ke posisi bawah
- l. Menu Lihat Data Kelompok
 - 1) Tekan tombol *OK*

- 2) Pilih data kelompok
 - 3) Tekan tombol *OK*
 - 4) Kirim data ke Serial Monitor
 - 5) Tampil data di Serial Monitor
- m. Menu Informasi *Micro SD*
- 1) Tekan tombol *OK*
 - 2) Tampil informasi kapasitas penyimpanan *micro SD* di LCD
- n. Menu Pengaturan Micro SD
- 1) Tekan tombol *OK*
 - 2) Pilih data kelompok
 - 3) Tekan tombol *OK*
 - 4) Hapus data kelompok
 - 5) Jika tombol *UP & OK & SAVE* ditekan
 - 6) Format *micro SD*
- o. Selesai

4. Algoritma Program

Algoritma program di dalam pembuatan alat ini terbagi menjadi 5 bagian yaitu algoritma *master controller*, algoritma *slave 1 controller*, algoritma *slave 2 controller*, algoritma *slave 3 controller*, dan yang terakhir adalah algoritma *database website*. Semua algoritma ini sangat penting guna menunjang keberhasilan pembuatan alat dengan sistem yang benar-benar terintegrasi.

- a. Algoritma Arduino Mega (*Master Controller*)

- 1) Start.
- 2) Baca *micro SD* card.
- 3) Apakah ada *micro SD* card.
- 4) Jika tidak ‘Masukkan *micro SD* card’.
- 5) Jika Ada lanjut step berikutnya.
- 6) Tulis data awal pada *micro SD* card.
- 7) Baca EEPROM.
- 8) Inisialisasi Port I/O, Inisialisasi Timer.
- 9) Masuk mode *booting*.
- 10) Masuk Menu StanBy.
- 11) Inisialisasi *Button UP, DOWN, OK, SAVE, BACK*.
- 12) Jika *button OK* ditekan masuk langkah 14. Jika tidak kembali ke langkah 10 “Menu StanBy”.
- 13) Deklarasi variable `menu=1, kelompok=1, user=1, PT=1, menu_PT=1, pilih_kelamin=1, data_TB=0, data_BB=0`.
- 14) Tampil Menu Utama “ Scan Berat & Tinggi, P. Tungkai [MANU], W Body Reaction, P. *Endurance*, Turunkan Tiang, Lihat Data Kelompok, Informasi *micro SD*, Pengaturan Penyimpanan ”. lanjut langkah berikutnya.
- 15) Jika *button DOWN* ditekan `menu++`. Jika tidak lanjut langkah berikutnya.
- 16) Jika *button UP* ditekan `menu--`. Jika tidak lanjut langkah berikutnya.
- 17) Jika `menu==1`. Lanjut langkah berikutnya.

- 18) Jika *button OK* ditekan, lanjut langkah 46. Jika tidak lanjut langkah berikutnya.
- 19) Jika menu==2.
- 20) Jika *button BACK* ditekan PT++.
- 21) Jika PT==1 P.Tungkai [MANU].
- 22) Jika PT==2 P.Tungkai [AUTO].
- 23) Jika *button OK* ditekan dan PT==1, lanjut step 46. Jika tidak lanjut step berikutnya.
- 24) Jika *button OK* ditekan dan PT==2, lanjut step 36. Jika tidak lanjut step berikutnya.
- 25) Jika menu==3.
- 26) Jika *button OK* ditekan, lanjut step 65. Jika tidak lanjut step berikutnya.
- 27) Jika menu==4.
- 28) Jika *button OK* ditekan, lanjut step 53. Jika tidak lanjut step berikutnya.
- 29) Jika menu==5.
- 30) Jika *button OK* ditekan, tampil “ B.B : data_BB T.B1 :data_TB
T.B2 : data_TB Prox: data_prox WBR”. Jika tidak lanjut step berikutnya.
- 31) Jika menu==6.
- 32) Jika *button OK* ditekan, lanjut step 90. Jika tidak lanjut step berikutnya.
- 33) Jika menu==7.

- 34) Jika *button OK* ditekan, tampil “ Sisa ruang:.....Mb Sisa ruang:.....Kb ”. Jika tidak lanjut step berikutnya.
- 35) Jika menu==8.
- 36) Jika *button OK* ditekan, lanjut step 42. Jika tidak lanjut step berikutnya.
- 37) Jika *button SAVE* ditekan, masuk mode simpan data.
- 38) “Kelompok= kelompok tekan <OK>”.
- 39) Jika *button UP* ditekan, kelompok++ lanjut step berikutnya. Jika tidak lanjut step berikutnya.
- 40) Jika *button DOWN* ditekan, kelompok— lanjut step berikutnya. Jika tidak lanjut step berikutnya.
- 41) Jika *button OK* ditekan “User: user tekan <SAVE>”. Jika tidak kembali ke step 38 (“Kelompok= kelompok tekan <OK>”)
- 42) Jika *button UP* ditekan, user++ lanjut step berikutnya. Jika tidak lanjut step berikutnya.
- 43) Jika *button DOWN* ditekan, user— lanjut step berikutnya. Jika tidak lanjut step berikutnya.
- 44) Jika *button BACK* ditekan, kembali ke step J (Menu StanBy). Jika tidak lanjut step berikutnya.
- 45) Jika *button SAVE* ditekan lanjut step 46. Jika tidak kembali ke step 38 (“Kelompok= kelompok tekan <OK>”).
- 46) Tampil “ SCANNING TINGGI & BERAT Tekan <OK> utk mulai TB: [data_TB]Cm, BB:[data_BB]Kg ”.

- 47) Jika *button OK* ditekan lanjut step 50. Jika tidak kembali ke step 46.
- 48) Jika *button BACK* ditekan kembali ke step 13. Jika tidak kembali ke step 46.
- 49) Pengukuran Tinggi Badan dan Berat Badan.
- 50) Simpan data_TB.
- 51) Simpan data_BB.
- 52) Kembali ke step 46.
- 53) Tampil “ *POWER ENDURANCE* Tekan <OK> utk mulai Tekan <UP> utk kalbi Tekan <BACK> utk bck ”.
- 54) Jika *button OK* ditekan lanjut step 57. Jika tidak kembali ke step 53.
- 55) Jika *button OK* ditekan lanjut step 63. Jika tidak kembali ke step 53.
- 56) Jika *button BACK* ditekan kembali ke step 13. Jika tidak kembali ke step 53.
- 57) Kirim Perintah Pengukuran *Power Endurance* pada *Slave 1 Controller*.
- 58) Terima data *Power Endurance* dari *Slave 1*.
- 59) Parsing data *Power Endurance*.
- 60) Simpan data Parsing *Power Endurance* ke variable PE1, PE2, PE3, PE4, PE5, PE6, PE7, PE8, PE9, PE10, RATA2PE, MAXPE, dan MINPE.

- 61) Tampilkan data “ PE1, PE2, PE3, PE4, PE5, PE6, PE7, PE8, PE9, PE10, RATA2PE, MAXPE, dan MINPE ”.
- 62) Kembali ke step 53.
- 63) Kirim Perintah Kalibrasi pada *Slave 1 Controller*.
- 64) Kembali ke step 53.
- 65) Tampil “JENIS KELAMIN Laki-laki Perempuan”.
- 66) Jika menu Laki-laki ditekan, lanjut ke step 68. Jika tidak kembali ke step 65.
- 67) Jika menu Perempuan ditekan, lanjut ke step 79. Jika tidak kembali ke step 65.
- 68) Jika Ada Orang lanjut ke step 71. Jika tidak lanjut ke step 69.
- 69) Tampil “ W BODY REACTION [L] BELUM SIAP Ambil Ancang Ancang Tekan <BACK> utk bck ”.
- 70) Kembali ke step 68.
- 71) Tampil “ W BODY REACTION [L] SUDAH SIAP Tekan <OK> utk mulai Tekan <BACK> utk bck ”.
- 72) Jika Ada Orang dan *button* <OK> ditekan, lanjut ke step 73. Jika tidak kembali ke step 71.
- 73) Tampil “ *WHOLE BODY REACTION* Reaction: [Timer] detik”.
- 74) Timer *ON* (berjalan).
- 75) Jika tidak Ada Orang, lanjut ke step 76. Jika tidak kembali ke step 74.
- 76) Timer *OFF* (berhenti).
- 77) Tampil “W. BODY REACTION Reaction [Timer]”. Lanjut step 78.

- 78) Jika *button SAVE* ditekan, kembali ke step 68. Jika tidak kembali ke step 77.
- 79) Jika Ada Orang lanjut ke step 82. Jika tidak lanjut ke step 80.
- 80) Tampil “ *W BODY REACTION [P] BELUM SIAP* Ambil Ancang Ancang Tekan *<BACK>* utk bck ”.
- 81) Kembali ke step 79.
- 82) Tampil “ *W BODY REACTION [P] SUDAH SIAP* Tekan *<OK>* utk mulai Tekan *<BACK>* utk bck ”.
- 83) Jika Ada Orang dan *button <OK>* ditekan, lanjut ke step 84. Jika tidak kembali ke step 82.
- 84) Tampil “ *WHOLE BODY REACTION* Reaction: [Timer] detik”.
- 85) Timer *ON* (berjalan).
- 86) Jika tidak Ada Orang, lanjut ke step 87. Jika tidak kembali ke step 84.
- 87) Timer *OFF* (berhenti).
- 88) Tampil “*W. BODY REACTION Reaction [Timer]*”. Lanjut step 89.
- 89) Jika *button SAVE* ditekan, kembali ke step 82. Jika tidak kembali ke step 88.
- 90) Tampil “ data: kelompok[kelompok] Tekan *<OK>* utk lihat”
- 91) Jika *button UP* ditekan kelompok=kelompok+1 lanjut step berikutnya. Jika tidak lanjut step berikutnya.
- 92) Jika *button DOWN* ditekan kelompok=kelompok-1 lanjut step berikutnya. Jika tidak lanjut step berikutnya.

- 93) Jika *button OK* ditekan Kirim data kelompok[kelompok] ke Serial Monitor lanjut step berikutnya. Jika tidak kembali ke step 90.
- 94) Jika *button BACK* ditekan kembali ke step 13. jika tidak kembali ke step 90.
- 95) Tampil “ Hapus :kelompok[kelompok] <OK> format<SAVE><UP><OK> ”.
- 96) Jika *button UP* ditekan kelompok=kelompok+1 lanjut step berikutnya. Jika tidak lanjut step berikutnya.
- 97) Jika *button DOWN* ditekan kelompok=kelompok-1 lanjut step berikutnya. Jika tidak lanjut step berikutnya.
- 98) Jika *button OK* ditekan, Hapus data kelompok[kelompok] ke Serial Monitor lanjut step berikutnya. Jika tidak kembali ke step 95.
- 99) Jika *button SAVE* dan *UP* dan *OK* ditekan, Format *micro SD* lanjut step berikutnya. Jika tidak kembali ke step 95.
- 100) Jika *button BACK* ditekan kembali ke step 13. jika tidak kembali ke step 95.
- 101) Simpan data kelompok.
- 102) Simpan data user.
- 103) Stop.
- b. Algoritma Arduino NANO 1 (*Slave 1 Controller*)
- 1) Start.
 - 2) Baca EEPROM.
 - 3) Inisialisasi Port I/O, Inisialisasi Timer.
 - 4) Kalibrasi Sensor Ultrasonik.

- 5) Simpan Data Kalibrasi di EEPROM.
 - 6) Jika Serial Available lanjut step 7. Jika tidak kembali ke step 6.
 - 7) Data Serial Masuk.
 - 8) Jika Data Serial == 1 lanjut ke step 11. Jika tidak lanjut ke step 9.
 - 9) Jika Data Serial == 5 lanjut ke step 12. Jika tidak lanjut ke step 10.
 - 10) Jika Data Serial == 7.lanjut ke step 13. Jika tidak lanjut ke step 23.
 - 11) Pengukuran *Power Endurance*.
 - 12) Parsing Data *Power Endurance*.
 - 13) Simpan Data *Power Endurance* di EEPROM.
 - 14) Kirim Data *Power Endurance* ke *Master Controller*.
 - 15) Lanjut step 23.
 - 16) Pengukuran *Vertikal jump*.
 - 17) Parsing Data *Vertikal jump*.
 - 18) Simpan Data Jarak dan Waktu *Vertikal jump* di EEPROM.
 - 19) Kirim Data Jarak dan Waktu *Vertikal jump* ke *Master Controller*.
 - 20) Lanjut step 23.
 - 21) Kalibrasi Sensor.
 - 22) Simpan Data Kalibrasi di EEPROM.
 - 23) Stop.
- c. Algoritma Arduino NANO 2 (*Slave 2 Controller*)
- 1) Start.
 - 2) Baca EEPROM.
 - 3) Inisialisasi Port I/O, Timer, Interrupt 0 dan 1.

- 4) Tampil “*STANDING BOARD JUMP* SIAP menunggu perintah dari Server ZIOS”.
 - 5) Jika Serial Available lanjut step 6. Jika tidak kembali ke step 4.
 - 6) Data Serial masuk.
 - 7) Jika Data Serial == 1 lanjut ke step 8. Jika tidak kembali ke step 4.
 - 8) Tampil “ZIOS H. *JUMP=Ready J.Lomp*”.
 - 9) Jika Serial Available lanjut step 10. Jika tidak kembali ke step 8.
 - 10) Data Serial masuk.
 - 11) Jika Data Serial==7 lanjut ke step 12. Jika tidak kembali ke step 8.
 - 12) Pengukuran *Standing board jump*.
 - 13) Simpan Data Jarak dan Waktu *Standing board jump* di EEPROM.
 - 14) Kirim Data Jarak dan Waktu *Standing board jump* ke *Master Controller*.
 - 15) Stop.
- d. Algoritma Wemos D1 (*Slave 3 Controller*)
- 1) Start.
 - 2) Inisialisasi SSID.
 - 3) Inisialisasi Password.
 - 4) Inisialisasi Host.
 - 5) Inisialisasi *Database Name*.
 - 6) Jika Serial Available lanjut step 7. Jika tidak kembali ke step 6.
 - 7) Data Serial masuk.
 - 8) Parsing Data.

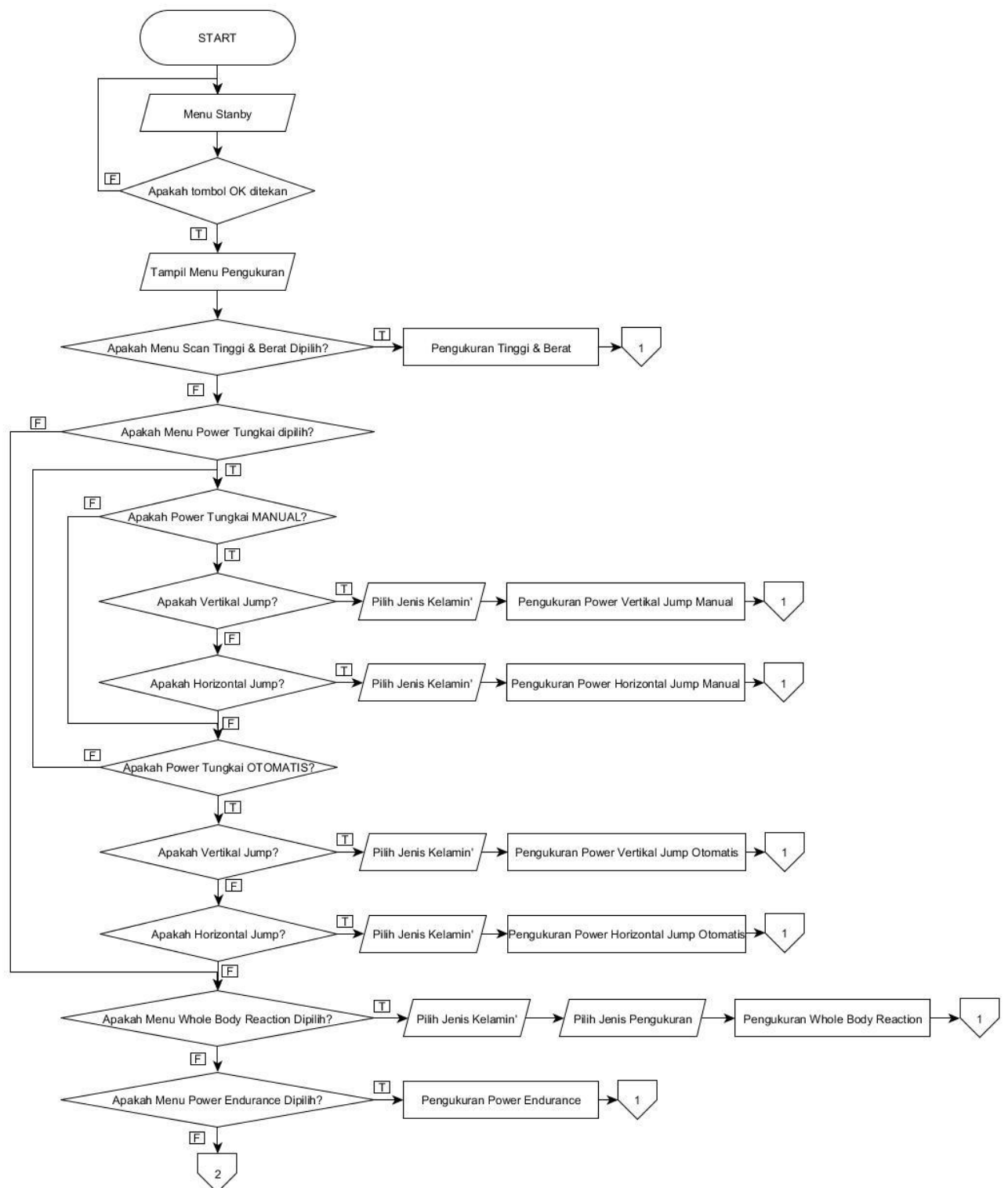
- 9) Simpan Data Tinggi Badan, Berat Badan, *Vertikal jump*, *Standing board jump*, *Whole Body Reaction*, *Power Endurance* di EEPROM.
- 10) Kirim Data Tinggi Badan, Berat Badan, *Vertikal jump*, *Standing board jump*, *Whole Body Reaction*, *Power Endurance* ke Server *Database Online*.
- 11) Stop.

e. Algoritma *Database Website*

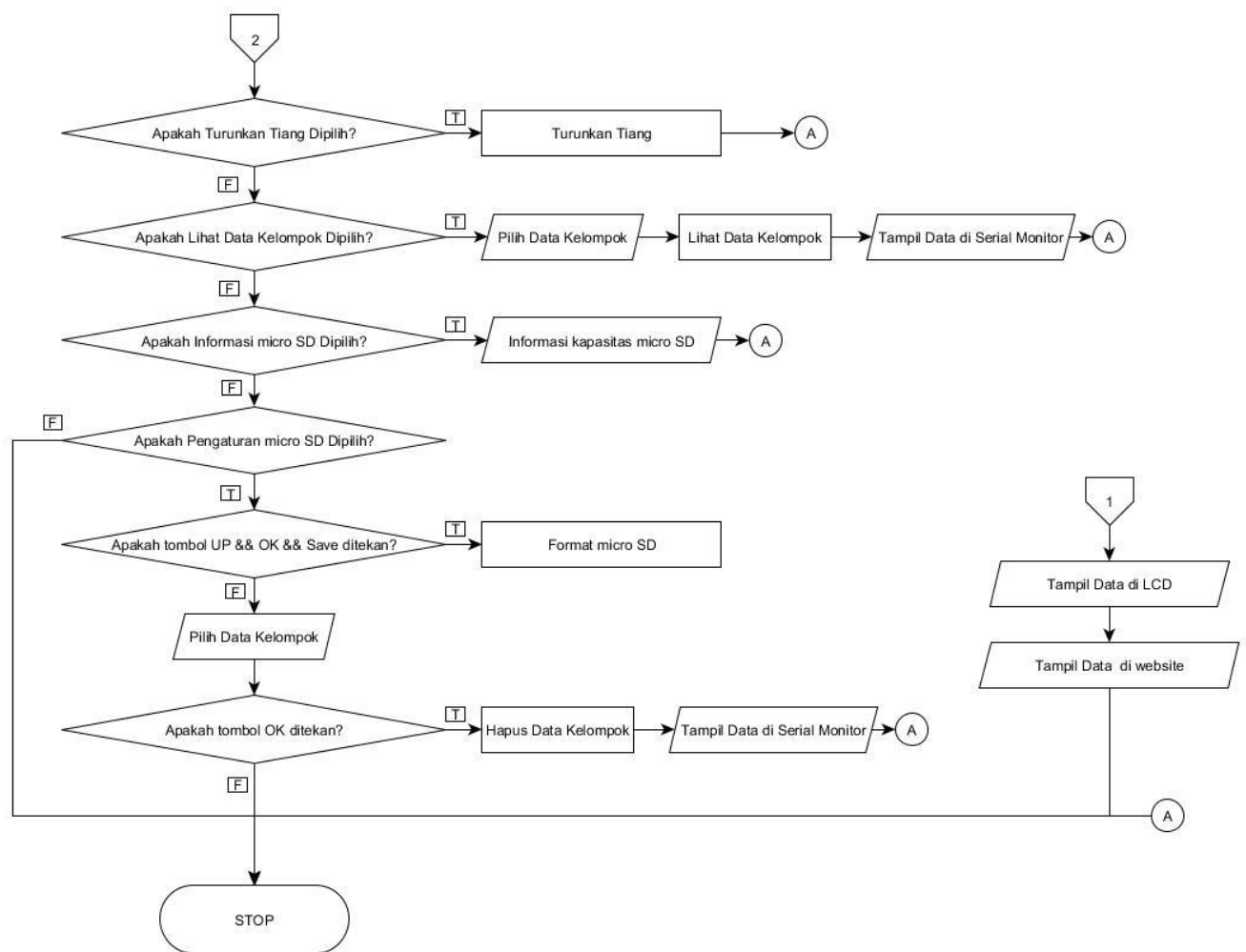
- 1) Start.
- 2) Inisialisasi *Database Host*.
- 3) Inisialisasi *Database User*.
- 4) Inisialisasi *Database Password*.
- 5) Inisialisasi *Database Name*.
- 6) Inisialisasi *Database Tabel*.
- 7) Inisialisasi *Database Id*.
- 8) Jika Ada data masuk '*Update database tabel*' lanjut langkah 9. Jika tidak kembali ke langkah 6.
- 9) Tampilkan pada *website*.
- 10) Stop.

5. *Flowchart Sistem*

Berikut ini adalah diagram alir keseluruhan program dari alat pengukur *power tungkai*, *Power Endurance*, dan *Whole Body Reaction* ditampilkan pada gambar dibawah ini. Pada gambar tersebut, setiap bagian dari menu pengukuran diwakilkan dengan proses.



Gambar 77. Flowchart Sistem 1

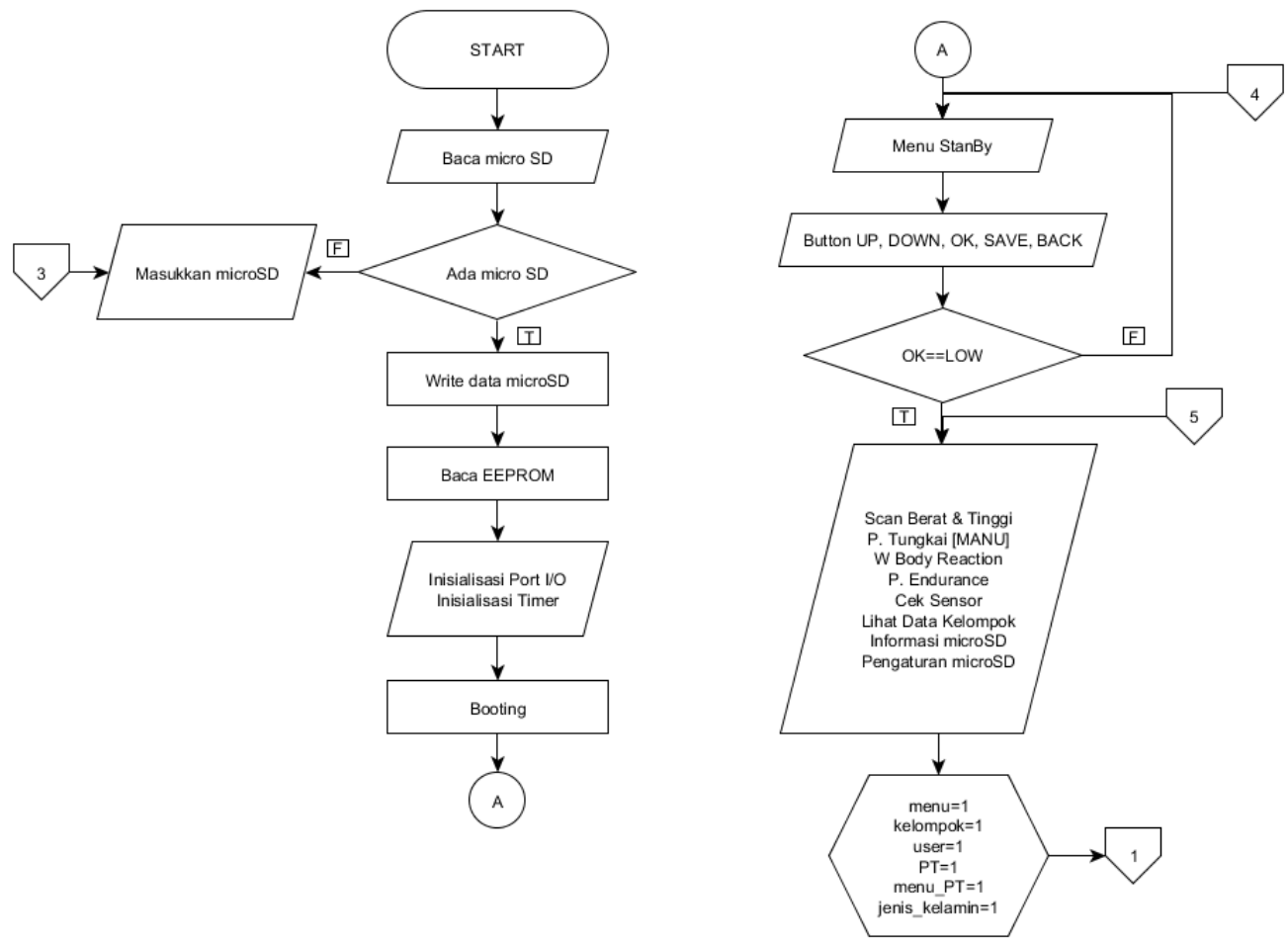


Gambar 78. Flowchart Sistem 2

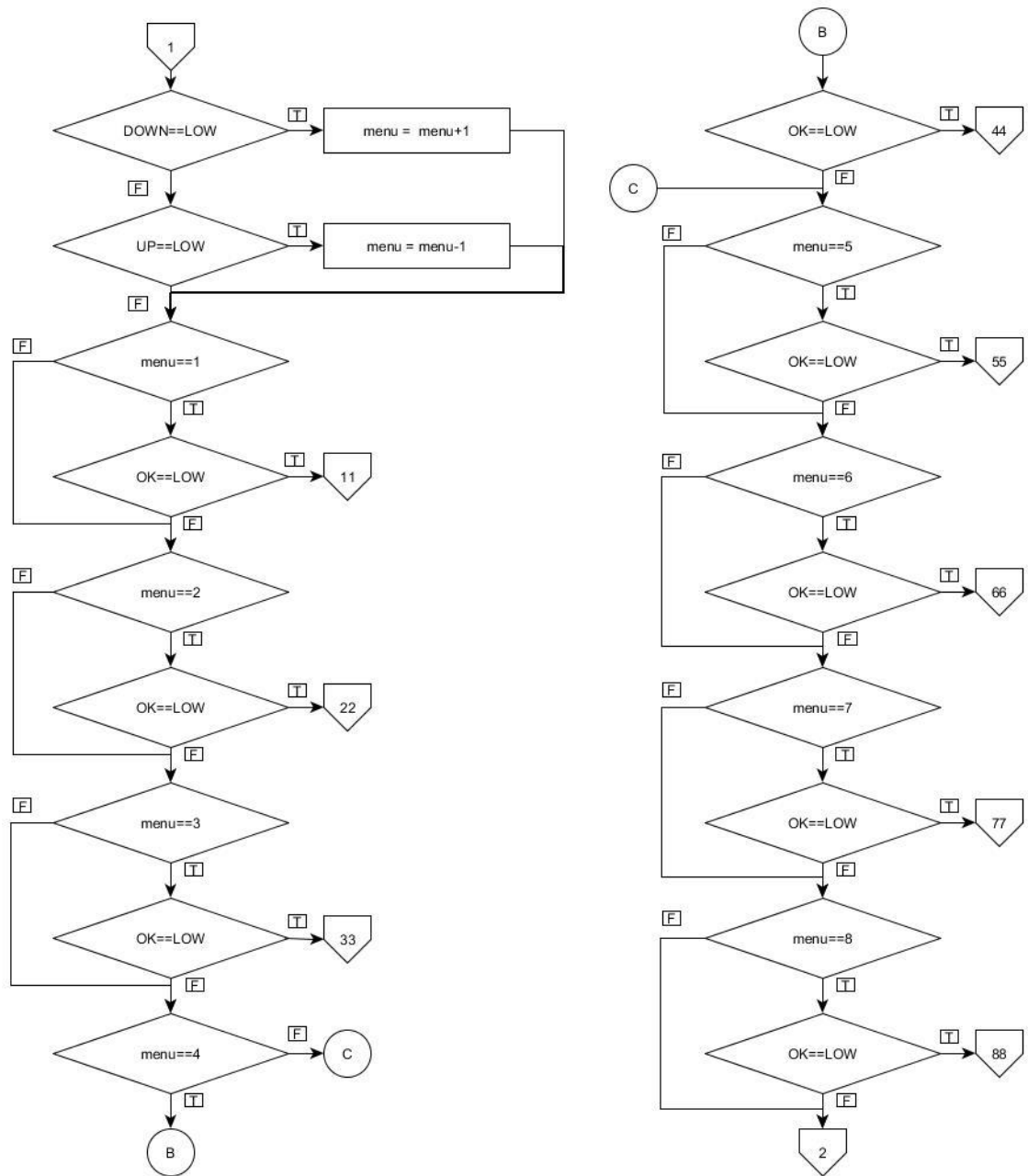
6. Flowchart Program

Berikut ini adalah diagram alir perbagian dari kontroler yang dipakai pada alat pengukur *power* tungkai, *Power Endurance*, dan *Whole Body Reaction*. Bagian-bagian yang terdefinisi antara lain meliputi *master controller*, *slave 1 controller*, *slave 2 controller*, dan *slave 3 controller*.

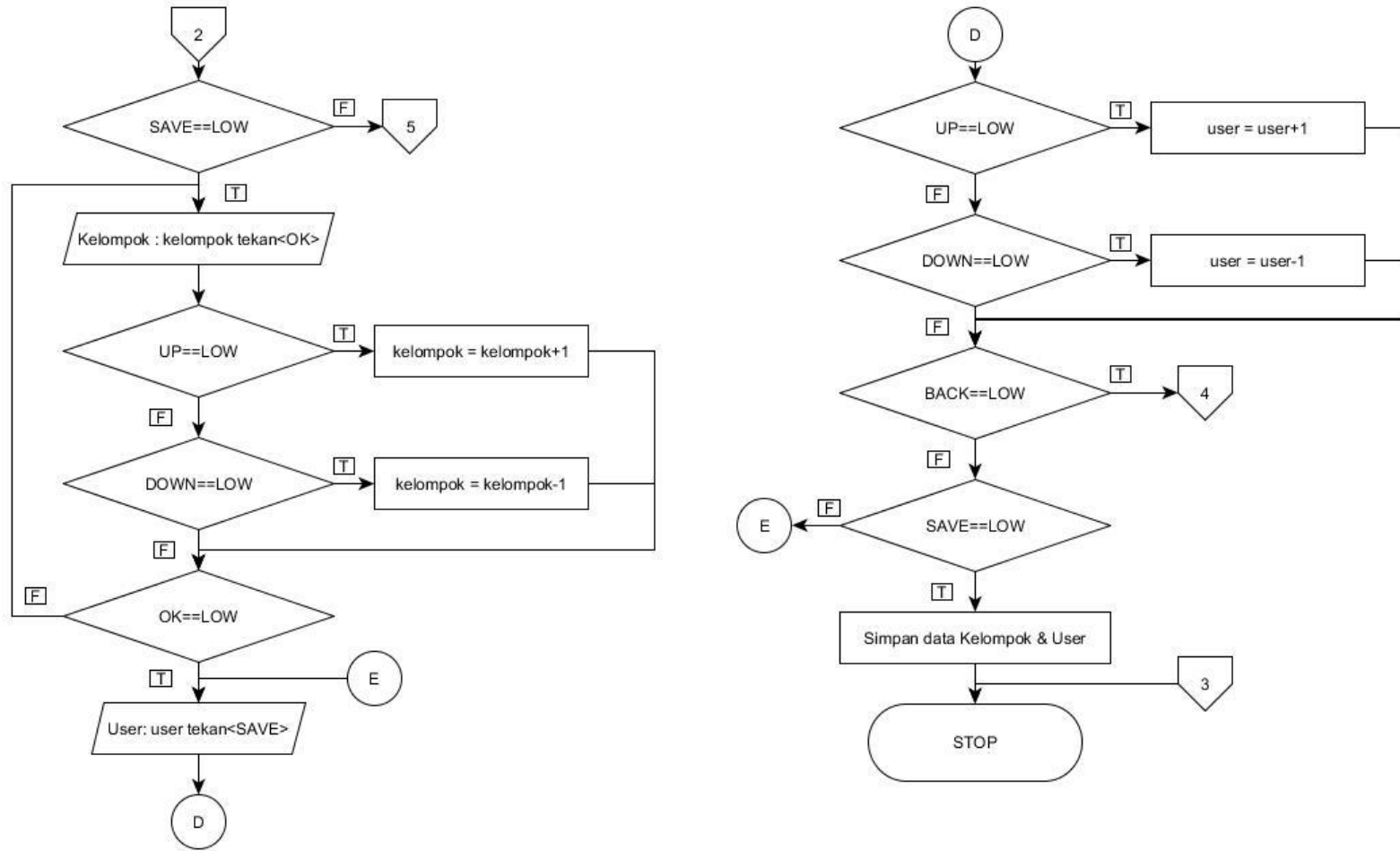
a. Flowchart Master Controller



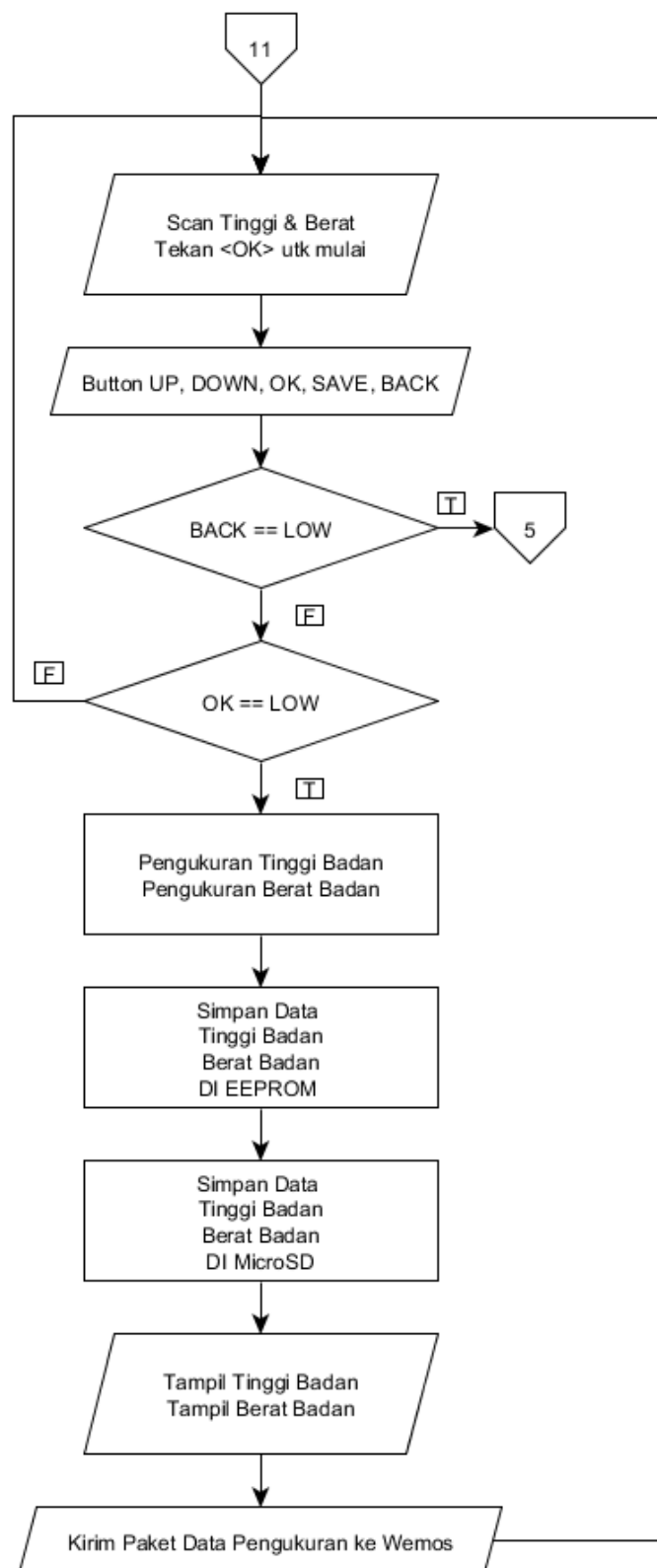
Gambar 79. Flowchart Master Controller Part1



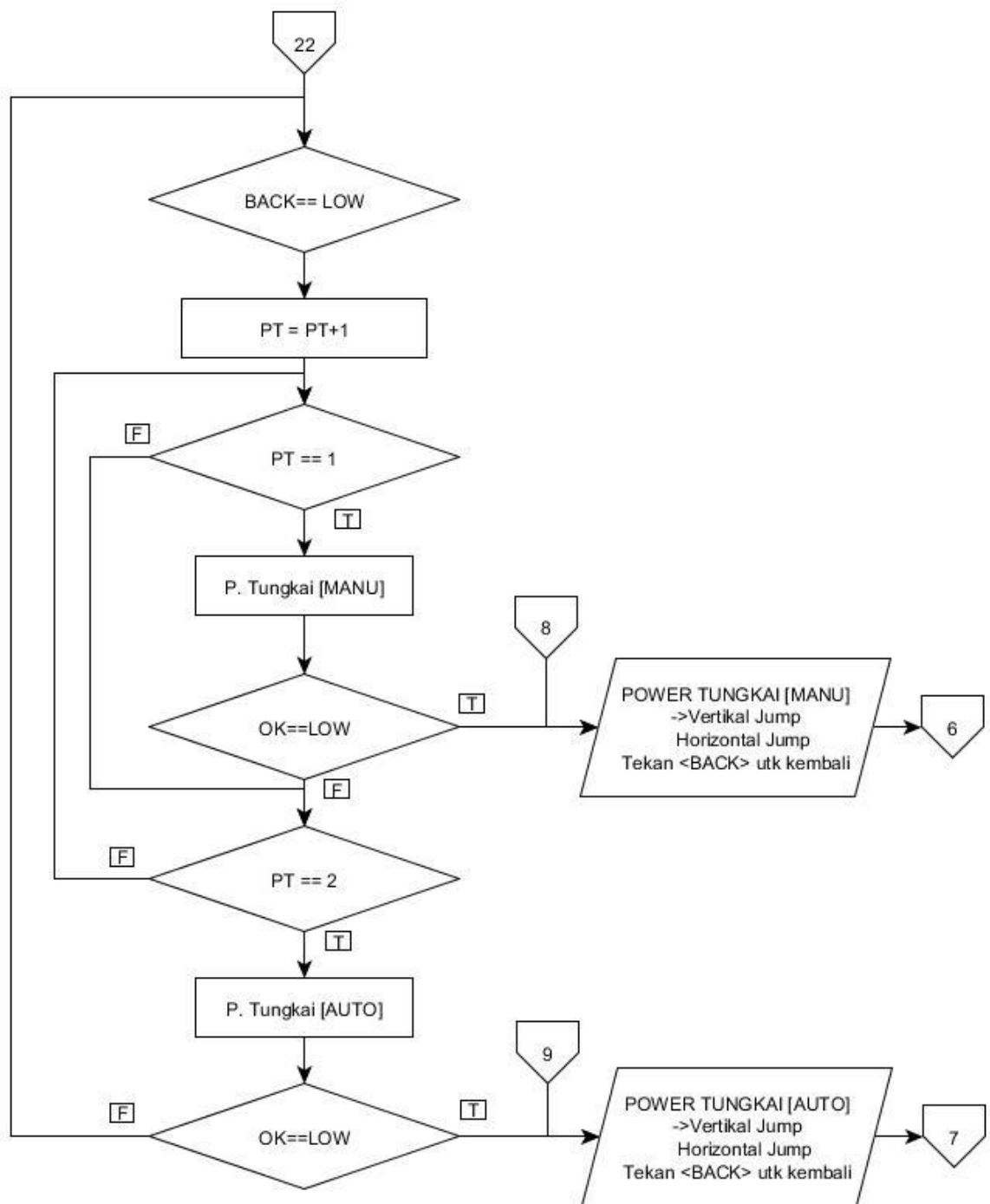
Gambar 80. Flowchart Master Controller Part2



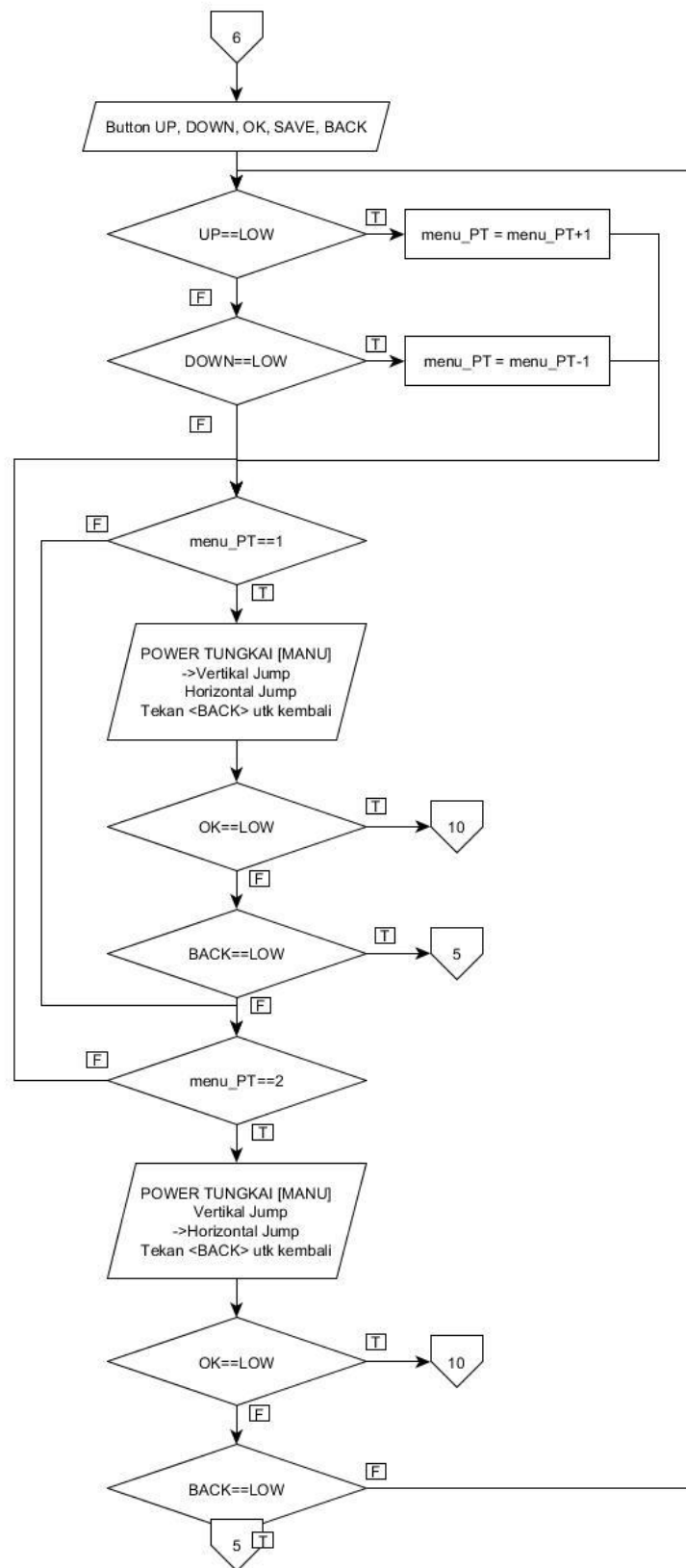
Gambar 81. Flowchart Master Controller Part3



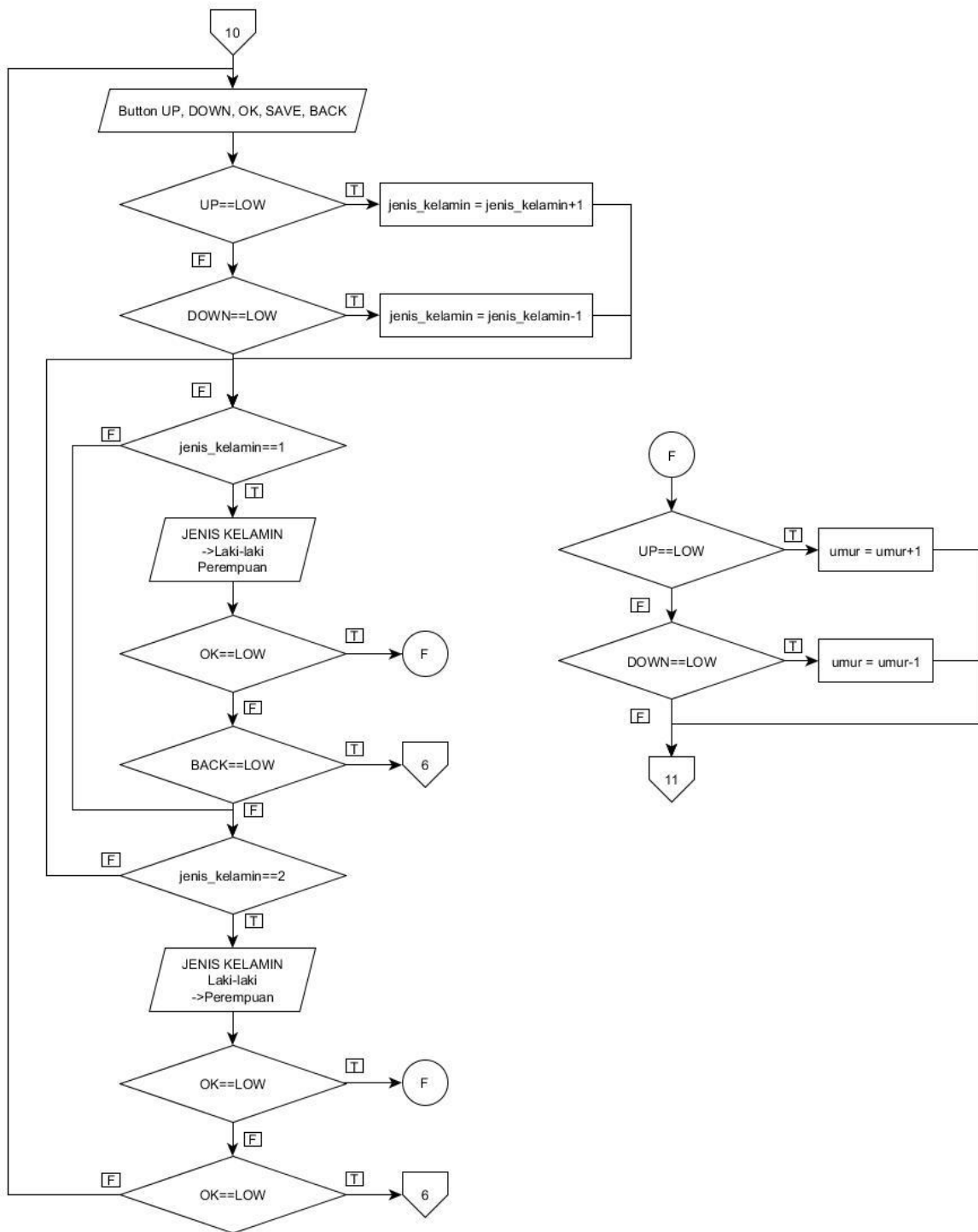
Gambar 82. Flowchart Master Controller Part4



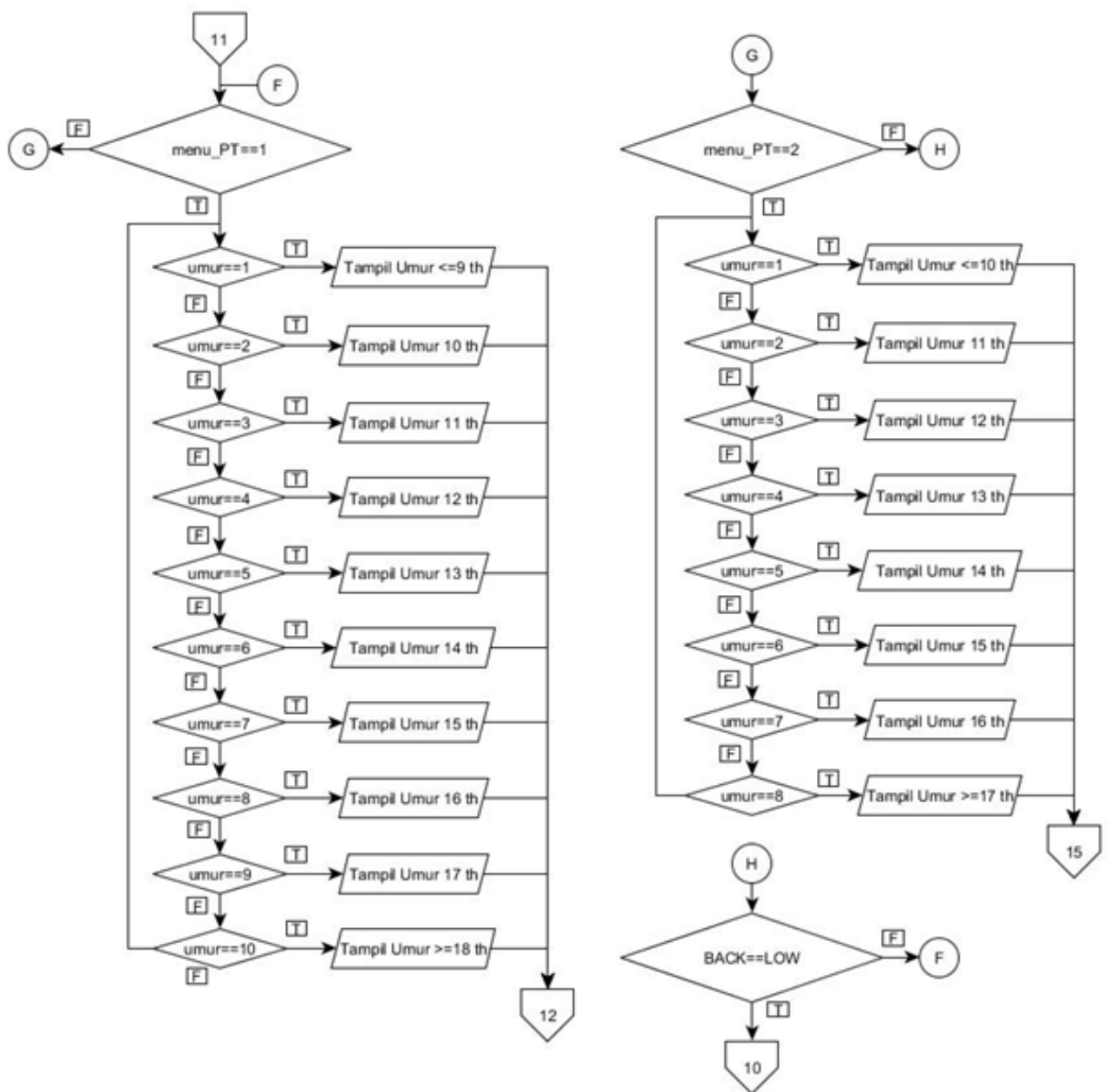
Gambar 83. Flowchart Master Controller Part5



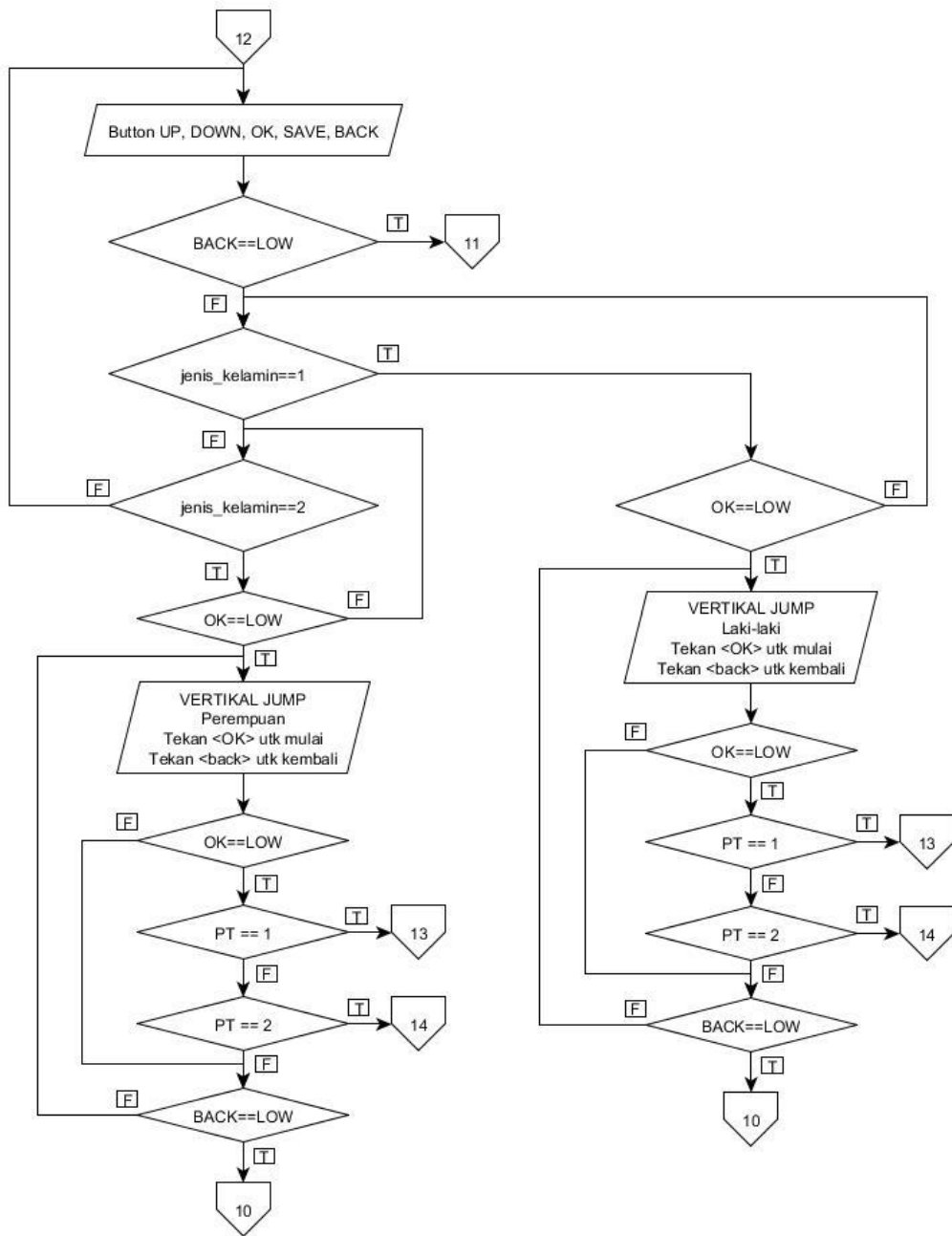
Gambar 84. Flowchart Master Controller Part6



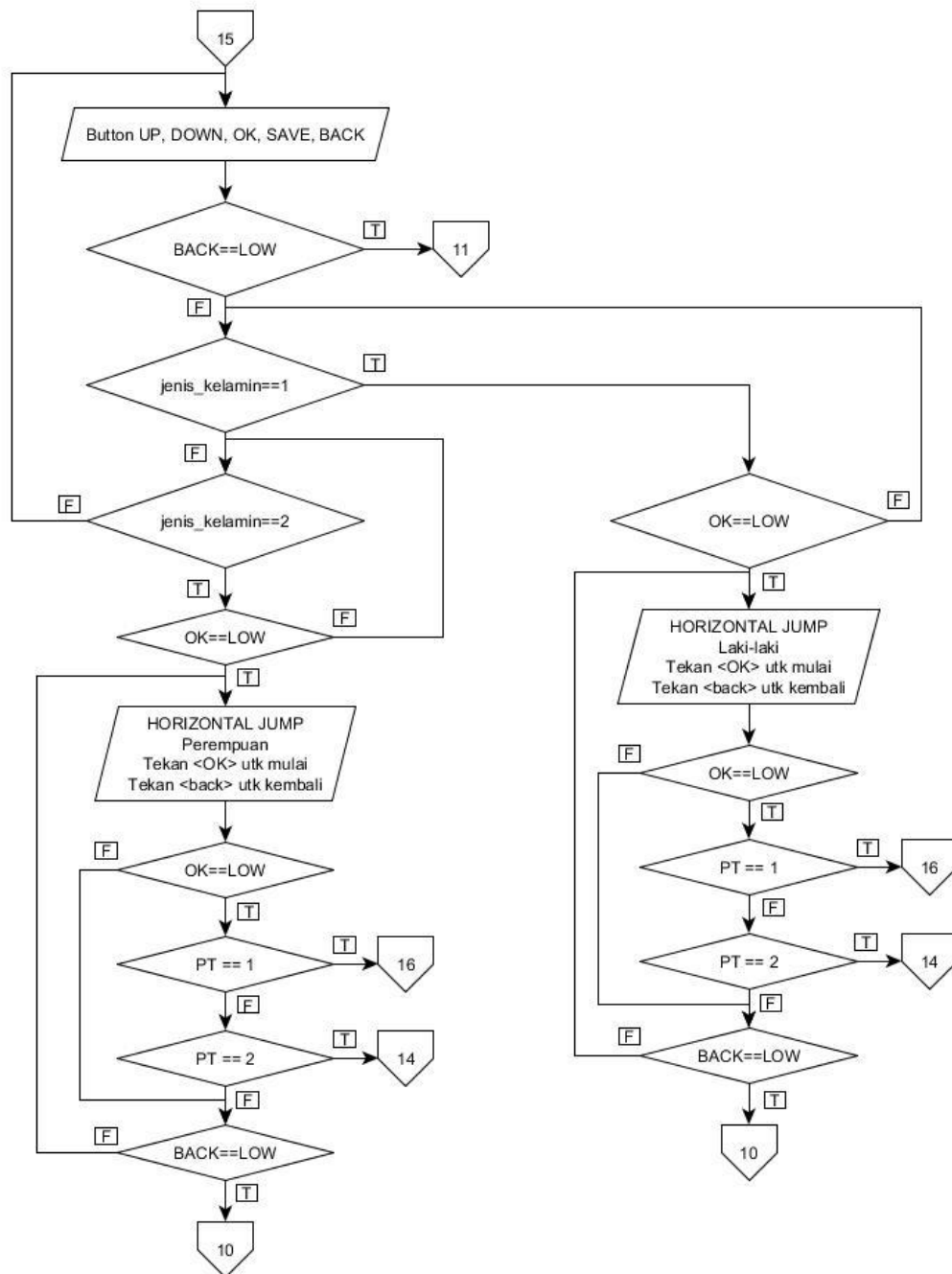
Gambar 85. Flowchart Master Controller Part7



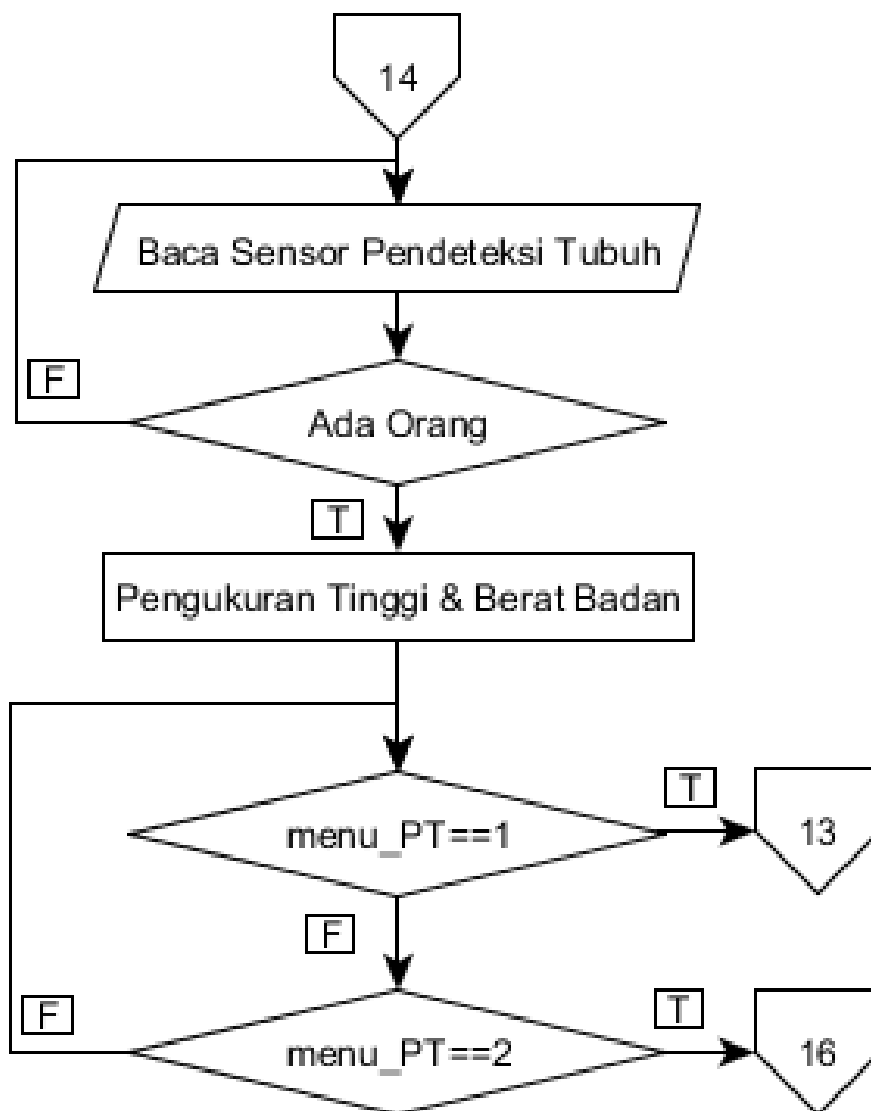
Gambar 86. Flowchart Master Controller Part8



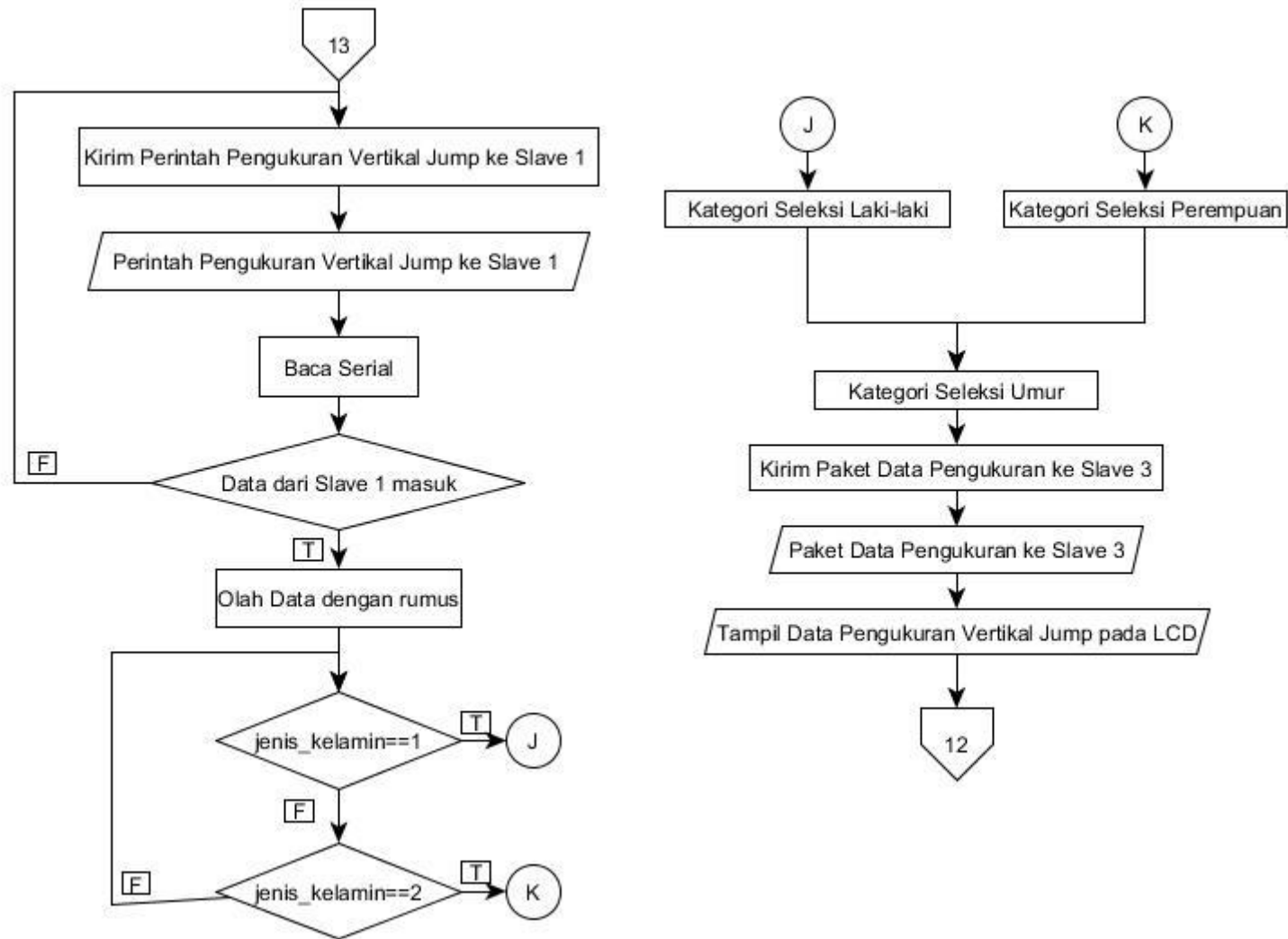
Gambar 87. Flowchart Master Kontroller Part9



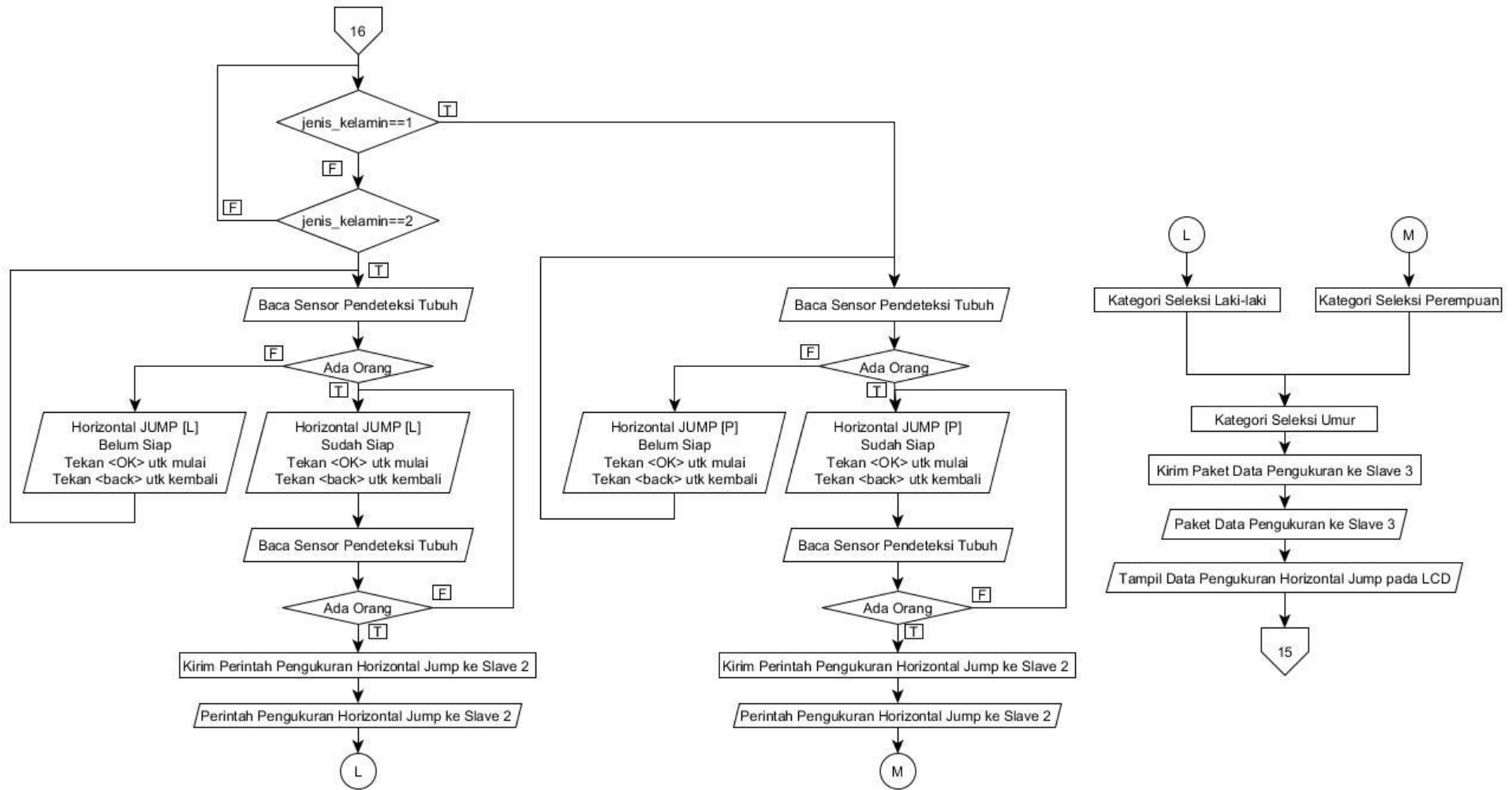
Gambar 88. Flowchart Master Controller Part10



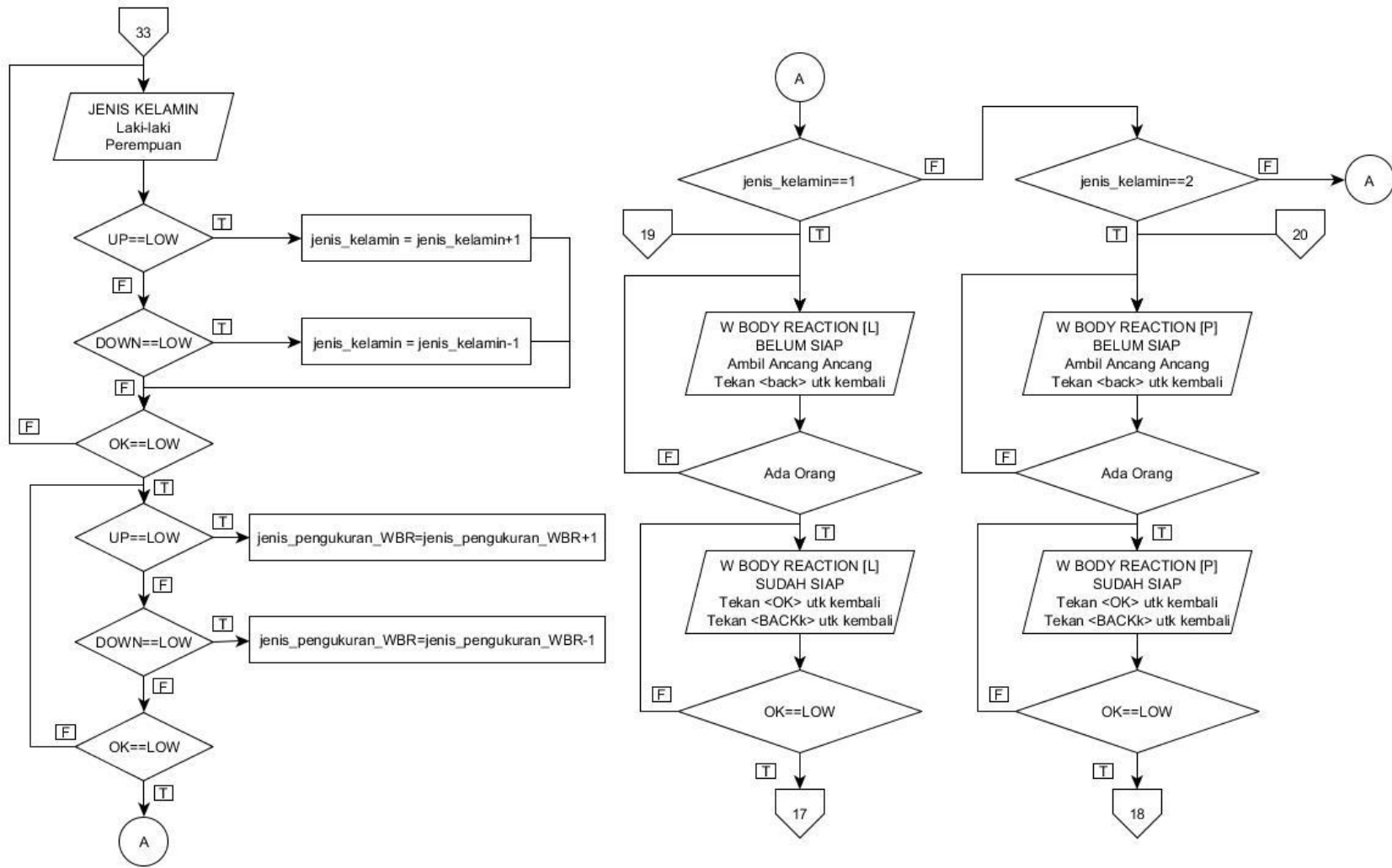
Gambar 89. *Flowchart Master Controller Part 11*



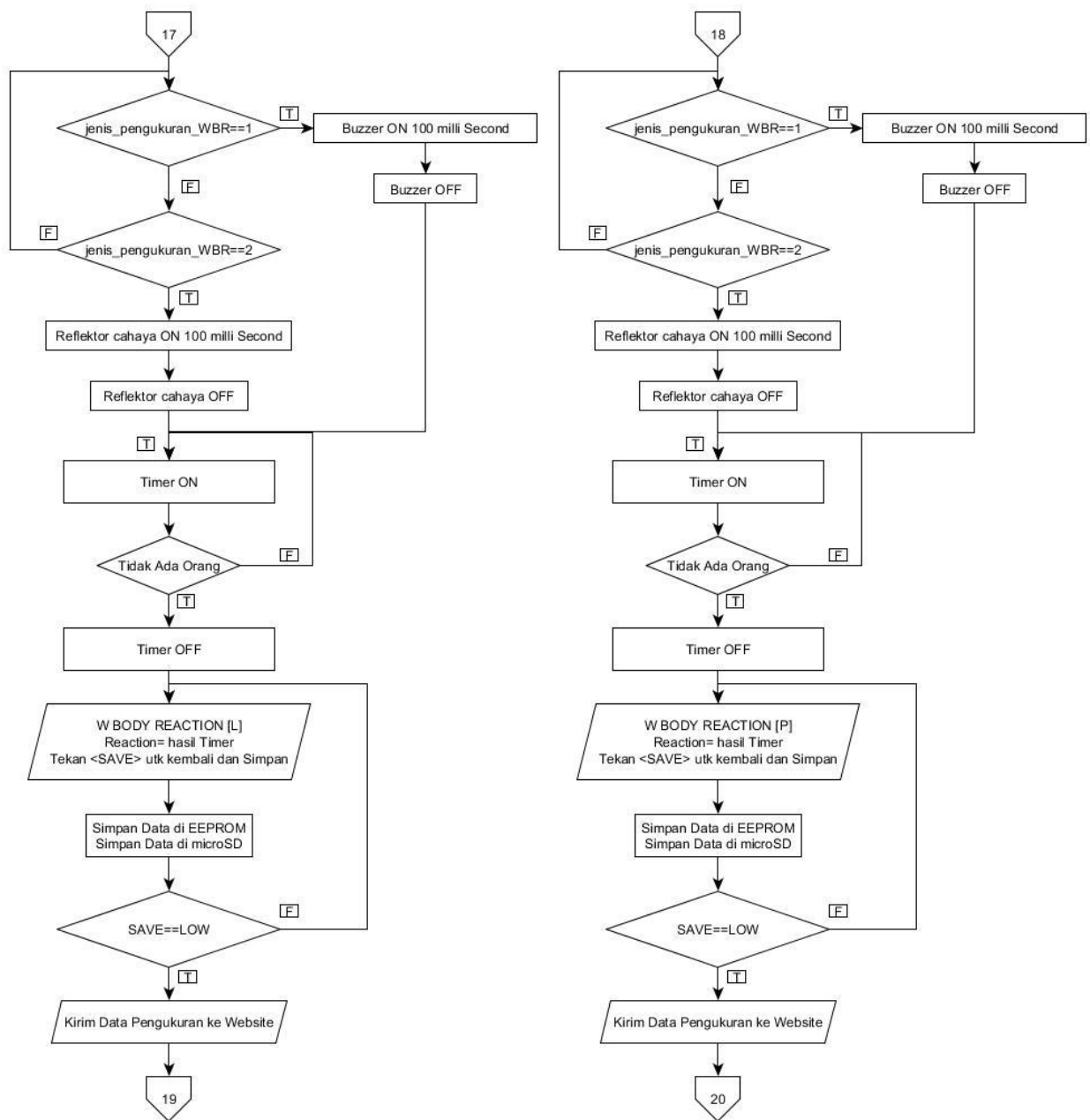
Gambar 90. Flowchart Master Controller Part12



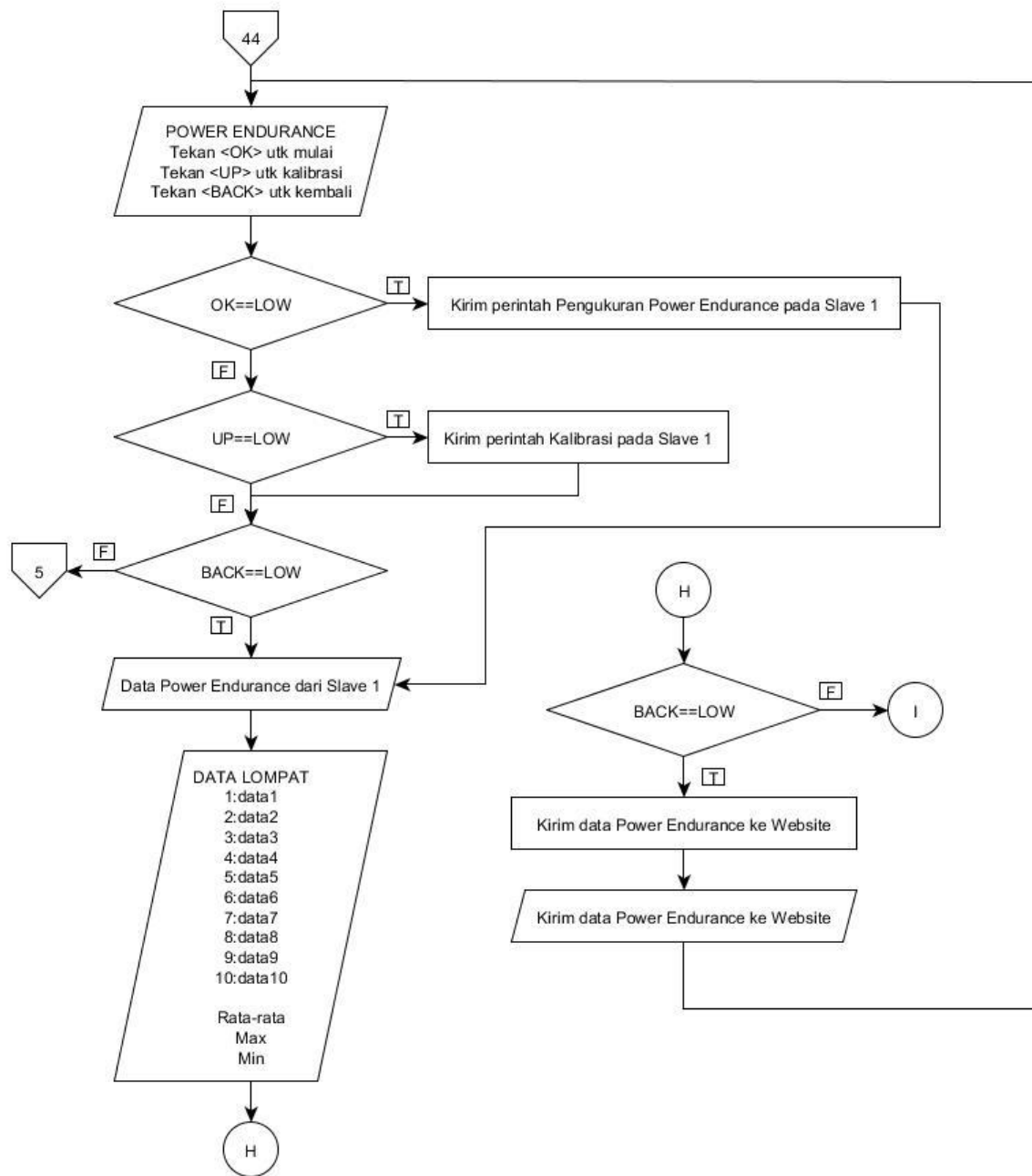
Gambar 91. Flowchart Master Controller Part13



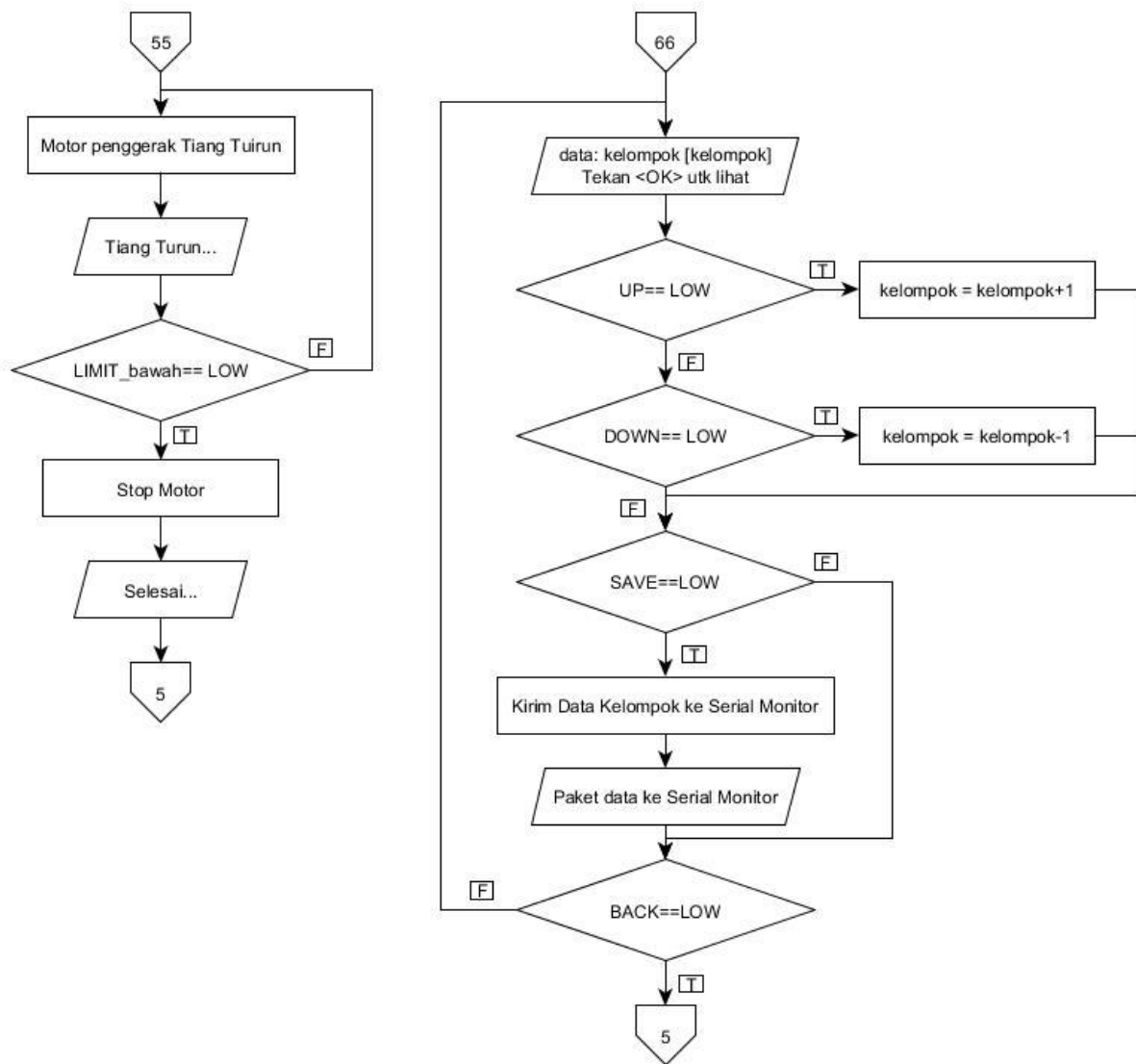
Gambar 92. Flowchart Master Controller Part14



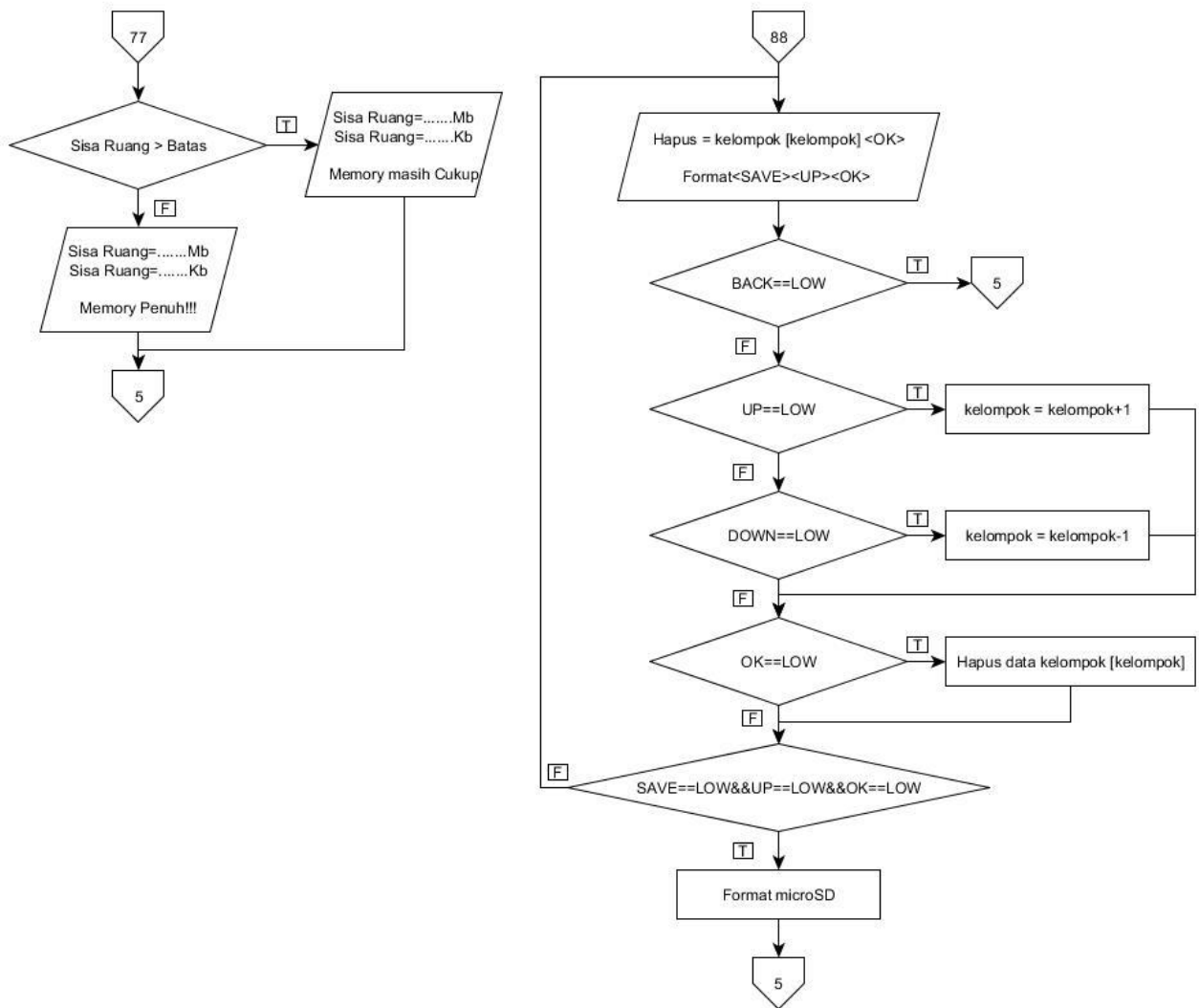
Gambar 93. Flowchart Master Controller Part15



Gambar 94. Flowchart Master Controller Part16

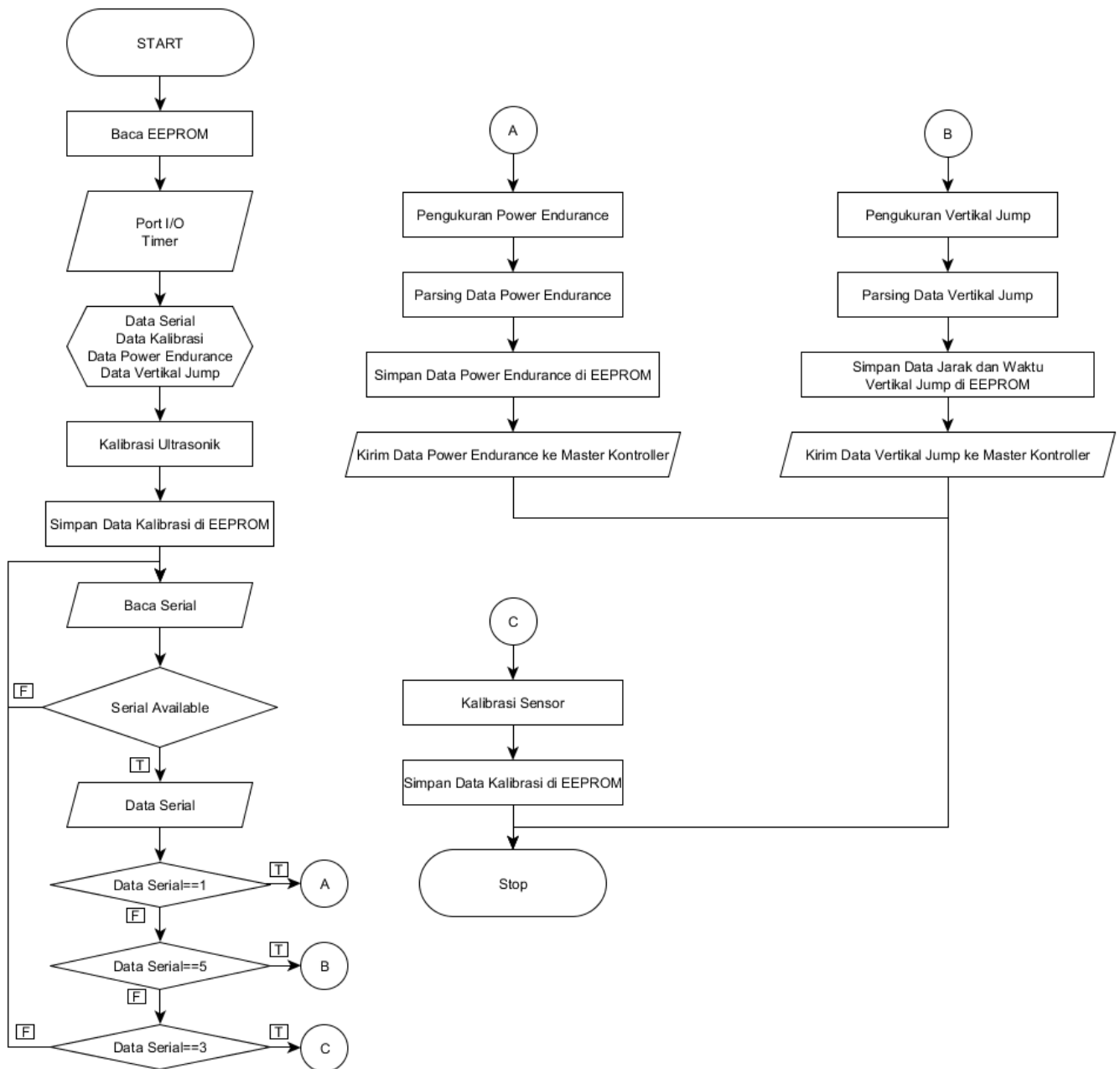


Gambar 95. Flowchart Master Controller Part17



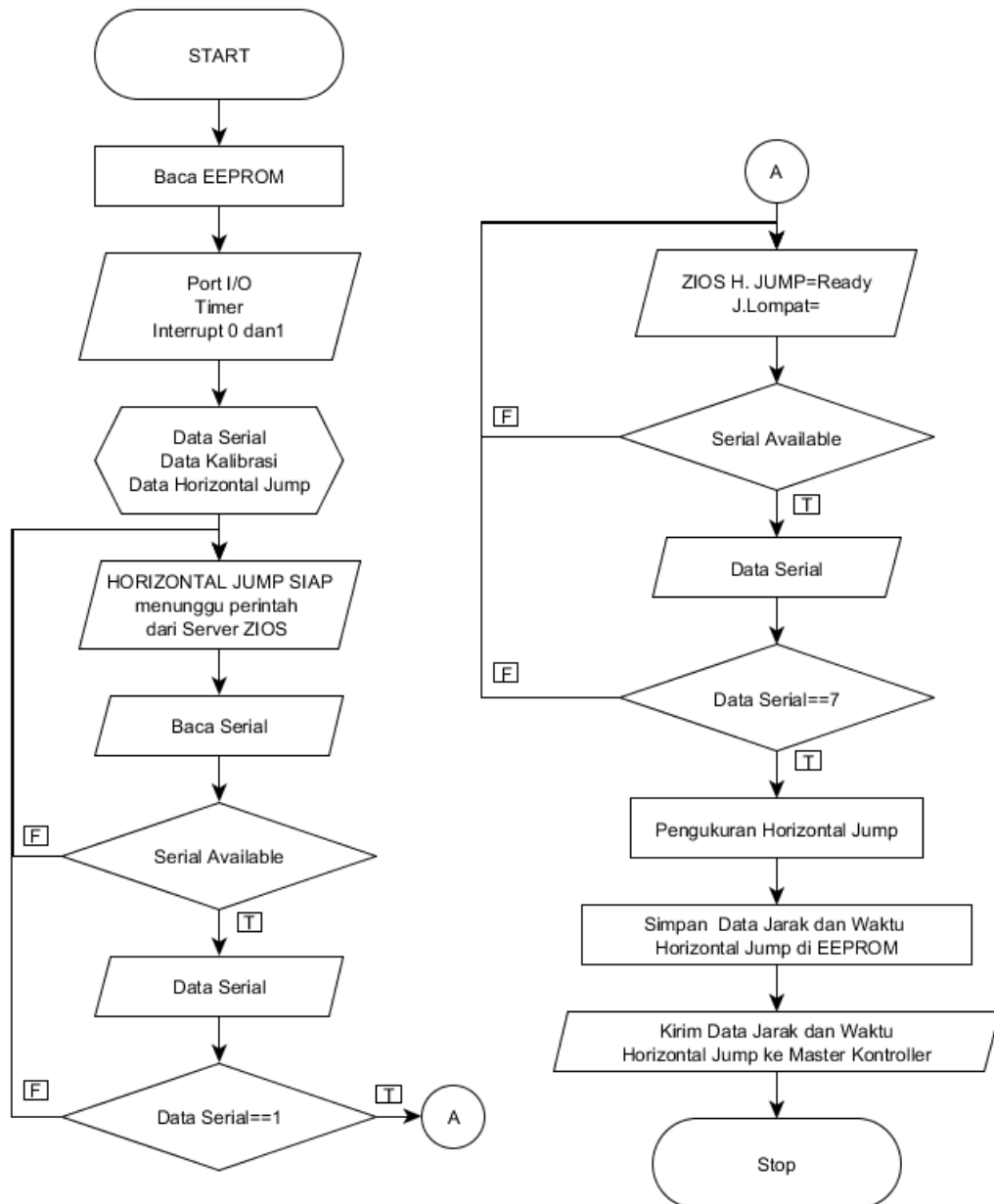
Gambar 96. Flowchart Master Controller Part18

b. *Flowchart Slave 1 Controller*

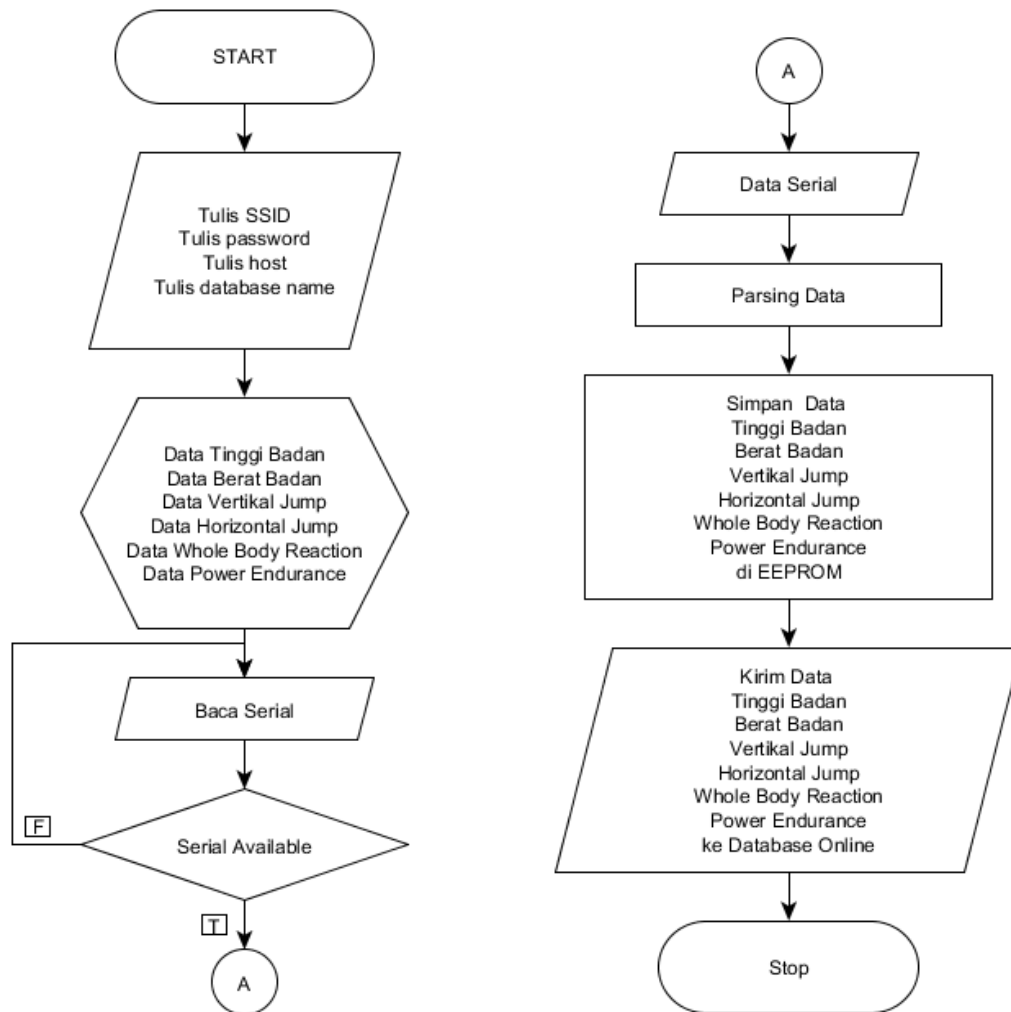


Gambar 97. *Flowchart Slave 1 Controller*

c. *Flowchart Slave 2 Controller*



Gambar 98. *Flowchart Slave 2 Controller*

d. *Flowchart Wemos Slave 3*Gambar 99. *Flowchart Slave 3*

G. Spesifikasi Alat

Spesifikasi Alat pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* pada Atlet adalah sebagai berikut:

1. *Hardware*

- a. Rangka terbuat dari bahan Alumunium $\frac{3}{4}$ inch kotak, Alumunium $\frac{3}{4}$ inch persegi, Alumunium siku (*spligot*) tebal 2 mm, Alumunium siku (*spligot*) tebal 5 mm.
- b. Dimensi base dan rangka 120 cm x 64 cm x 125 cm. walaupun besar tapi tetap ringan dan kuat karena bahan terbuat dari aluminium.
- c. Alumunium *slider* sepanjang 3 meter sebagai rel pengukur panjang lompatan.
- d. Seperangkat mekanikal *slider* sebagai alat pengukur panjang lompatan (*standing board jump*).
- e. 1 buah Box kontrol dengan dimensi 15 cm x 15 cm x 7,5 cm sebagai panel untuk menjalankan alat.
- f. 1 buah marker kaki dengan dimensi 10 cm x 7 cm x 2,5 cm sebagai alat pengukur tinggi lompatan (*vertical jump*).
- g. 1 buah motor DC *power window*.
- h. 2 buah motor DC *gearbox* 12V 400 rpm.
- i. 2 buah roda mobil-mobilan RC diameter 54 cm.
- j. 1 set *gearbox* (2 gear dan 1 rantai).
- k. 1 buah *driver* motor H-*bridge* Mosfet beserta heatsink.
- l. 1 buah *power supply switching* 12V 5 A.
- m. 1 buah kabel AC 220V.

- n. 1 buah mikrokontroler Arduino Mega .
 - o. 2 buah mikrokontroler Arduino Nano.
 - p. 1 buah mikrokontroler Wemos.
 - q. 1 buah timbangan dengan baban maksimal 180 kg.
2. *Software*
- a. Satu buah CD berisi full program alat.
 - b. Situs *website* sendiri www.smartech515.xyz
3. Media penyimpanan
- a. *Micro SD Card* 2 Gb.
 - b. EEPROM.
 - c. *Online database*.
4. Sensor
- a. 5 buah Ultrasonik.
 - b. 1 *Rotary Encoder Optical*.
 - c. 1 buah *Optocoupler*.
 - d. 8 buah *proximity*.
 - e. 1 buah *infrared sharpGP21Y0k*.
 - f. 3 buah *limit switch*.
 - g. 1 buah sensor warna.
5. Sumber tegangan
- a. *Adaptor switching* 12V 5A.
 - b. *Adaptor switching* 12V 2A
 - c. Baterai lipo 2 cell 7,8V 850 mAh.
 - d. Baterai lipo 3 cell 11,3V 2200 mAh.

H. Rencana Pengujian Alat

Pengujian alat dilakukan untuk mendapatkan data penelitian. Ada dua metode pengujian yang harus dilakukan yaitu:

1. Uji fungsional

Proses pengujian dilakukan untuk mengetahui apakah sistem yang dirancang telah bekerja dengan baik atau tidak sesuai dengan yang diharapkan. Pengujian ini juga digunakan untuk mengetahui tingkat validitas dan reabilitas dari setiap blok utama yang dijalankan. Pengujian dilakukan pada tiap blok utama pada sistem secara mandiri, lalu dilakukan pengujian terhadap sistem secara keseluruhan.

a. Pengujian Catu Daya

Pengujian catu daya dilakukan untuk memastikan tegangan keluaran tidak melebihi ambang batas tegangan kerja sensor atau komponen lainnya. Pengujian ini dilakukan dengan cara mengukur tegangan *output* catu daya. Pengujian dilakukan dengan dua metode yaitu tanpa beban dan dengan beban.

b. Pengujian Slot *micro SD*

- 1) Siapkan satu buah *micro SD* dengan kapasitas minimal 1 Gb.
- 2) Pasang *micro SD* pada slot *micro SD* yang tersedia pada box kontrol.
- 3) Lakukan sejumlah 10 kali.
- 4) Catat hasilnya dan amati perubahan pada *display*.

c. Pengujian perangkat pengukur Berat Badan

- 1) Siapkan *prototype* timbangan yang sudah di oprek dan timbangan standar yang masih orisinil pabrik.
 - 2) Siapkan media pembanding (pastikan beban fix yang standar).
 - 3) Objek diletakkan diatas timbangan *prototype*.
 - 4) Mulai menimbang Objek pembanding terlebih dahulu.
 - 5) Catat hasil perbandingan pengukuran yang dilakukan.
 - 6) Objek diletakkan diatas timbangan orisinil.
 - 7) Mulai menimbang Objek pembanding ke timbangan orisinil.
 - 8) Catat hasil perbandingan pengukuran yang dilakukan.
 - 9) Mengulang langkah 3 sampai 8 sebanyak 10 kali dengan media pembanding yang berbeda.
 - 10) Amati hasil perbandingan pengukuran yang dilakukan menggunakan dua alat tersebut.
 - 11) Pastikan selisih pengukuran antara timbangan *prototype* dan timbangan orisinil bernilai 0 atau mendekati 0.
- d. Pengujian perangkat pengukur Tinggi Badan
- 1) Siapkan *prototype* alat
 - 2) Siapkan meteran standar sebagai pembanding.
 - 3) Posisikan tubuh objek pada titik tumpuan yang sudah ditandai.
 - 4) Mulai ukur tinggi badan Objek menggunakan *prototype*.
 - 5) Catat hasilnya.
 - 6) Kemudian ukur tinggi badan Objek menggunakan meteran.
 - 7) Catat hasilnya.

- 8) Mengulang langkah 3 sampai 7 sebanyak 10 kali dengan media Objek yang berbeda.
 - 9) Amati hasil perbandingan pengukuran yang dilakukan menggunakan dua alat tersebut.
 - 10) Pastikan selisih pengukuran keduanya bernilai 0 atau mendekati 0.
- e. Pengujian perangkat pengukur panjang lompatan
- 1) Siapkan *prototype* alat.
 - 2) Siapkan meteran standar sebagai pembanding.
 - 3) Letakkan meteran standar sejajar dengan rel pengukur.
 - 4) Mulai ukur panjang lompatan seseorang menggunakan *prototype* dan menggunakan meteran secara bersamaan.
 - 5) Catat hasil perbandingan pengukuran yang dilakukan menggunakan dua alat tersebut.
 - 6) Amati hasil perbandingan pengukuran yang dilakukan menggunakan dua alat tersebut.
 - 7) Pastikan selisih pengukuran keduanya bernilai 0 atau mendekati 0.
- f. Pengujian perangkat pengukur tinggi lompatan
- 1) Siapkan *prototype* alat.
 - 2) Siapkan meteran standar sebagai pembanding.
 - 3) Tempelkan meteran pada tembok.
 - 4) Mulai ukur tinggi lompatan seseorang menggunakan *prototype* dan menggunakan meteran secara bersamaan.
 - 5) Catat hasil perbandingan pengukuran yang dilakukan menggunakan dua alat tersebut.

- 6) Amati hasil perbandingan pengukuran yang dilakukan menggunakan dua alat tersebut.
 - 7) Pastikan selisih pengukuran keduanya bernilai 0 atau mendekati 0.
- g. Pengujian perangkat pengukur *Whole Body Reaction*
- 1) Siapkan *prototype* alat.
 - 2) Siapkan stopwatch sebagai pembanding.
 - 3) Posisikan tubuh objek pada titik tumpuan yang sudah ditandai.
 - 4) Mulai pengukuran *Whole Body Reaction* menggunakan *prototype* dan menggunakan Stopwatch secara bersamaan.
 - 5) Catat hasil perbandingan pengukuran yang dilakukan menggunakan dua alat tersebut.
 - 6) Amati hasil perbandingan pengukuran yang dilakukan menggunakan dua alat tersebut.
 - 7) Pastikan selisih pengukuran keduanya bernilai 0 atau mendekati 0.
- h. Pengujian sistem penyimpanan
- Pengujian sistem penyimpanan dilakukan untuk memastikan bahwa data yang telah didapat dari setiap percobaan dapat tersimpan di *micro SD card*. Pengujian sistem penyimpanan dilakukan dengan cara:
- 1) Pastikan sebelumnya *micro SD* dalam keadaan kosong (Format terlebih dahulu).
 - 2) Hidupkan *prototype* dan mulai ambil data mulai dari Tinggi Badan, Berat Badan, *Power Tungkai (Standing board jump dan Vertikal jump)*, *Whole Body Reaction*, dan *Power Endurance*.
 - 3) Setelah pengambilan data selesai, jangan langsung matikan alat.

- 4) Pada saat masih pada tampilan menu pencet tombol *SAVE*.
 - 5) Kemudian akan keluar tulisan “Kelompok: 1 tekan <OK>”. Untuk merubah lokasi kelompok penyimpanan tekan *UP* atau *DOWN*, setelah itu tekan *OK*.
 - 6) Berikutnya akan keluar tulisan “User: 1 tekan <SAVE>”. Untuk merubah lokasi user penyimpanan tekan *UP* atau *DOWN*, setelah itu tekan *SAVE*.
 - 7) Akan muncul tulisan “Simpan” pada lcd, berarti proses penyimpanan data telah selesai dilakukan.
 - 8) Selanjutnya cabut *micro SD* card dari alat, kemudian masukkan pada laptop. Buka penyimpanan *micoSD*, maka akan terlihat *file* bernama “Kelompok 1.txt” didalamnya akan berisi data yang tadi telah diambil dengan diberi label nama “User 1”.
 - 9) Mengulangi langkah 4 sampai 8 dengan pengambilan data yang berbeda.
- i. Pengujian komunikasi dan *Pairing Bluetooth* antara *Master Controller* dengan *Slave* kontroller. Pengujian komunikasi *Bluetooth* sangatlah penting bagi alat ini, Karena sistem komunikasi utama alat ini adalah menggunakan *serial wireless communication*. Pengujian Komunikasi dilakukan dengan cara melakukan pengiriman data acak dari *Master Controller* ke *Slave 1 Controller*, *Slave 2 Controller*, dan *Slave 3 Kontroller*. Sedangkan pengujian *Pairing* dilakukan dengan cara melihat indicator *LED Bluetooth* yang ada pada box kontroller.

Pengujian dilakukan antar perangkat *Master Controller* dengan *Slave 1* dan *Master Controller* dengan *Slave 2*.

j. Pengujian sistem telemetri

Pengujian sistem telemetri ini dilakukan untuk memastikan data yang di upload ke *website* secara *online* tidak mengalami *loss connection*.

Adapun langkah-langkahnya adalah sebagai berikut:

- 1) Siapkan modem wifi “ToKoZik” sebagai router.
- 2) Pastikan modem masih mempunyai kuota data.
- 3) Buka *website* www.smartech515.xyz
- 4) Masuk ke menu monitoring.
- 5) Kemudian mulai pengukuran menggunakan alat.
- 6) Misal pengukuran Tinggi dan Berat badan.
- 7) Pada saat akhir pengukuran dan saat tekan tombol *SAVE* atau *BACK*, akan muncul tulisan “Kirim Data ke Web”. Berarti alat sudah mengirim data Tinggi dan Berat badan ke *website*
- 8) Lakukan refresh pada halaman *website* tadi yang telah dibuka dengan menekan F5 pada keyboard atau tunggu selama 30 detik, sebab web ini sudah memiliki fasilitas auto refresh.
- 9) Lihat pada kolom TB dan BB pada menu monitoring, apakah ada perubahan sesuai dengan pengukuran pada alat atau tidak.
- 10) Catat hasilnya.
- 11) Mengulang langkah 5 sampai 10 dengan pengukuran yang berbeda-beda.

k. Pengujian daya tahan Baterai lipo

Pengujian baterai lipo ini dilakukan untuk melihat daya tahan baterai saat alat digunakan dengan cara mengukur tegangan baterai saat tidak ada beban dan saat ada beban, dilakukan secara berkala setiap kelipatan 10 menit.

2. Uji unjuk kerja

Proses pengujian unjuk kerja dilakukan untuk mengetahui apakah sistem yang dirancang telah bekerja dengan baik atau tidak sesuai dengan yang diharapkan. Hal-hal yang perlu diamati antara lain: sensor tinggi badan, sensor berat badan, mekanisme pengukur panjang lompatan, sensor tinggi lompatan, sensor *Whole Body Reaction*, slot *mico SD*, *push button*, tampilan LCD, tampilan data pada *Website*, dan hasil pengolahan rumus-rumus. Sehingga apa yang diuji dapat diketahui bagaimana kinerja setiap bagian bagiannya.

I. Tabel Uji Alat

1. Pengujian Catu Daya

a. Pengujian Tanpa Beban

Tabel 20. Rencana Pengukuran Catu Daya Tanpa Beban

No	Pengukuran Pada	Jumlah pengukuran	V out	V out terbaca	Error	Selisih
1.	<i>Power supply Switching 12 V 5A.</i>	1	12			
		2	12			
		3	12			
		4	12			
		5	12			
2.	<i>Power supply Switching 12 V 2A.</i>	1	12			
		2	12			
		3	12			
		4	12			
		5	12			

3.	Modul XL-4015 pada Arduino Mega	1	5			
		2	5			
		3	5			
		4	5			
		5	5			

b. Pengujian Dengan Beban

Tabel 21. Rencana Pengukuran Catu Daya Dengan Beban

No	Pengukuran Pada	Jumlah pengukuran	V out	V out terbaca	Error	Ket
1.	<i>Power supply Switching 12 V 5A.</i>	1	12			
		2	12			
		3	12			
		4	12			
		5	12			
2.	<i>Power supply Switching 12 V 2A.</i>	1	12			
		2	12			
		3	12			
		4	12			
		5	12			
3.	Modul XL-4015 pada Arduino Mega	1	5			
		2	5			
		3	5			
		4	5			
		5	5			

2. Pengujian Slot *Micro SD Adapter*

Tabel 22. Rencana pengujian Slot *Micro SD*

No	Percobaan	Error / Sukses	Tampilan LCD	Keterangan
1	I			
2	II			
3	III			
4	IV			
5	V			
6	VI			
7	VII			
8	VIII			
9	IX			

10	X			
----	---	--	--	--

3. Pengujian *Loadcell***Tabel 23.** Rencana Pengujian *Loadcell* terhadap Berat Badan

No	Hasil Timbangan Orisinil (Kg)	Hasil Ukur Alat (Kg)	Selisih (Kg)	Error (%)
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

4. Pengujian Perangkat pengukur Tinggi Badan

Tabel 24. Rencana Pengukuran Tinggi Badan

No	Tinggi Sampel Acak (Cm)	Hasil Ukur Alat (Cm)	Selisih (Cm)	Error (%)
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

5. Pengujian Perangkat pengukur Panjang Lompatan

Tabel 25. Rencana Pengukuran Panjang Lompatan

No	Hasil Ukur Meteran Acak (Cm)	Hasil Ukur Alat (Cm)	Selisih (Cm)	Error (%)
1				

2				
3				
4				
5				
6				
7				
8				
9				
10				

6. Pengujian Ultrasonik SRF04 terhadap Tinggi Lompatan

Tabel 26. Rencana Pengukuran Ultrasonik SRF04 terhadap Tinggi Lompatan

No	Hasil Ukur Meteran (Cm)	Hasil Ukur Alat (Cm)	Selisih (Cm)	<i>Error (%)</i>
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

7. Pengujian perangkat pengukur *Whole Body Reaction*

Tabel 27. Rencana Pengujian perangkat pengukur *Whole Body Reaction*

No	Hasil Ukur Stopwatch (detik)	Hasil Ukur Alat (detik)	Selisih (detik)	<i>Error (%)</i>
1				
2				
3				
4				
5				
6				
7				
8				

9				
10				

8. Pengujian sistem penyimpanan *micro SD***Tabel 28.** Rencana pengujian sistem penyimpanan *micro SD*

No	Jenis pengukuran yang dilakukan	Hasil Ukur	Data di EEprom	Data di <i>Micro SD</i>	Valid/Tidak Valid
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

9. Pengujian Komunikasi dan *Pairing Bluetooth***Tabel 29.** Rencana pengujian Komunikasi *Bluetooth*

No	Data dikirim dari <i>Master</i> kontroler	Data yang diterima <i>Slave 1</i> kontroler	Data yang diterima <i>Slave 2</i> kontroler	<i>Error (%)</i>	Valid/Tidak Valid
1					
2					
3					
4					
5					

Tabel 30. Rencana Pengujian *Pairing Bluetooth*

No	Koneksi Antara	Percobaan ke-	Tersambung / Tidak Tersambung	Keterangan
1	<i>Bluetooth Slave 1</i> dengan <i>Master Controller</i>	I		
		II		
		III		
		IV		
		V		

2	<i>Bluetooth Slave 2 dengan Master Controller</i>	I		
		II		
		III		
		IV		
		V		

10. Pengujian Sistem Telemetry

Tabel 31. Rencana pengujian sistem Telemetry

No	Jenis pengukuran yang dilakukan	Hasil Ukur Alat	Data di EEprom	Data di Website	Keterangan
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

11. Pengujian *Battery*

Pengukuran *battery* dilakukan pada *output battery*. Pengukuran *battery* dilakukan sebanyak 5 kali untuk memastikan kondisi *battery* dalam keadaan baik.

a. Pengujian Tanpa Beban

Tabel 32. Rencana Pengukuran *battery* lipo 12,6 volt 2,2A Tanpa Beban

No	Waktu (menit)	V Out (V)
1	10	
2	20	
3	30	
4	40	
5	50	

6	60	
7	70	
8	80	
9	90	
10	100	

Tabel 33. Rencana Pengukuran *battery* lipo 8,4 volt 850 mAh Tanpa

Beban

No	Waktu (menit)	V Out (V)
1	10	
2	20	
3	30	
4	40	
5	50	
6	60	
7	70	
8	80	
9	90	
10	100	

b. Pengujian Dengan Beban

Tabel 34. Rencana Pengukuran *battery* lipo 12,6 volt 2,2A Dengan

Beban

No	Waktu (menit)	V Out (V)
1	10	
2	20	
3	30	
4	40	
5	50	
6	60	
7	70	
8	80	
9	90	
10	100	

Tabel 35. Rencana Pengukuran *battery* lipo 8,4 volt 850 mAh Dengan Beban

No	Waktu (menit)	V Out (V)
1	10	
2	20	
3	30	
4	40	
5	50	
6	60	
7	70	
8	80	
9	90	
10	100	

J. Pengoperasian Alat

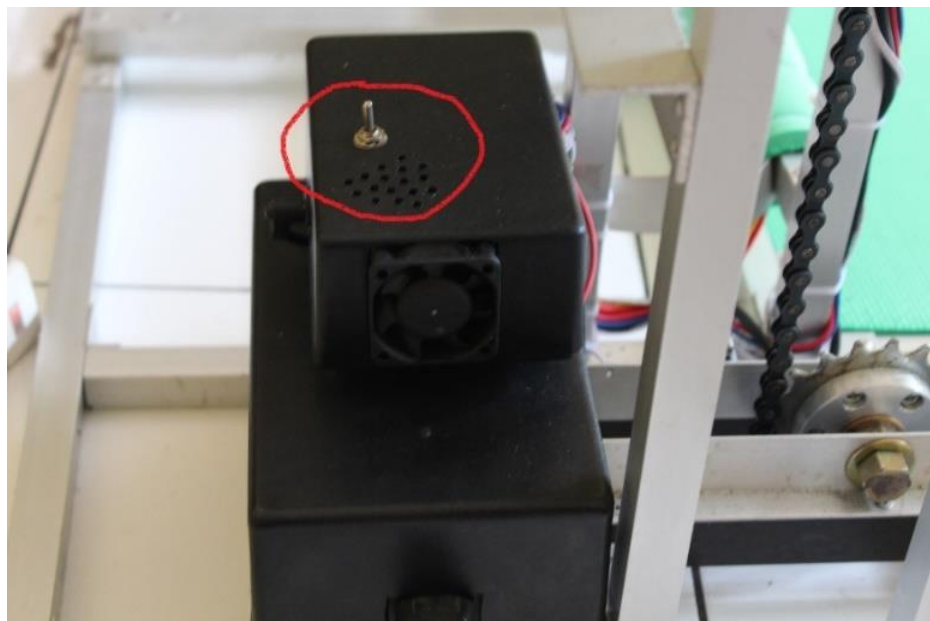
Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* pada Atlet ini terbagi menjadi 3 perangkat terpisah penggunaannya tapi tetap dalam satu sistem yang terintegrasi. Adapun cara pengoperasiannya adalah sebagai berikut:

1. Colokkan soket adaptor pada tegangan listrik 220VAC.
2. Masukkan *Micro SD* pada slot yang ada di box kontrol.
3. Hidupkan tombol Emergency berwarna merah.



Gambar 100. Tombol *Emergency*

4. Hidupkan *Driver* motor dengan cara menggeser saklar toggle diatas box Diver Motor.



Gambar 101. Saklar *Driver* Motor

5. Tancapkan soket 1 pada port 1 seperti terlihat pada gambar dibawah.



Gambar 102. Soket 1

6. Tancapkan Jack DC 12V.



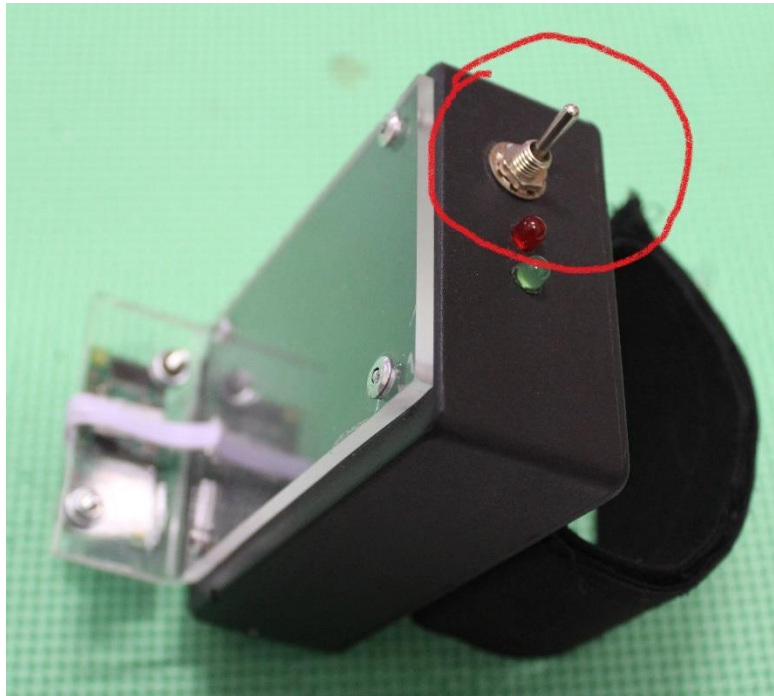
Gambar 103. Jack DC 12V

7. Tancapkan soket 2 pada port 2 seperti terlihat pada gambar dibawah.



Gambar 104. Soket 2

8. Hidupkan marker (*Slave 1 Controller*) dengan cara menggeser saklar toggle yang ada pada marker.



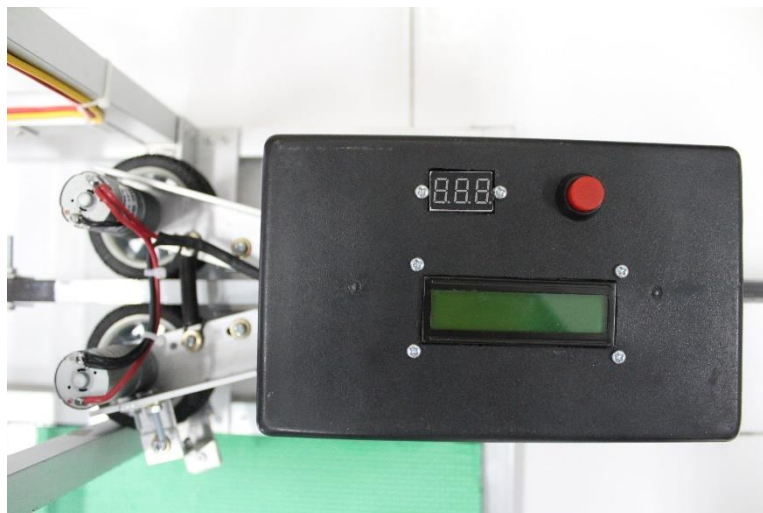
Gambar 105. Saklar *Slave 1 Controller*

9. Pasang marker di bagian mata kaki kanan/kiri. Hadapkan sensor marker ke arah lantai dengan jarak antara 3 sampai 5 cm untuk kalibrasi.



Gambar 106. Sensor Pengukur Tinggi Lompatan pada *Slave 1 Controller*

10. Tunggu sampai led berkedip dan *buzzer* menyala tiga kali.
11. Hidupkan *Slave 2 Controller* dengan cara menekan tombol merah yang ada pada alat.



Gambar 107. Kondisi pada saat *Slave 2 Controller* masih mati



Gambar 108. Kondisi pada saat *Slave 2 Controller* sudah hidup

12. Hidupkan *Master Controller* dengan cara menekan tombol *power* yang ada di sebelah samping kanan box kontrol.



Gambar 109. Tombol *Power* pada *Master Controller*

13. Tampilan pada saat Proses Booting.



Gambar 110. Tampilan pada saat proses Booting

14. Menu StanBy pada Box Kontrol.



Gambar 111. Tampilan Menu StanBy

15. Tekan OK/MENU untuk masuk menu.

16. Tampilan Menu pada *slide* pertama.



Gambar 112. Tampilan Menu *slide* satu

17. Tampilan Menu pada *slide* kedua.



Gambar 113. Tampilan Menu *slide* dua

18. Buka *website* www.smartech515.xyz dan dan masuk menu monitoring untuk melihat data hasil pengukuran.
19. Pilih menu pengukuran pada *slide* satu dengan menekan tombol “OK”. Berikut hal-hal yang perlu diperhatikan sebagai berikut:
 - a. *Scan* Berat & Tinggi
 - 11) Testi naik pada timbangan dan sejajar lurus dengan posisi alat.
 - 12) Badan tegap dan kaki harus rapat.
 - 13) Lalu testor menekan tombol “OK” pada menu.
 - 14) Tunggu sampai tiang pengukur tinggi badan turun dan menyentuh ujung kepala.
 - 15) Setelah itu tunggu lagi proses pengukuran berat badan selama 5 detik.
 - 16) Buzzer akan berbunyi tiga kali.
 - 17) Selesai.
 - 18) Data pengukuran akan tertera pada LCD box kontrol (*Master Controller*).
 - 19) Data pengukuran juga langsung *update* pada *website* www.smartech515.xyz.
 - 20) Selesai.

b. *Power* Tungkai

Power Tungkai [AUTO] adalah pengukuran secara otomatis yaitu pengukuran dimulai dengan mengukur berat badan lalu mengukur tinggi badan sama seperti menu ‘*Scan* Berat & Tinggi’ baru mulai mengukur *power vertical* atau *horizontal jump* (pengukuran ini digunakan untuk jumlah peserta yang banyak agar proses pengukuran bisa berlangsung cepat). *Power* Tungkai [MANU] adalah pengukuran secara manual yaitu pengukuran *power vertical* atau *horizontal jump* tanpa mengukur berat badan dan tinggi badan terlebih dahulu (pengukuran ini digunakan untuk individu dan tidak dipergunakan untuk jumlah peserta yang banyak). Cara memilih nya dengan menekan tombol “*BACK*”. Setelah pilih mode otomatis atau manual lalu tekan tombol “*OK*” untuk masuk pengukuran.

- 1) Didalam menu pengkuran *power* tungkai ini ada pilihan pengukuran *Vertical Jump* dan *Horizontal Jump* tekan tombol “*OK*” untuk masuk.
- 2) Setelah memilih jenis pengukuran maka akan muncul pilihan jenis kelamin dan umur. Pilih jenis kelamin terlebih dahulu dengan menekan tombol “*UP*” atau “*DOWN*” lalu tekan tombol “*OK*”. Selanjutnya pilih umur dengan menekan tombol “*UP*” atau “*DOWN*” lalu tekan tombol “*OK*” untuk mengakhiri.
- 3) Setelah memilih jenis pengukuran, jenis kelamin, dan umur selanjutnya mulai pengukuran dengan memperhatikan hal-hal berikut:

a) Pengukuran mode otomatis *vertical jump*.

- Testi naik pada timbangan dan sejajar lurus dengan posisi alat.
- Badan tegap dan kaki harus rapat.
- Tunggu sampai tiang pengukur tinggi badan turun dan menyentuh ujung kepala.
- Setelah itu tunggu lagi proses pengukuran berat badan selama 5 detik.
- Buzzer akan berbunyi tiga kali.
- Maju kedepan dan cari tempat yang menurut testi nyaman untuk melakukan *vertical jump*.
- Siap-siap untuk memulai lompatan.
- Saat led hijau pada marker menyala, langsung mulai lompat.
- Data pengukuran akan tertera pada LCD box kontrol (*Master Controller*).
- Data pengukuran juga langsung *update* pada *website* www.smartech515.xyz.
- Selesai.

b) Pengukuran mode otomatis *horizontal jump*.

- Testi naik pada timbangan dan sejajar lurus dengan posisi alat.
- Badan tegap dan kaki harus rapat.
- Tunggu sampai tiang pengukur tinggi badan turun dan menyentuh ujung kepala.

- Setelah itu tunggu lagi proses pengukuran berat badan selama 5 detik.
- Buzzer akan berbunyi tiga kali.
- Maju pada tumpuan depan.
- Siap-siap untuk memulai lompatan, setelah siap langsung lompat secara *horizontal* ke arah depan.
- Tunggu sampai mekanik *slider* maju kedepan dan mengukur.
- Data pengukuran akan tertera pada LCD box kontrol (*Master Controller*).
- Data pengukuran juga langsung *update* pada *website* www.smartech515.xyz.
- Selesai.

c) Pengukuran mode manual *vertical jump*.

- Cari tempat yang menurut testi nyaman untuk melakukan *vertical jump*.
- Testor menekan tombol “OK” untuk memulai pengukuran.
- Siap-siap untuk memulai lompatan.
- Saat led hijau pada marker menyala, langsung mulai lompat.
- Data pengukuran akan tertera pada LCD box kontrol (*Master Controller*).
- Data pengukuran juga langsung *update* pada *website* www.smartech515.xyz.
- Selesai.

d) Pengukuran mode manual *horizontal jump*.

- Testi berdiri pada tumpuan depan alat.
- Testor menekan tombol “OK” untuk memulai pengukuran.
- Siap-siap untuk memulai lompatan, setelah siap langsung lompat secara *horizontal* ke arah depan.
- Tunggu sampai mekanik *slider* maju kedepan dan mengukur.
- Data pengukuran akan tertera pada LCD box kontrol (*Master Controller*).
- Data pengukuran juga langsung *update* pada *website* www.smartech515.xyz.
- Selesai.

c. *Whole Body Reaction*

- 1) Setelah masuk menu *Whole Body Reaction* maka akan muncul pilihan jenis kelamin dan umur. Pilih jenis kelamin terlebih dahulu dengan menekan tombol “UP” atau “DOWN” lalu tekan tombol “OK”. Selanjutnya pilih mode pengukuran ada dua yaitu *Audio* dan *Visual* dengan menekan tombol “UP” atau “DOWN” lalu tekan tombol “OK” untuk mengakhiri.
- 2) Hubungkan tumpuan pengukur *Whole Body Reaction* dengan box kontrol (*Master Controller*) menggunakan USB.
- 3) Hadapkan tumpuan dengan *reflector* cahaya dengan jarak 3 meter.
- 4) Testi naik pada tumpuan pengukur *Whole Body Reaction*.
- 5) Jika yang dipilih adalah mode *Audio*, testi harus fokus dengan pendengarannya.

- 6) Jika yang dipilih adalah mode *Visual*, testi harus fokus dengan *reflector* cahaya.
- 7) Lalu testor menekan tombol “OK” pada menu.
- 8) Tunggu sampai *Buzzer* atau *reflector* cahaya menyala sesuai dengan jenis pengukuran yang diilih. Ketika menyala, segera keluarkan kaki dari tumpuan secepatnya.
- 9) Data pengukuran akan tertera pada LCD box kontrol (*Master Controller*).
- 10) Data pengukuran juga langsung *update* pada *website* www.smartech515.xyz.
- 11) Selesai.

d. *Power Endurance*

- 1) Setelah masuk menu *Power Endurance* akan muncul dua pilihan yaitu mulai pengukuran dan kalibrasi.
- 2) Untuk melakukan kalibrasi maka tekan tombol “SAVE”.
- 3) Untuk melakukan pengukuran maka tekan tombol “OK”.
- 4) Pastikan marker (*Slave 1 Controller*) sudah terpasang pada kaki testi.
- 5) Testi siap.
- 6) Testor menekan tombol “OK” untuk memulai pengukuran.
- 7) Saat led hijau menyala langsung melakukan lompatan secara *vertical* keatas sebanyak 10 kali dengan interval antara lompatan satu dengan lompatan berikutnya 2 detik.

- 8) Setelah melakukan lompatan, *buzzer* akan menyala sebanyak jumlah lompatan dan diakhiri dengan nada panjang pada *buzzer*.
- 9) Data pengukuran akan tertera pada LCD box kontrol (*Master Controller*).
- 10) Data pengukuran juga langsung *update* pada *website* www.smartech515.xyz.
- 11) Selesai.

20. Hal-hal yang perlu diperhatikan pada menu *slide* dua yaitu:

a. Turunkan Tiang

Digunakan untuk menurunkan tiang pengukur tinggi badan jika proses pengukuran sudah selesai dan alat akan dimatikan dengan cara menekan tombol “OK”.

b. Lihat Data Kelompok

Digunakan untuk melihat data pengukuran yang sudah dilakukan secara keseluruhan dengan cara:

- 1) Hubungkan box kontrol (*Master Controller*) dengan laptop/PC menggunakan kabel USB.
- 2) Buka *software* Arduino IDE.
- 3) Pastikan *port* sesuai dengan port yang aktif dan jenis *board* Arduino sesuai dengan Arduino Mega 2560.
- 4) lalu pilih tekan menu Serial Monitor pada pojok kanan atas.
- 5) Tekan tombol “OK” pada pilihan menu Lihat Data Kelompok.
- 6) Setelah masuk menu ada pilihan kelompok 1 sampai 5 dengan cara menekan tombol “UP” atau “DOWN”.

- 7) Lalu tekan tombol “OK”.
- 8) Lihat layer Serial Monitor akan muncul data kelompok.
- 9) Selesai.

c. Informasi *Micro SD*

Tekan tombol “OK” untuk masuk menu informasi *micro SD*, lalu akan tampil kapasitas *micro SD* dan keterangan. Jika memori sudah mencapai batas minimal maka akan muncul keterangan ‘Memori Penuh’ dan jika memori masih longgar maka akan muncul keterangan ‘Memori Masih Cukup’.

d. Pengaturan *Micro SD*

Berisi pengaturan untuk menghapus data kelompok 1 sampai 5 sesuai pilihan dengan cara menekan tombol “UP” atau “DOWN” lalu diakhiri dengan menekan tombol “OK”. Kemudian ada fungsi tombol untuk Format seluruh isi dari *micro SD* dengan cara menekan tombol “UP” dan “OK” dan “SAVE” secara bersamaan.

21. Perintah penyimpanan data dengan cara menekan tombol “SAVE” pada menu pilihan pengukuran. Setelah itu akan ada pilihan user 1 sampai 50 dengan cara menekan tombol “UP” atau “DOWN” lalu tekan “OK”. Setelah itu akan ada pilihan kelompok penyimpanan data 1 sampai 5 dengan cara menekan tombol “UP” atau “DOWN” lalu tekan “OK” lalu proses penyimpanan akan berjalan. Setelah itu tampilan akan kembali pada menu *stanby*.

22. Selesai.

BAB IV

PENGUJIAN DAN PEMBAHASAN

Tujuan dari pengujian dan pembahasan adalah untuk mengetahui kinerja alat yang sudah dibuat baik secara perbagian blok rangkaian maupun keseluruhan sistem, apakah sudah seperti yang diharapkan atau belum. Pengujian ini meliputi:

A. Hasil Pengujian

1. Pengujian Fungsional

Pengujian fungsional ini dilakukan untuk mengetahui fungsionalitas dari alat serta mengetahui validitas dan reabilitas pada alat. Hal ini sangat penting, karena pada setiap pembuatan alat pengukuran harus ada instrument validitas dan reabilitas supaya alat yang digunakan pada pengukuran tidak melenceng dari teori yang ada.

a. Pengujian validitas dan reabilitas Kerja Catu Daya

Pengujian validitas dan reabilitas Catu Daya dilakukan untuk memastikan tegangan yang keluar tidak melebihi tegangan yang dibutuhkan komponen. Pengujian ini dilakukan menggunakan dua metode, tanpa beban dan dengan beban dilakukan sebanyak lima kali dengan tujuan untuk memastikan hasil keluaran tetap stabil pada nilai yang dibutuhkan oleh masing-masing komponen. Kemudian kedua data dibandingkan dan dihitung selisih dan nilai *error* nya sehingga dapat diketahui tingkat validitas nya jika *error* tidak melebihi 5%

1) Pengujian Tanpa Beban

Tabel 36. Pengujian Validitas dan Reabilitas Catu Daya Tanpa Beban

No	Pengukuran Pada	Jumlah pengukuran	V out	V out terbaca	Error	Selisih	Keterangan
1.	<i>Power supply Switching</i> 12 V 5A.	1	12	12,22	1,8%	0,2V	Valid
		2	12	12,23	1,9%	0,3V	Valid
		3	12	12,22	1,8%	0,2V	Valid
		4	12	12,22	1,8%	0,2V	Valid
		5	12	12,22	1,8%	0,2V	Valid
2.	<i>Power supply Switching</i> 12 V 2A.	1	12	12,13	1,07%	0,13V	Valid
		2	12	12,13	1,07%	0,13V	Valid
		3	12	12,13	1,07%	0,13V	Valid
		4	12	12,13	1,07%	0,13V	Valid
		5	12	12,13	1,07%	0,13V	Valid
3.	Modul XL-4015 pada Arduino Mega	1	5	5,02	0,3%	0,02V	Valid
		2	5	5,02	0,3%	0,02V	Valid
		3	5	5,02	0,3%	0,02V	Valid
		4	5	5,02	0,3%	0,02V	Valid
		5	5	5,02	0,3%	0,02V	Valid

2) Pengujian Dengan Beban





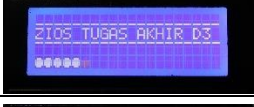


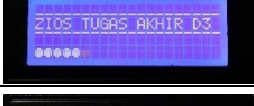
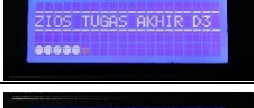
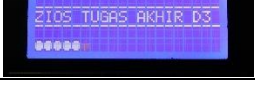
Tabel 37. Pengujian Validitas dan Reabilitas Catu Daya Dengan Beban

No	Pengukuran pada	Jumlah pengukuran	V out	V out terbaca	Error	Selisih	Keterangan
1.	<i>Power supply Switching</i> 12 V 5A.	1	12	12,11	0,9%	0,11V	Valid
		2	12	12,11	0,9%	0,11V	Valid
		3	12	12,11	0,9%	0,11V	Valid
		4	12	12,11	0,9%	0,11V	Valid
		5	12	12,11	0,9%	0,11V	Valid
2.	<i>Power supply Switching</i> 12 V 2A.	1	12	12,07	0,57%	0,07V	Valid
		2	12	12,07	0,57%	0,07V	Valid
		3	12	12,07	0,57%	0,07V	Valid
		4	12	12,08	0,66%	0,08V	Valid
		5	12	12,07	0,57%	0,07V	Valid
3.	Modul XL-4015 pada Arduino Mega	1	5	5,00	0%	0V	Valid
		2	5	5,01	0,19%	0,01V	Valid
		3	5	5,00	0%	0,01V	Valid
		4	5	5,01	0,19%	0V	Valid
		5	5	5,00	0%	0,01V	Valid

b. Pengujian validitas dan reabilitas Kerja Slot *Micro SD Adapter*

Pengujian validitas dan reabilitas Slot *Micro SD* dilakukan untuk mengetahui tingkat keberhasilan slot *micro SD* dalam membaca *micro SD card* yang sudah dimasukkan. Hal ini sangat penting karena alat ini tidak akan berjalan jika tidak ada penyimpanan *external* yaitu *micro SD card*. Berikut adalah hasil uji slot *micro SD* pada tabel 38.

Tabel 38. Pengujian Validitas dan Reabilitas Slot *Micro SD*

No	Percobaan di Tempat	Error / Sukses	Tampilan LCD	Keterangan
1	I	Sukses		Valid
2	II	Sukses		Valid
3	III	Sukses		Valid
4	IV	Sukses		Valid
5	V	Sukses		Valid
6	VI	Sukses		Valid
7	VII	Sukses		Valid
8	VIII	Sukses		Valid
9	IX	Sukses		Valid
10	X	Sukses		Valid

c. Pengujian validitas dan reabilitas Kerja *Loadcell*

Pengujian validitas dan reabilitas *Loadcell* dilakukan dengan cara membandingkan beban dengan menggunakan media gallon air mineral yang diisi dengan air keran dengan berat tertentu dan di bandingkan antara timbangan yang orisinil dan timbangan yang sudah di oprek. Kemudian kedua data tersebut dibandingkan dan dihitung selisih dan nilai *error* nya sehingga dapat diketahui tingkat validitas nya jika *error* tidak melebihi 5%. Hasil pengujian *Loadcell* dapat dilihat pada tabel 39.

Tabel 39. Pengujian Validitas dan Reabilitas *Loadcell* terhadap Berat Badan

Percobaan di Tempat	No	Sampel Berat (Kg)	Hasil Ukur Alat (Kg)	Selisih (Kg)	Error (%)	Keterangan
1	1	10,0	10,1	0,1	0,99	Valid
	2	20,0	20,0	0	0	Valid
	3	30,0	30,0	0	0	Valid
	4	40,0	40,1	0,1	0,25	Valid
	5	50,0	50,0	0	0	Valid
	6	60,0	60,0	0	0	Valid
	7	70,0	70,1	0,1	0,14	Valid
	8	80,0	80,0	0	0	Valid
	9	90,0	90,0	0	0	Valid
	10	100,0	100,0	0	0	Valid
2	1	10,0	10,0	0	0	Valid
	2	20,0	20,0	0	0	Valid
	3	30,0	30,0	0	0	Valid
	4	40,0	40,1	0,1	0,25	Valid
	5	50,0	50,0	0	0	Valid
	6	60,0	60,0	0	0	Valid
	7	70,0	70,0	0	0	Valid
	8	80,0	80,0	0	0	Valid
	9	90,0	90,0	0	0	Valid
	10	100,0	100,1	0,1	0,09	Valid
	1	10,0	10,0	0	0	Valid
	2	20,0	20,0	0	0	Valid

3	3	30,0	30,0	0	0	Valid
	4	40,0	40,0	0	0	Valid
	5	50,0	50,0	0	0	Valid
	6	60,0	60,0	0	0	Valid
	7	70,0	70,0	0	0	Valid
	8	80,0	80,0	0	0	Valid
	9	90,0	90,0	0	0	Valid
	10	100,0	100,0	0	0	Valid

d. Pengujian validitas dan reabilitas Kerja perangkat pengukur Tinggi Badan

Pengujian validitas dan reabilitas perangkat pengukur tinggi badan dilakukan dengan cara mengukur beberapa sampel tinggi badan menggunakan meteran. Kemudian kedua data tersebut dibandingkan dan dihitung selisih dan nilai *error* nya sehingga dapat diketahui tingkat validitas nya jika *error* tidak melebihi 5%.

Tabel 40. Pengujian Validitas dan Reabilitas perangkat pengukur Tinggi Badan

Percobaan di Tempat	No	Sampel Tinggi (Cm)	Hasil Ukur Alat (Cm)	Selisih (Cm)	<i>Error</i> (%)	Keterangan
1	1	10,0	10,1	0,1	0,99	Valid
	2	30,0	30,0	0	0	Valid
	3	50,0	50,0	0	0	Valid
	4	70,0	70,1	0,1	0,14	Valid
	5	90,0	90,0	0	0	Valid
	6	110,0	110,0	0	0	Valid
	7	130,0	130,1	0,1	0,07	Valid
	8	150,0	150,0	0	0	Valid
	9	170,0	170,0	0	0	Valid
	10	190,0	190,0	0	0	Valid
2	1	10,0	10,0	0	0	Valid
	2	30,0	30,0	0	0	Valid
	3	50,0	50,0	0	0	Valid
	4	70,0	70,1	0,1	0,14	Valid
	5	90,0	90,0	0	0	Valid

	6	110,0	110,0	0	0	Valid
	7	130,0	130,0	0	0	Valid
	8	150,0	150,0	0	0	Valid
	9	170,0	170,0	0	0	Valid
	10	190,0	190,1	0,1	0,05	Valid
3	1	10,0	10,0	0	0	Valid
	2	30,0	30,0	0	0	Valid
	3	50,0	50,0	0	0	Valid
	4	70,0	70,0	0	0	Valid
	5	90,0	90,0	0	0	Valid
	6	110,0	110,0	0	0	Valid
	7	130,0	130,0	0	0	Valid
	8	150,0	150,0	0	0	Valid
	9	170,0	170,0	0	0	Valid
	10	190,0	190,0	0	0	Valid

e. Pengujian validitas dan reabilitas Kerja perangkat pengukur Panjang Lompatan

Pengujian validitas dan reabilitas perangkat pengukur panjang lompatan dilakukan dengan cara mengukur beberapa sampel panjang lompatan seseorang menggunakan meteran konvensional dan perangkat pengukur panjang lompatan. Kemudian kedua data tersebut dibandingkan dan dihitung selisih dan nilai *error* nya sehingga dapat diketahui tingkat validitas nya jika *error* tidak melebihi 5%.

Tabel 41. Pengujian Validitas dan Reabilitas Panjang Lompatan

Percobaan di Tempat	No	Sampel Panjang (Cm)	Hasil Ukur Alat (Cm)	Selisih (Cm)	Error (%)	Keterangan
1	1	50,0	51,0	1	1,96	Valid
	2	75,0	75,0	0	0	Valid
	3	100,0	100,0	0	0	Valid
	4	125,0	125,5	0,5	0,39	Valid
	5	150,0	150,0	0	0	Valid
	6	175,0	175,0	0	0	Valid
	7	200,0	201,0	1	0,49	Valid
	8	225,0	225,0	0	0	Valid

	9	250,0	250,0	0	0	Valid
	10	275,0	275,0	0	0	Valid
2	1	50,0	50,0	0	0	Valid
	2	75,0	75,0	0	0	Valid
	3	100,0	100,0	0	0	Valid
	4	125,0	126,0	1	0,79	Valid
	5	150,0	150,0	0	0	Valid
	6	175,0	175,0	0	0	Valid
	7	200,0	202,0	2	0,99	Valid
	8	225,0	225,0	0	0	Valid
	9	250,0	250,0	0	0	Valid
	10	275,0	278,0	3	1,07	Valid
3	1	50,0	50,0	0	0	Valid
	2	75,0	75,0	0	0	Valid
	3	100,0	100,5	0,5	0,49	Valid
	4	125,0	125,0	0	0	Valid
	5	150,0	151,5	1,5	0,99	Valid
	6	175,0	175,0	0	0	Valid
	7	200,0	200,0	0	0	Valid
	8	225,0	225,0	0	0	Valid
	9	250,0	250,5	0,5	0,19	Valid
	10	275,0	275,0	0	0	Valid

f. Pengujian validitas dan reabilitas Kerja Ultrasonik SRF04 terhadap Tinggi Lompatan

Pengujian validitas dan reabilitas Ultrasonik SRF04 terhadap Tinggi Lompatan dilakukan dengan cara mengukur beberapa sampel tinggi lompatan menggunakan meteran konvensional dan sensor ultrasonik SRF04. Kemudian kedua data tersebut dibandingkan dan dihitung selisih dan nilai *error* nya sehingga dapat diketahui tingkat validitas nya jika *error* tidak melebihi 5%.

Tabel 42. Pengujian Validitas dan Reabilitas Ultrasonik SRF04 terhadap Tinggi Lompatan

Percobaan di Tempat	No	Sampel Tinggi	Hasil Ukur	Selisih (Cm)	Error (%)	Keterangan
---------------------	----	---------------	------------	--------------	-----------	------------

		Lompat (Cm)	Alat (Cm)			
1	1	15,0	15,5	0,5	3,22	Valid
	2	20,0	20,0	0	0	Valid
	3	25,0	25,2	0,2	0,79	Valid
	4	30,0	30,0	0	0	Valid
	5	35,0	35,1	0,1	0,28	Valid
	6	40,0	40,7	0,7	1,71	Valid
	7	45,0	45,0	0	0	Valid
	8	50,0	50,0	0	0	Valid
	9	55,0	55,6	0,6	1,07	Valid
	10	60,0	60,0	0	0	Valid
2	1	15,0	15,2	0,2	1,31	Valid
	2	20,0	20,2	0,2	0,99	Valid
	3	25,0	25,3	0,3	1,18	Valid
	4	30,0	30,1	0,1	0,33	Valid
	5	35,0	35,7	0,7	1,96	Valid
	6	40,0	40,0	0	0	Valid
	7	45,0	45,8	0,8	1,74	Valid
	8	50,0	50,5	0,5	0,99	Valid
	9	55,0	55,3	0,3	0,54	Valid
	10	60,0	60,0	0	0	Valid
3	1	15,0	15,0	0	0	Valid
	2	20,0	20,9	0,9	4,30	Valid
	3	25,0	25,9	0,9	3,47	Valid
	4	30,0	30,3	0,3	0,99	Valid
	5	35,0	35,6	0,6	1,68	Valid
	6	40,0	40,7	0,7	1,71	Valid
	7	45,0	45,4	0,4	0,88	Valid
	8	50,0	50,3	0,3	0,59	Valid
	9	55,0	55,8	0,8	1,43	Valid
	10	60,0	60,0	0	0	Valid

g. Pengujian validitas dan reabilitas Kerja perangkat pengukur *Whole Body*

Reaction

Pengujian validitas dan reabilitas perangkat pengukur *Whole Body Reaction* dilakukan dengan cara mengukur kecepatan reaksi seseorang secara acak menggunakan *stopwatch* dan perangkat pengukur *Whole Body Reaction*. Kemudian kedua data tersebut dibandingkan dan

dihitung selisih dan nilai *error* nya sehingga dapat diketahui tingkat validitas nya jika *error* tidak melebihi 5%.

Tabel 43. Pengujian Validitas dan Reabilitas perangkat pengukur

Whole Body Reaction

Percobaan di Tempat	No	Sampel Waktu Timer (Detik)	Hasil Ukur Alat (Detik)	Selisih (Detik)	Error (%)	Keterangan
1	1	0,100	0,100	0	0	Valid
	2	0,200	0,200	0	0	Valid
	3	0,300	0,300	0	0	Valid
	4	0,400	0,400	0	0	Valid
	5	0,500	0,500	0	0	Valid
	6	0,600	0,600	0	0	Valid
	7	0,700	0,700	0	0	Valid
	8	0,800	0,800	0	0	Valid
	9	0,900	0,900	0	0	Valid
	10	1,00	1,00	0	0	Valid
2	1	0,100	0,100	0	0	Valid
	2	0,200	0,200	0	0	Valid
	3	0,300	0,300	0	0	Valid
	4	0,400	0,400	0	0	Valid
	5	0,500	0,500	0	0	Valid
	6	0,600	0,600	0	0	Valid
	7	0,700	0,700	0	0	Valid
	8	0,800	0,800	0	0	Valid
	9	0,900	0,900	0	0	Valid
	10	1,00	1,00	0	0	Valid
3	1	0,100	0,100	0	0	Valid
	2	0,200	0,200	0	0	Valid
	3	0,300	0,300	0	0	Valid
	4	0,400	0,400	0	0	Valid
	5	0,500	0,500	0	0	Valid
	6	0,600	0,600	0	0	Valid
	7	0,700	0,700	0	0	Valid
	8	0,800	0,800	0	0	Valid
	9	0,900	0,900	0	0	Valid
	10	1,00	1,00	0	0	Valid

h. Pengujian validitas dan reabilitas Kerja sistem penyimpanan *micro SD*

Pengujian validitas dan reabilitas sistem penyimpanan *micro SD* dilakukan dengan cara membandingkan data hasil pengukuran dengan data yang disimpan pada EEPROM dan data yang disimpan pada *micro SD*. Ketika ketiga data tersebut dibandingkan lalu diamati apakah ada perbedaan nilai atau tidak, jika tidak berarti sistem ini berjalan dengan baik. System dapat dinyatakan valid jika data dari ketiga media penyimpanan menunjukkan hasil yang sama.

Tabel 44. Pengujian Validitas dan Reabilitas sistem penyimpanan
micro SD

No	Jenis pengukuran yang dilakukan	Hasil Ukur	Data di EEprom	Data di <i>Micro SD</i>	Valid/Tidak Valid
1	Tinggi Badan	167,79 Cm	167,79	167,79	Valid
2	Berat Badan	61,24 Cm	61,24	61,24	Valid
3	Tinggi Lompatan	57 Kg	57	57	Valid
4	Waktu Lompatan	0,71 Detik	0,71	0,71	Valid
5	Panjang Lompatan	226 Cm	226	226	Valid
6	Waktu Lompatan	2,55 Detik	2,55	2,55	Valid
7	<i>Whole Body Reaction</i>	1.05 Detik	1.05	1.05	Valid
8	Rata-rata Lompatan	62 Cm	62	62	Valid
9	Maksimal Lompatan	79 Cm	79	79	Valid
10	Minimal Lompatan	9 Cm	9	9	Valid

- i. Pengujian validitas dan reabilitas Kerja Komunikasi dan *Pairing Bluetooth*

Pengujian validitas dan reabilitas komunikasi *Bluetooth* dilakukan untuk mengetahui apakah data yang dikirimkan dari *Master Controller* dapat diterima dengan baik oleh *slave controller* atau tidak. Kemudian pengujian *pairing Bluetooth* dilakukan untuk mengetahui seberapa baik koneksi antara *Bluetooth master* dengan *Bluetooth slave*. Komunikasi data dapat dikatakan valid jika data yang ditransmisikan dari master kontroler dapat diterima dengan baik oleh slave kontroler. *Pairing* data dapat dikatakan valid jika proses *pairing* antara modul *Bluetooth master* dengan modul *Bluetooth slave* dapat tersambung.

Tabel 45. Pengujian Validitas dan Reabilitas Komunikasi Bluetooth

No	Data dikirim dari <i>Master</i> kontroler	Data yang diterima <i>Slave 1</i> kontroler	Data yang diterima <i>Slave 2</i> kontroler	<i>Error</i> (%)	Valid/Tidak Valid
1	10	10	10	0	Valid
2	A	A	A	0	Valid
3	a	a	a	0	Valid
4	-60	-60	-60	0	Valid
5	30,5	30,5	30,5	0	Valid

Tabel 46. Pengujian Validitas dan Reabilitas *Pairing Bluetooth*

No	Koneksi Antara	Percobaan ke-	Tersambung / Tidak Tersambung	Kondisi	Keterangan
1	<i>Bluetooth Slave 1</i> dengan <i>Master Controller</i>	I	Tersambung	Led <i>Bluetooth</i> 2 kali kedip	Valid
		II	Tersambung	Led <i>Bluetooth</i> 2 kali kedip	Valid
		III	Tersambung	Led <i>Bluetooth</i>	Valid

				2 kali kedip	
		IV	Tidak Tersambung	Led kedip-kedip terus	Tidak Valid
		V	Tersambung	Led <i>Bluetooth</i> 2 kali kedip	Valid
2	<i>Bluetooth Slave 2 dengan Master Controller</i>	I	Tersambung	Led <i>Bluetooth</i> 2 kali kedip	Valid
		II	Tersambung	Led <i>Bluetooth</i> 2 kali kedip	Valid
		III	Tersambung	Led <i>Bluetooth</i> 2 kali kedip	Valid
		IV	Tersambung	Led <i>Bluetooth</i> 2 kali kedip	Valid
		V	Tersambung	Led <i>Bluetooth</i> 2 kali kedip	Valid

j. Pengujian validitas dan reabilitas Kerja Sistem Telemetri

Pengujian validitas dan reabilitas sistem telemetri ini dilakukan untuk mengetahui seberapa baik proses pengiriman data yang dilakukan oleh perangkat ke *database* secara *online*. Sistem telemetri dapat dikatakan valid jika data yang di terima *database website* sesuai dengan data pengukuran yang dilakukan.

Tabel 47. Pengujian Validitas dan Reabilitas sistem *Telemetry*

No	Jenis pengukuran yang dilakukan	Hasil Ukur Alat	Data di EEprom	Data di Website	Keterangan
1	Tinggi Badan	167,79 Cm	167,79	167,79	Valid
2	Berat Badan	61,24 Cm	61,24	61,24	Valid
3	Tinggi Lompatan	57 Kg	57	57	Valid
4	Waktu Lompatan	0,71 Detik	0,71	0,71	Valid
5	Panjang Lompatan	226 Cm	226	226	Valid
6	Waktu Lompatan	2,55 Detik	2,55	2,55	Valid
7	<i>Whole Body Reaction</i>	1.05 Detik	1.05	1.05	Valid
8	Rata-rata Lompatan	62 Cm	62	62	Valid
9	Maksimal Lompatan	79 Cm	79	79	Valid
10	Minimal Lompatan	9 Cm	9	9	Valid

k. Pengujian validitas dan reabilitas Kerja *Battery*

Pengujian validitas dan reabilitas *battery* dilakukan pada *output battery*. Pengukuran *battery* dilakukan sebanyak 10 kali untuk memastikan kondisi *battery* dalam keadaan baik.

1) Pengujian Tanpa Beban

Tabel 48. Pengujian Validitas dan Reabilitas *battery* lipo 11,1 volt

2,2A Tanpa Beban

No	Waktu (menit)	V Out (V)
1	10	12,6
2	20	12,6
3	30	12,6
4	40	12,6

5	50	12,5
6	60	12,5
7	70	12,5
8	80	12,5
9	90	12,4
10	100	12,4

Tabel 49. Pengujian Validitas dan Reabilitas *battery* lipo 7,8 volt

850 mAh Tanpa Beban

No	Waktu (menit)	V Out (V)
1	10	8,4
2	20	8,4
3	30	8,4
4	40	8,4
5	50	8,4
6	60	8,4
7	70	8,3
8	80	8,3
9	90	8,3
10	100	8,3

2) Pengujian Dengan Beban

Tabel 50. Pengujian Validitas dan Reabilitas *battery* lipo 11,1 volt

2,2A Dengan Beban

No	Waktu (menit)	V Out (V)
1	10	12,5
2	20	12,4
3	30	12,3
4	40	12,2
5	50	12,1
6	60	12
7	70	11,9
8	80	11,8
9	90	11,7

10	100	11,6
----	-----	------

Tabel 51. Pengujian Validitas dan Reabilitas *battery* lipo 7,8 volt
850 mAh Dengan Beban

No	Waktu (menit)	V Out (V)
1	10	8,4
2	20	8,4
3	30	8,3
4	40	8,3
5	50	8,2
6	60	8,2
7	70	8,1
8	80	8,1
9	90	8
10	100	8

2. Pengujian Unjuk Kerja

Pengujian unjuk kerja dilakukan dengan cara menguji sistem alat secara keseluruhan. Pengujian ini bertujuan untuk mengetahui kekurangan dan kelebihan dari sebuah sistem terintegrasi yang telah dibuat.

a. Unjuk Kerja Catu Daya

1) Catu Daya Tanpa Beban

Tabel 52. Pengujian Catu Daya Tanpa Beban

No	Pengukuran pada	Jumlah pengukuran	V out	V out terbaca	Error	Ket
1.	<i>Power supply Switching</i> 12 V 5A.	1	12	12,22	1,8%	0,2V
		2	12	12,23	1,9%	0,3V
		3	12	12,22	1,8%	0,2V
		4	12	12,22	1,8%	0,2V
		5	12	12,22	1,8%	0,2V
2.	<i>Power supply Switching</i> 12 V 2A.	1	12	12,13	1,07%	0,13V
		2	12	12,13	1,07%	0,13V
		3	12	12,13	1,07%	0,13V
		4	12	12,13	1,07%	0,13V





		5	12	12,13	1,07%	0,13V
3.	Modul XL-4015 pada Arduino Mega	1	5	5,02	0,3%	0,02V
		2	5	5,02	0,3%	0,02V
		3	5	5,02	0,3%	0,02V
		4	5	5,02	0,3%	0,02V
		5	5	5,02	0,3%	0,02V






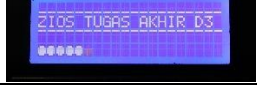
2) Catu Daya Dengan Beban

Tabel 53. Pengujian Catu Daya Dengan Beban

No	Pengukuran pada	Jumlah pengukuran	V out	V out terbaca	Error	Ket
1.	<i>Power supply Switching 12 V 5A.</i>	1	12	12,11	0,9%	0,11V
		2	12	12,11	0,9%	0,11V
		3	12	12,11	0,9%	0,11V
		4	12	12,11	0,9%	0,11V
		5	12	12,11	0,9%	0,11V
2.	<i>Power supply Switching 12 V 2A.</i>	1	12	12,07	0,57%	0,07V
		2	12	12,07	0,57%	0,07V
		3	12	12,07	0,57%	0,07V
		4	12	12,08	0,66%	0,08V
		5	12	12,07	0,57%	0,07V
3.	Modul XL-4015 pada Arduino Mega	1	5	5,00	0%	0V
		2	5	5,01	0,19%	0,01V
		3	5	5,00	0%	0,01V
		4	5	5,01	0,19%	0V
		5	5	5,00	0%	0,01V

b. Unjuk Kerja Slot *Micro SD Adapter***Tabel 54.** Unjuk Kerja Slot *Micro SD*

No	Percobaan	Error / Sukses	Tampilan LCD
1	I	Sukses	
2	II	Sukses	
3	III	Sukses	
4	IV	Sukses	

5	V	Sukses	
6	VI	Sukses	
7	VII	Sukses	
8	VIII	Sukses	
9	IX	Sukses	
10	X	Sukses	

c. Unjuk Kerja *Loadcell*

Tabel 55. Unjuk Kerja *Loadcell* terhadap Berat Badan

No	Hasil Timbangan Orisinil	Hasil Ukur Alat	Selisih	<i>Error (%)</i>
1	14,5	14,3	0,2	1,3
2	50,4	50,7	0,3	0,59
3	75,02	75,04	0,2	0,26
4	64,6	64,8	0,2	0,3
5	20,3	20,4	0,1	0,49
6	34,1	34,2	0,1	0,29
7	47,7	47,7	0	0
8	81,9	81,8	0,1	0,12
9	10,6	10,6	0	0
10	19,5	19,5	0	0

d. Unjuk Kerja perangkat pengukur Tinggi Badan

Tabel 56. Unjuk Kerja perangkat pengukur Tinggi Badan

No	Tinggi Sampel Acak (Cm)	Hasil Ukur Alat (Cm)	Selisih (Cm)	<i>Error (%)</i>
1	150,4	150,7	0,3	0,19
2	155,1	155,3	0,2	0,13

3	171,5	171,7	0,2	0,12
4	137,9	140,1	0,2	0,14
5	167,6	167,8	0,2	0,12
6	175,9	176	0,1	0,05
7	164,8	164,9	0,1	0,06
8	157,3	157,5	0,2	0,12
9	149,2	149,4	0,2	0,13
10	195,5	195,7	0,2	0,10

e. Unjuk Kerja perangkat pengukur Panjang Lompatan

Tabel 57. Unjuk Kerja Panjang Lompatan

No	Hasil Ukur Meteran Acak (Cm)	Hasil Ukur Alat (Cm)	Selisih (Cm)	Error (%)
1	70	72,2	2,2	3,04
2	100	101,4	1,4	1,38
3	130	131,1	1,1	0,83
4	160	159,1	0,9	0,56
5	190	189,8	0,2	0,1
6	210	209,8	0,2	0,09
7	230	229,7	0,3	0,13
8	90	90,6	0,6	0,66
9	45	45,7	0,7	1,53
10	60	60,3	0,3	0,49

f. Unjuk Kerja Ultrasonik SRF04 terhadap Tinggi Lompatan

Tabel 58. Unjuk Kerja Ultrasonik SRF04 terhadap Tinggi Lompatan

No	Hasil Ukur Meteran (Cm)	Hasil Ukur Alat (Cm)	Selisih (Cm)	Error (%)
1	15	16,9	1,9	11,24
2	20	21,4	1,4	6,54
3	25	26,1	1,1	4,21
4	30	30,9	0,9	2,91
5	35	35,7	0,7	1,96
6	40	40,3	0,3	0,74
7	45	45,1	0,1	0,22
8	50	49,9	0,1	0,2
9	55	54,7	0,3	0,54
10	60	59,5	0,5	0,84

g. Unjuk Kerja perangkat pengukur *Whole Body Reaction*

Tabel 59. Unjuk Kerja perangkat pengukur *Whole Body Reaction*

No	Hasil Ukur Stopwatch (detik)	Hasil Ukur Alat (detik)	Selisih (detik)	Error (%)
1	0,739	0,738	0,001	0,135
2	0,431	0,433	0,002	0,461
3	0,567	0,565	0,001	0,176
4	1,479	1,478	0,001	0,067
5	2,951	2,951	0	0
6	3,379	3,379	0	0
7	4,195	4,196	0,001	0,023
8	5,469	5,467	0,002	0,036
9	6,240	6,240	0	0
10	7,837	7,835	0,002	0,255

h. Unjuk Kerja sistem penyimpanan *micro SD***Tabel 60.** Unjuk Kerja sistem penyimpanan *micro SD*

No	Jenis pengukuran yang dilakukan	Hasil Ukur	Data di EEprom	Data di <i>Micro SD</i>	Valid/Tidak Valid
1	Tinggi Badan	167,79 Cm	167,79	167,79	Valid
2	Berat Badan	61,24 Cm	61,24	61,24	Valid
3	Tinggi Lompatan	57 Kg	57	57	Valid
4	Waktu Lompatan	0,71 Detik	0,71	0,71	Valid
5	Panjang Lompatan	226 Cm	226	226	Valid
6	Waktu Lompatan	2,55 Detik	2,55	2,55	Valid
7	<i>Whole Body Reaction</i>	1.05 Detik	1.05	1.05	Valid
8	Rata-rata Lompatan	62 Cm	62	62	Valid
9	Maksimal Lompatan	79 Cm	79	79	Valid
10	Minimal Lompatan	9 Cm	9	9	Valid

i. Unjuk Kerja Komunikasi dan *Pairing Bluetooth*

Tabel 61. Unjuk Kerja Komunikasi Bluetooth

No	Data dikirim dari <i>Master</i> kontroler	Data yang diterima <i>Slave 1</i> kontroler	Data yang diterima <i>Slave 2</i> kontroler	<i>Error (%)</i>	Valid/Tidak Valid
1	10	10	10	0	Valid
2	A	A	A	0	Valid
3	A	a	a	0	Valid
4	-60	-60	-60	0	Valid
5	30,5	30,5	30,5	0	Valid

Tabel 62. Unjuk Kerja *Pairing Bluetooth*

No	Koneksi Antara	Percobaan ke-	Tersambung / Tidak Tersambung	Keterangan
1	<i>Bluetooth Slave 1</i> dengan <i>Master Controller</i>	I	Tersambung	Led <i>Bluetooth</i> 2 kali kedip
		II	Tersambung	Led <i>Bluetooth</i> 2 kali kedip
		III	Tersambung	Led <i>Bluetooth</i> 2 kali kedip
		IV	Tidak Tersambung	Led kedip-kedip terus
		V	Tersambung	Led <i>Bluetooth</i> 2 kali kedip
2	<i>Bluetooth Slave 2</i> dengan <i>Master Controller</i>	I	Tersambung	Led <i>Bluetooth</i> 2 kali kedip
		II	Tersambung	Led <i>Bluetooth</i> 2 kali kedip
		III	Tersambung	Led <i>Bluetooth</i> 2 kali kedip
		IV	Tersambung	Led <i>Bluetooth</i> 2 kali kedip
		V	Tersambung	Led <i>Bluetooth</i> 2 kali kedip

j. Unjuk Kerja Sistem Telemetry

Tabel 63. Unjuk Kerja sistem *Telemetry*

No	Jenis pengukuran yang dilakukan	Hasil Ukur Alat	Data di EEprom	Data di <i>Website</i>	Keterangan
----	---------------------------------	-----------------	----------------	------------------------	------------

1	Tinggi Badan	167,79 Cm	167,79	167,79	Valid
2	Berat Badan	61,24 Cm	61,24	61,24	Valid
3	Tinggi Lompatan	57 Kg	57	57	Valid
4	Waktu Lompatan	0,71 Detik	0,71	0,71	Valid
5	Panjang Lompatan	226 Cm	226	226	Valid
6	Waktu Lompatan	2,55 Detik	2,55	2,55	Valid
7	<i>Whole Body Reaction</i>	1.05 Detik	1.05	1.05	Valid
8	Rata-rata Lompatan	62 Cm	62	62	Valid
9	Maksimal Lompatan	79 Cm	79	79	Valid
10	Minimal Lompatan	9 Cm	9	9	Valid

k. Unjuk Kerja *Battery*

1) *Battery* Tanpa Beban

Tabel 64. Unjuk Kerja *battery* lipo 11,1 volt 2,2A Tanpa Beban

No	Waktu (menit)	V Out (V)
1	10	12,6
2	20	12,6
3	30	12,6
4	40	12,6
5	50	12,5
6	60	12,5
7	70	12,5
8	80	12,5
9	90	12,4
10	100	12,4

Tabel 65. Unjuk Kerja *battery* lipo 7,8 volt 850 mAh Tanpa Beban

No	Waktu (menit)	V Out (V)
1	10	8,4
2	20	8,4

3	30	8,4
4	40	8,4
5	50	8,4
6	60	8,4
7	70	8,3
8	80	8,3
9	90	8,3
10	100	8,3

2) *Battery* Dengan Beban**Tabel 66.** Unjuk Kerja *battery* lipo 11,1 volt 2,2A Dengan Beban

No	Waktu (menit)	V Out (V)
1	10	12,5
2	20	12,4
3	30	12,3
4	40	12,2
5	50	12,1
6	60	12
7	70	11,9
8	80	11,8
9	90	11,7
10	100	11,6

Tabel 67. Unjuk Kerja *battery* lipo 7,8 volt 850 mAh Dengan Beban

No	Waktu (menit)	V Out (V)
1	10	8,4
2	20	8,4
3	30	8,3
4	40	8,3
5	50	8,2
6	60	8,2
7	70	8,1
8	80	8,1
9	90	8
10	100	8

Tabel 68. Tabel Uji Unjuk Kerja Pengukur *Power* Tungkai

No	Power Tungkai													
	T.B	B.B	Vertikal Jump						Horizontal Jump					
			T.L	WKT	KEC	POW	Kategori		P.L	WKT	KEC	POW	Kategori	
							Laki-laki	Perempuan					Laki-laki	Perempuan
1.	167,8	61,2	57	0,71	0,802	49,13	BAIK		226	2,69	0,840	51,41	CUKUP	
2.	155,7	61,4	46	0,54	0,851	52,30	CUKUP		179	2,15	0,832	51,11	KURANG SEKALI	
3.	157,7	44,3	57	0,64	0,890	39,45	BAIK		175	2,21	0,791	35,07	KURANG SEKALI	
4.	156,7	50,7	45	0,64	0,703	35,64	CUKUP		154	2,14	0,719	36,48	GAGAL	
5.	170,9	58,4	75	1,08	0,694	40,55	BAIK SEKALI		217	2,72	0,797	46,59	CUKUP	
6.	167,8	57,2	60	0,68	0,882	50,47	BAIK		209	2,82	0,741	42,39	KURANG	
7.	171,0	52,1	24	0,81	0,296	15,43	KURANG		204	2,55	0,8	41,68	KURANG	
8.	160,1	58,8	44	0,57	0,771	45,38	CUKUP		152	2,01	0,756	44,46	GAGAL	
9.	172,5	65,3	48	0,78	0,615	40,18	BAIK		171	2,21	0,773	50,52	KURANG SEKALI	
10.	170,8	72,4	55	0,81	0,679	49,16	BAIK		224	2,99	0,749	54,23	CUKUP	

Keterangan:

1. T.B : Tinggi Badan (Cm)
2. B.B : Berat Badan (Kg)
3. T.L : Tinggi Lompatan (Cm)
4. WKT : Waktu Lompatan (Detik)
5. KEC : Kecepatan (m/s)
6. POW : *Power* (Kgm/s)
7. P.L : Panjang Lompatan (Cm)



Gambar 114. Tampilan Hasil Pengukuran Tinggi Badan dan Berat Badan



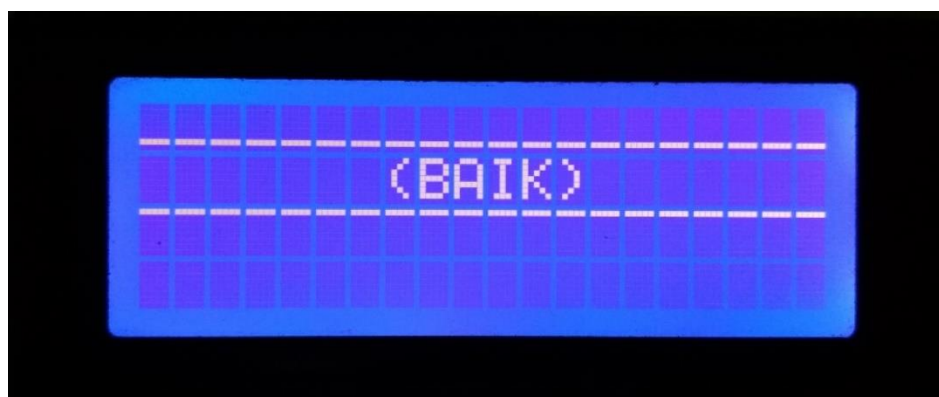
Gambar 115. Tampilan Hasil Pengukuran Power Tungkai Vertikal



Gambar 116. Tampilan Hasil Kategori Pengukuran Power Tungkai Vertikal



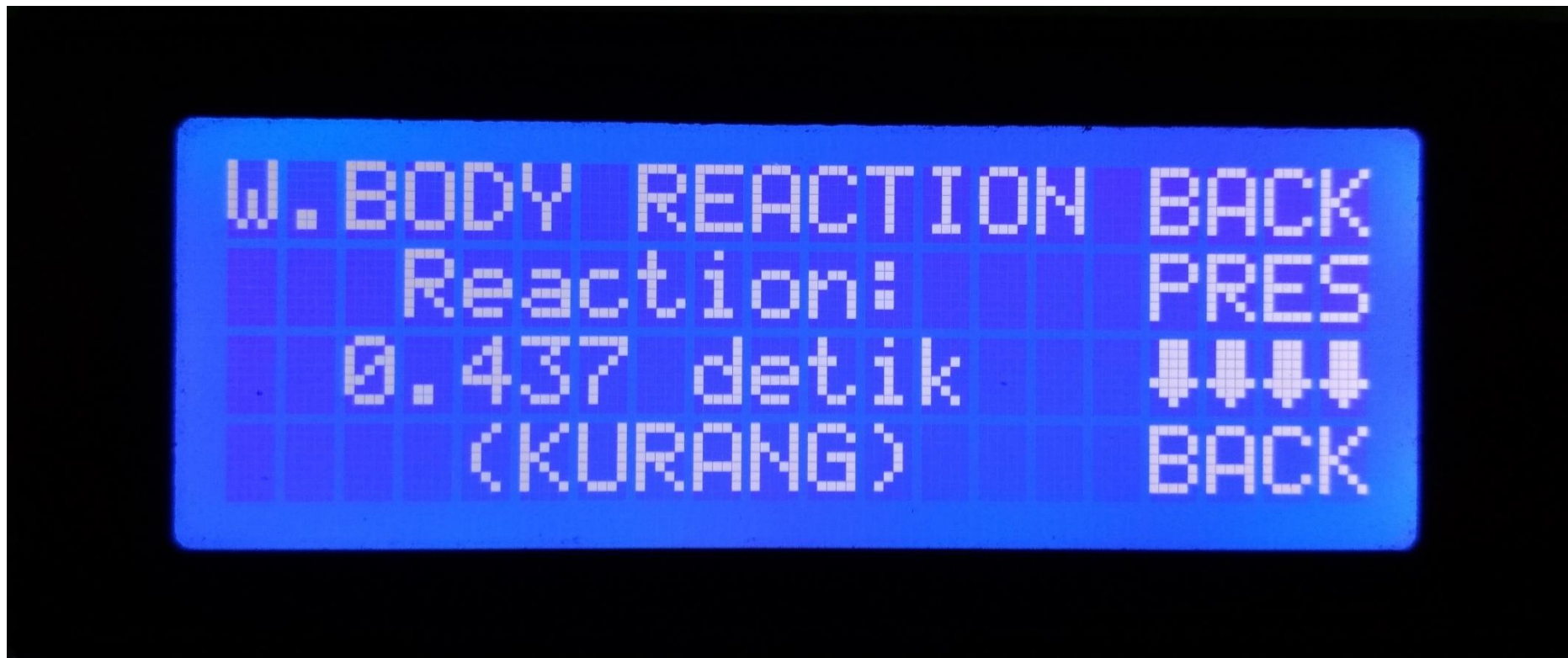
Gambar 117. Tampilan Hasil Pengukuran Power Tungkai Horizontal



Gambar 118. Tampilan Hasil Kategori Pengukuran Power Tungkai Vertikal

Tabel 69. Tabel Uji Unjuk Kerja Pengukur Whole Body Reaction

No	<i>Whole Body Reaction</i>							
	Laki-laki				Perempuan			
	<i>Audio</i>		<i>Visual</i>		<i>Audio</i>		<i>Visual</i>	
	<i>Reaction</i>	<i>Kategori</i>	<i>Reaction</i>	<i>Kategori</i>	<i>Reaction</i>	<i>Kategori</i>	<i>Reaction</i>	<i>Kategori</i>
1.	0,375	SEDANG	0,313	SEDANG	0,375	SEDANG	0,411	KURANG
2.	0,391	SEDANG	0,343	SEDANG	0,391	SEDANG	0,451	KURANG
3.	0,489	KURANG	0,473	KURANG	0,489	KURANG	0,493	KURANG
4.	0,501	KURANG	0,513	KURANG SEKALI	0,501	KURANG	0,579	KURANG SEKALI
5.	0,531	KURANG SEKALI	0,527	KURANG SEKALI	0,531	KURANG SEKALI	0,591	KURANG SEKALI
6.	0,580	KURANG SEKALI	0,555	KURANG SEKALI	0,580	KURANG SEKALI	0,547	KURANG SEKALI
7.	0,583	KURANG SEKALI	0,571	KURANG SEKALI	0,583	KURANG SEKALI	0,539	KURANG SEKALI
8.	0,639	KURANG SEKALI	0,641	KURANG SEKALI	0,639	KURANG SEKALI	0,677	KURANG SEKALI
9.	0,652	KURANG SEKALI	0,659	KURANG SEKALI	0,652	KURANG SEKALI	0,690	KURANG SEKALI
10.	0,677	KURANG SEKALI	0,694	KURANG SEKALI	0,677	KURANG SEKALI	0,794	KURANG SEKALI



Gambar 119. Tampilan Hasil Pengukuran Whole Body Reaction

Tabel 70. Tabel Uji Unjuk Kerja Pengukur *Power Endurance*

No	Berat Badan	<i>Power Setiap Lompatan</i>										<i>Power Endurance</i>		
		P.L 1	P.L 2	P.L 3	P.L 4	P.L 5	P.L 6	P.L 7	P.L 8	P.L 9	P.L 10	Max P.E	Min P.E	Rata-rata P.E
1.	75,9	70,5	67,2	66,1	64,0	60,7	67,2	64,0	70,5	70,5	62,9	70,5	60,7	66,1
2.	61,4	51,8	57,0	54,4	51,8	54,4	55,3	54,4	60,5	52,6	52,6	60,5	51,8	54,4
3.	44,3	39,2	41,1	41,1	41,1	36,1	41,1	39,2	39,2	41,1	36,1	41,1	36,1	39,2
4.	50,7	44,9	43,9	44,9	50,0	50,0	52,9	59,4	52,9	50,0	52,9	59,4	41,3	48,5
5.	58,4	54,2	57,6	57,6	60,9	55,9	60,9	55,9	51,7	49,2	43,4	60,9	43,4	54,2
6.	57,2	53,1	56,4	53,1	53,1	59,7	59,7	51,5	53,1	50,7	46,6	59,7	46,6	53,1
7.	52,1	51,4	48,4	49,9	48,4	46,9	48,4	51,4	46,1	48,4	44,7	51,4	44,7	48,4
8.	58,8	54,6	52,1	58,0	58,0	54,6	59,6	61,3	61,3	58,0	52,9	61,3	52,1	56,3
9.	65,3	64,4	64,4	64,4	68,1	68,1	64,4	66,2	64,4	58,8	56,0	68,1	56,0	63,4
10.	72,4	71,4	75,5	75,5	77,6	91,0	91,0	75,5	65,2	62,1	53,8	91,0	53,8	73,4

Keterangan:

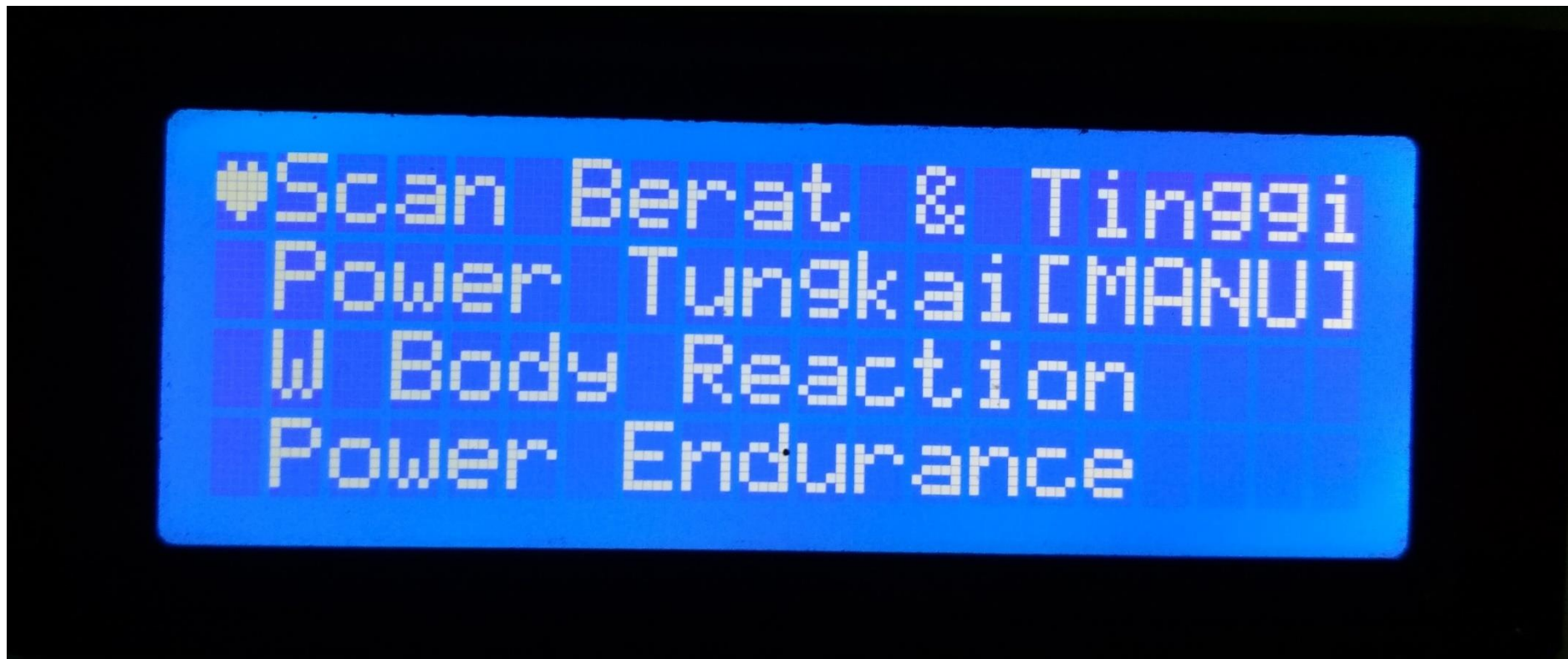
- | | | | |
|----------|----------------------------|-------------------|--|
| 1. P.L 1 | : Power Lompatan 1 (Kgm/s) | 8. P.L 8 | : Power Lompatan 8 (Kgm/s) |
| 2. P.L 2 | : Power Lompatan 2 (Kgm/s) | 9. P.L 9 | : Power Lompatan 9 (Kgm/s) |
| 3. P.L 3 | : Power Lompatan 3 (Kgm/s) | 10. P.L 10 | : Power Lompatan 10 (Kgm/s) |
| 4. P.L 4 | : Power Lompatan 4 (Kgm/s) | 11. Max P.E | : Maksimal <i>Power Endurance</i> (Kgm/s) |
| 5. P.L 5 | : Power Lompatan 5 (Kgm/s) | 12. Min P.E | : Minimal <i>Power Endurance</i> (Kgm/s) |
| 6. P.L 6 | : Power Lompatan 6 (Kgm/s) | 13. Rata-rata P.E | : Rata-rata <i>Power Endurance</i> (Kgm/s) |
| 7. P.L 7 | : Power Lompatan 7 (Kgm/s) | | |



Gambar 120. Tampilan Hasil Pengukuran Power Endurance

Tabel 71. Tabel Uji Unjuk Kerja Menu pada slide Pertama dari LCD

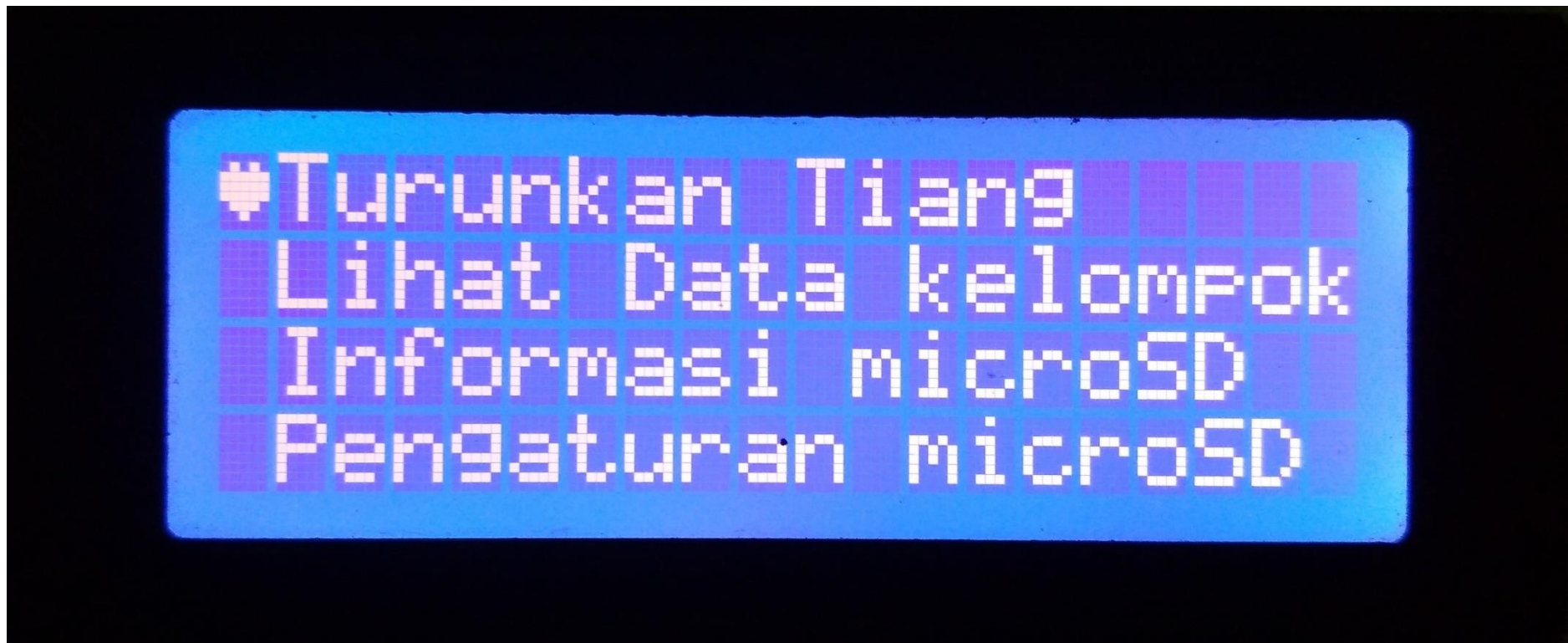
No	Scan Berat & Tinggi		Power Tungkai				W Body Reaction		Power Endurance	
			[MANU]		[AUTO]					
	Berhasil	Tidak	Berhasil	Tidak	Berhasil	Tidak	Berhasil	Tidak	Berhasil	Tidak
1.	V		V		V		V		V	
2.	V		V		V		V		V	
3.	V		V		V		V		V	
4.	V		V		V		V		V	
5.	V		V		V		V		V	
6.	V		V		V		V		V	
7.	V		V		V		V		V	
8.	V		V		V		V		V	
9.	V		V		V		V		V	
10.	V		V		V		V		V	



Gambar 121. Tampilan Menu Pada Slide pertama

Tabel 72. Tabel Uji Unjuk Kerja Menu pada *slide* kedua dari LCD

No	Turunkan Tiang		Lihat Data Kelompok		Informasi Micro SD		Pengaturan Micro SD			
							Hapus Data Kelompok		Format Semua Data	
	Berhasil	Tidak	Berhasil	Tidak	Berhasil	Tidak	Berhasil	Tidak	Berhasil	Tidak
1.	V		V		V		V		V	
2.	V		V		V		V		V	
3.	V		V		V		V		V	
4.	V		V		V		V		V	
5.	V		V		V		V		V	
6.	V		V		V		V		V	
7.	V		V		V		V		V	
8.	V		V		V		V		V	
9.	V		V		V		V		V	
10.	V		V		V		V		V	



Gambar 122. Tampilan Menu pada Slide kedua

Tabel 73. Tabel Uji Unjuk Kerja Tampilan Data

No .	Data Pengukuran		Tampilan Data					Keterangan
			LCD	EEPROM	Micro SD	Database	Website	
1.	Tinggi Badan & Berat Badan	Tinggi Badan	167,79	167,79	167,79	167,79	167,79	Valid
		Berat Badan	61,24	61,24	61,24	61,24	61,24	Valid
2.	<i>Power Tungkai Vertikal Jump</i>	Tinggi Lompatan	57	57	57	57	57	Valid
		Waktu Lompatan	0,71	0,71	0,71	0,71	0,71	Valid
		Kecepatan Lompatan	0,802	0,802	0,802	0,802	0,802	Valid
		<i>Power</i> Lompatan	49,13	49,13	49,13	49,13	49,13	Valid
3.	<i>Power Tungkai Horizontal Jump</i>	Panjang Lompatan	226	226	226	226	226	Valid
		Waktu Lompatan	2,69	2,69	2,69	2,69	2,69	Valid
		Kecepatan Lompatan	0,840	0,840	0,840	0,840	0,840	Valid
		<i>Power</i> Lompatan	51,41	51,41	51,41	51,41	51,41	Valid
4.	<i>Whole Body Reaction</i>	Reaksi	0,375	0,375	0,375	0,375	0,375	Valid
5.	<i>Power Endurance</i>	Tinggi Lompatan 1	75	75	75	75	75	Valid
		Tinggi Lompatan 2	75	75	75	75	75	Valid
		Tinggi Lompatan 3	76	76	76	76	76	Valid
		Tinggi Lompatan 4	77	77	77	77	77	Valid
		Tinggi Lompatan 5	79	79	79	79	79	Valid
		Tinggi Lompatan 6	78	78	78	78	78	Valid
		Tinggi Lompatan 7	78	78	78	78	78	Valid
		Tinggi Lompatan 8	75	75	75	75	75	Valid
		Tinggi Lompatan 9	75	75	75	75	75	Valid
		Tinggi Lompatan 10	71	71	71	71	71	Valid
		Rata-rata Lompatan	61,8	61,8	61,8	61,8	61,8	Valid
		Maksimal Lompatan	79	79	79	79	79	Valid
		Minimal Lompatan	71	71	71	71	71	Valid

B. Pembahasan

Berdasarkan hasil pengukuran dari alat pengukur *power* tungkai *Power Endurance* dan *Whole Body Reaction*, dapat disimpulkan bahwa secara kuantitas alat ini dapat bekerja dengan baik dan sesuai dengan fungsinya. Namun pada pengukuran beberapa rangkaian sistem terdapat selisih antara hasil pengukuran alat dengan hasil pengukuran sebenarnya. Perbedaan tersebut disebabkan oleh beberapa faktor, seperti kondisi komponen yang kurang baik, tingginya tingkat *error* komponen, dan kurangnya sensitivitas serta kepresisian dari komponen yang digunakan.

1. Analisa Tegangan Catu Daya

a. Tanpa Beban

Setelah dilakukan pengukuran sebanyak 5 kali diperoleh hasil bahwa rata-rata tegangan pada pengukuran catu daya 12V 5A yaitu sebesar 12,22 Volt dengan rata-rata *error* sebesar 1,82%. Pada catu daya 12V 2A memiliki rata-rata tegangan sebesar 12,13 Volt dengan rata-rata *error* sebesar 1,07%. Sedangkan tegangan keluaran pada Modul XL-4015 memiliki tegangan rata-rata sebesar 5,02 Volt dengan rata-rata *error* sebesar 0,3%.

b. Dengan Beban

Setelah dilakukan pengukuran sebanyak 5 kali diperoleh hasil bahwa rata-rata tegangan pada pengukuran catu daya 12V 5A yaitu sebesar 12,11 Volt dengan rata-rata *error* sebesar 0,99%. Pada catu daya 12V 2A memiliki rata-rata tegangan sebesar 12,07 Volt dengan rata-rata *error* sebesar 0,58%. Sedangkan tegangan keluaran pada

Modul XL-4015 memiliki tegangan rata-rata sebesar 5,004 Volt dengan rata-rata *error* sebesar 0,19%.

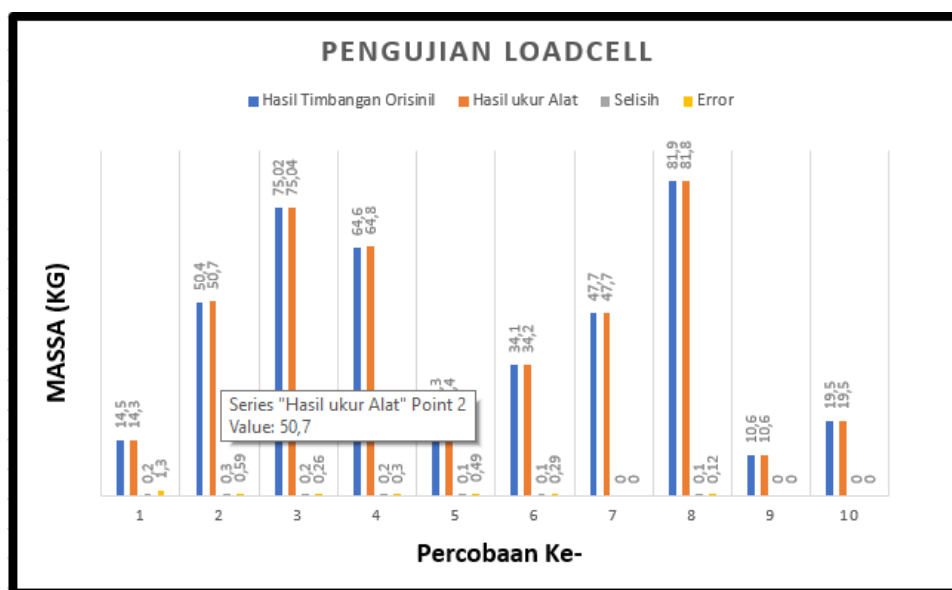
2. Analisa Pengujian Slot *Micro SD Adapter*

Berdasarkan analisis kebutuhan, komponen *micro SD Adapter* digunakan sebagai syarat pertama penggunaan alat yaitu dengan dengan memasang *micro SD Card* ke dalam slot *micro SD Adapter*. Dari hasil pengujian, diperoleh hasil dari 10 kali percobaan slot *micro SD* dapat bekerja dengan baik. Dari 10 kali percobaan tingkat *error* dari kinerja komponen ini adalah 0%, dan hal itu sangat ditekankan sebab semua syarat awal penggunaan alat adalah dari kinerja komponen ini. Hal yang perlu dihindarkan agar kinerja slot *micro SD Adapter* bekerja dengan baik yaitu koneksi kabel antara *micro SD Adapter* dengan mikrokontroler harus benar-benar baik, terutama pada bagian sumber tegangan 5V dan GND.

3. Analisis Pengujian *Loadcell*

Rangkaian *loadcell* pada pembuatan proyek akhir ini berfungsi sebagai pengukur berat badan. Rangkaian *loadcell* ini sendiri terbuat dari hasil oprek timbangan digital yang di ambil kontroler nya dan di ganti menggunakan IC HX711. Hasil dari pengukuran lumayan bagus dari 10 kali percobaan yang dilakukan menggunakan sampel acak. *Loadcell* bekerja sangat baik

dibuktikan dengan data pada diagram dibawah ini.

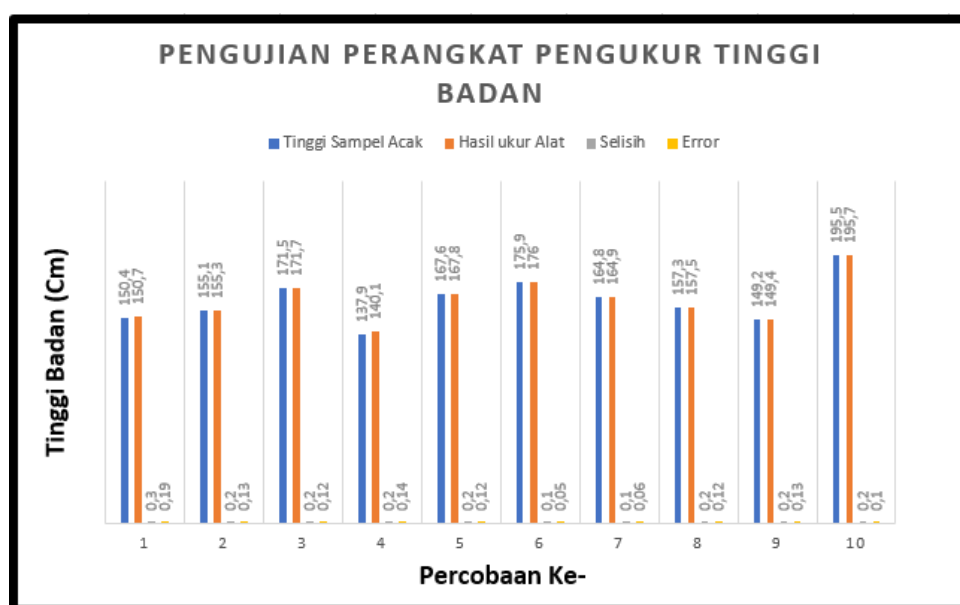


Gambar 123. Diagram Hasil Pengujian Loadcell

Hasil ukur alat sudah memenuhi standar ketelitian dari timbangan digital orisinil pabrikan. Selisih hasil pengukuran dari alat dan timbangan digital orisinil pabrikan sangat kecil hanya memiliki rata-rata selisih sekitar 0,12 kg dan *error* terbesar pada alat yakni mencapai 1,3%. Selisih angka yang terjadi itu disebabkan karena beban muatan yang diukur tidak stabil. Hal ini sangat berpengaruh bagi pembacaan oleh *loadcell*, karena *loadcell* bekerja dengan prinsip deformasi sebuah material akibat adanya tegangan mekanisme yang bekerja. Gaya mekanisme tersebut akan dirubah menjadi sinyal listrik proporsional dengan gaya atau beban yang diterimanya. Apabila *loadcell* diberi beban pada inti besinya, maka besaran resistansi di *strain gauge* akan berubah. Dari perubahan nilai resistensi yang sangat kecil tersebut, kemudian dikuatkan oleh modul HX711 dan dihubungkan pada Arduino untuk dikonversi menjadi satuan berat (Kg).

4. Analisis Pengujian Perangkat Pengukur Tinggi Badan

Data pengujian hasil ukur alat dengan pembanding manual menunjukkan pengukuran yang sangat baik, hal ini dibuktikan dengan 10 kali percobaan hasil pengukuran perangkat ini sangat akurat sesuai dengan jarak yang diukur dengan mencapai ketelitian 0,1 Cm. Perangkat pengukur tinggi badan ini bekerja sangat baik di buktikan dengan diagram dibawah ini.



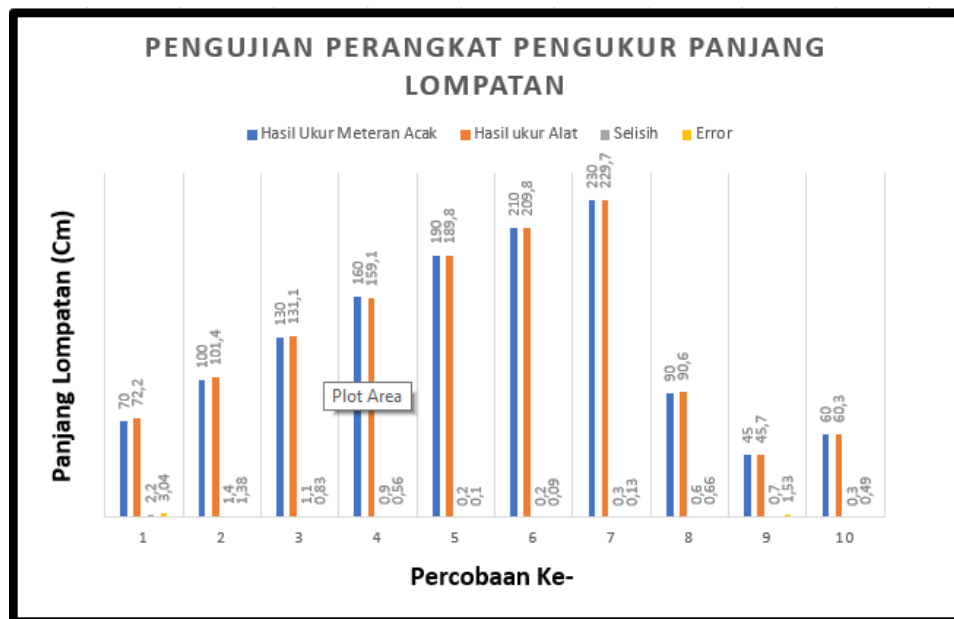
Gambar 124. Diagram Hasil Pengujian Perangkat Pengukur Tinggi Badan

Dari percobaan pertama sampai ke sepuluh, ditunjukkan bahwa hasil pengukuran selalu mendekati nilai ukur sebenarnya dari meteran konvensional. Selisih pengukuran antara alat dengan meteran sangatlah kecil hanya mencapai rata-rata 0,19 Cm tingkat *error* mencapai rata-rata 0,116%. Dengan tingkat kepresisian seperti itu maka sudah mencapai nilai batas standar pengukuran yang paling tidak mencapai maksimal 0,5%. Prinsip kerja dari Perangkat Pengukur Tinggi Badan adalah menggunakan

mekanisme tiang pengukur yang bergerak turun sampai menyentuh ujung kepala, kemudian nilai dari putaran motor penggerak tiang dibaca oleh sensor *rotary encoder* menggunakan perintah interupsi dan di konversi menjadi nilai jarak (Cm) oleh Arduino.

5. Analisis Pengujian Perangkat Pengukur Panjang Lompatan

Dari hasil percobaan pengukuran ketinggian sebanyak 10 kali ini diperoleh bahwa hasil pengukuran alat dapat dikategorikan baik, hal ini dibuktikan dengan perbandingan panjang secara manual yaitu dengan menggunakan meteran. Berikut ini gambar dari grafik hasil pengujian panjang lompatan dengan menggunakan pembanding meteran.



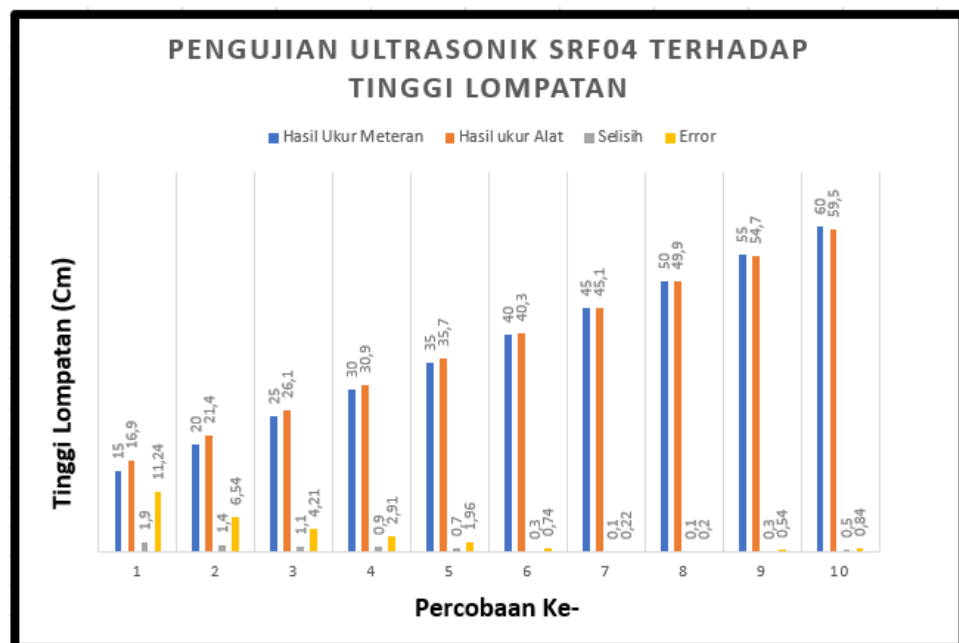
Gambar 125. Diagram Hasil Pengujian Perangkat Pengukur Panjang Lompatan

Pada saat mengukur jarak 70 Cm, terdapat *error* pembacaan sensor sebesar 2,2 Cm (3,04%) yang disebabkan karena kemungkinan mekanik pengukur

panjang lompatan belum pada kondisi *default* nya sehingga pada saat mengukur start awal terlalu mundur sebanyak kurang lebih 2 Cm kemudian saat stop pengukuran nilai hasil pengukurannya bertambah 2 Cm. Faktor lain penyebab ketidak presisian pengukuran panjang lompatan adalah baterai sebagai sumber tegangan pada perangkat ini sudah terlalu habis atau mekanik *slider* terlalu seret sehingga proses pengukuran jadi terhambat.

6. Analisis Pengujian Ultrasonik SRF04 terhadap Tinggi Lompatan

Data pengujian hasil ukur alat dengan pembandingan meteran manual menunjukkan pengukuran yang lumayan baik, hal ini dibuktikan dengan 10 kali percobaan hasil sensor lumayan akurat sesuai dengan jarak yang diukur.



Gambar 126. Diagram Hasil Pengujian Sensor Ultrasonik SRF04 terhadap Tinggi Lompatan

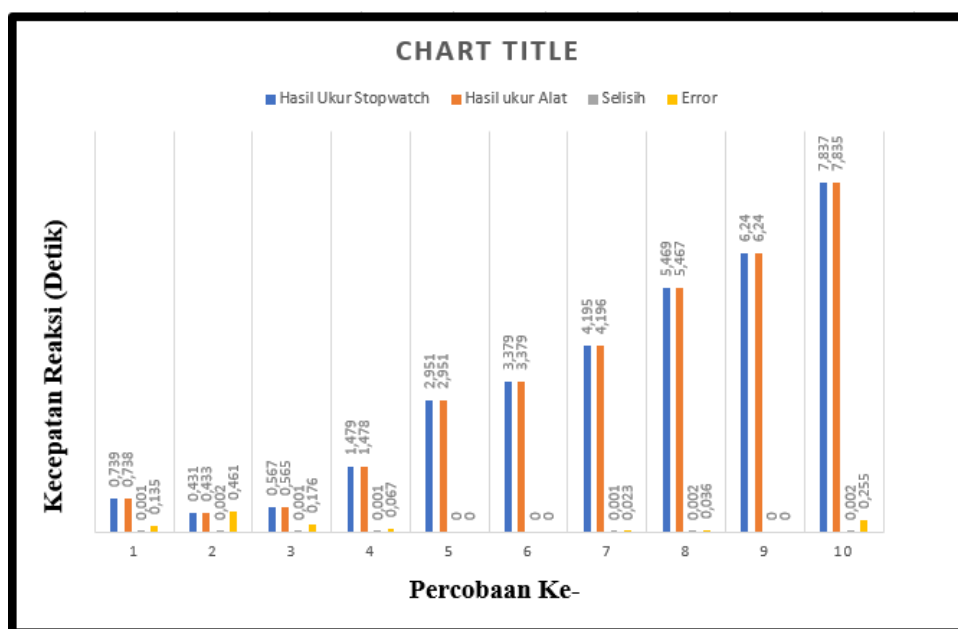
Kenapa dibidang ‘lumayan’ Karena tidak semua hasil pengukuran berjalan dengan baik, masih ada *error* yang lumayan tinggi yaitu 1,9 Cm (11,24%).

Hal ini disebabkan karena pada saat proses kalibrasi penempatan marker

pada kaki kurang pas. Bisa juga karena pada saat pengukuran tinggi lompatan, ketika sensor naik dan sensor turun posisinya tidak stabil. Pemilihan sensor ultrasonik dengan tipe SRF04 karena selama proses riset, komponen ini yang paling stabil dari komponen lain yang pernah dicoba seperti sensor barometer. Pembacaan sensor barometer cukup baik, tapi hanya untuk range diatas 1 meter. Sedangkan kebutuhan untuk alat ini dibawah nilai pengukuran 1 meter.

7. Analisis Pengujian perangkat pengukur *Whole Body Reaction*

Dari hasil pengujian perangkat pengukur *Whole Body Reaction* yang dilakukan sebanyak 10 kali, diperoleh hasil pengukuran alat dapat dikategorikan sangat baik. Hal ini dibuktikan dengan perbandingan pengukuran antara *stopwatch* dan alat. Dibawah ini adalah gambar dari diagram pengujian perangkat pengukur *Whole Body Reaction*.



Gambar 127. Diagram Hasil Pengujian Perangkat Pengukur *Whole Body Reaction*

Berdasarkan data hasil pengujian di atas, selisih antara nilai pengukuran *stopwatch* dan alat sangatlah kecil yaitu mencapai 0,002 detik dan *error* mencapai rata-rata 0,115%. Secara peraturan perangkat ini sudah memenuhi standar pengukuran *Whole Body Reaction*.

8. Analisis Pengujian sistem penyimpanan *micro SD*

Dari hasil pengujian sistem penyimpanan *micro SD* yang dilakukan, sistem ini bekerja dengan baik. Berdasarkan 10 kali percobaan yang dilakukan dengan cara membandingkan data hasil pengukuran, data yang disimpan di EEPROM, dan data yang disimpan di *micro SD*, sistem ini sudah sinkron dan tidak ada selisih maupun *error*.

9. Analisis Pengujian Komunikasi dan *Pairing Bluetooth*

Dari hasil pengujian komunikasi *Bluetooth*, modul *Bluetooth slave 1* dan *slave 2 controller* dapat menerima data dari *Bluetooth master* sesuai dengan data yang dikirimkan. Selain itu pada pengujian komunikasi *Bluetooth* ini, tipe data yang digunakan antara lain seperti *integer (int)*, *float*, *character (char)* dan dapat dikirim dan diterima dengan baik. Kemungkinan data *loss* bisa disebabkan karena *setting baudrate* antara *master* dan *slave controller* berbeda. Untuk *pairing Bluetooth* antara *master controller* dengan *slave controller* juga berjalan dengan baik. Hal itu dibuktikan dari 10 kali percobaan masing-masing 5 kali pada 2 *slave controller* hanya satu kali *loss pairing* antara *Bluetooth master* dengan *Bluetooth slave 1 controller*, disebabkan karena *start pairing* tidak bersamaan. Sehingga dapat disimpulkan bahwa pada pengiriman data antara *Master Controller* dengan *Slave 1 dan Slave 2 Controller* menggunakan *Bluetooth HC05* tidak

terpengaruh dengan tipe data yang digunakan. Sedangkan untuk *pairing Bluetooth* agar mengurangi *loss connection* yaitu dengan cara men-*start Bluetooth* secara bersamaan.

10. Pengujian Sistem Telemetry

Dari hasil pengujian sistem telemetry yang dilakukan, sistem ini bekerja dengan baik. Berdasarkan dari 10 kali percobaan yang dilakukakan dengan cara membandingkan data hasil pengukuran, data yang disimpan di EEPROM, dan data tampil pada *website*, sistem ini sudah sinkron dan tidak ada selisih maupun *error*. Hal yang bisa mengganggu proses pengiriman data dari hasil pengukuran ke *website* adalah koneksi internet dari akses poin tidak boleh terputus saat proses pengukuran sedang berlangsung.

11. Analisa Hasil Pengujia Unjuk Kerja

Uji unjuk kerja yang dilakukan di LPPM UNY pada acara pameran teknologi se Yogyakarta pada bulan Mei lalu. Berdasarkan dari data yang didapat, hasil analisis nya adalah sebagai berikut:

a. Uji Unjuk Kerja pengukuran *Power Tungkai*

Pengukuran *power tungkai vertical jump* dan *horizontal jump* dilakukan sebanyak 10 kali dengan mengambil sampel anak FIK UNY prodi PKO A 2014. Untuk pengukuran *power tungkai* sendiri dibagi menjadi dua yaitu:

1) *Vertikal Jump*

Pengukuran *power tungkai vertical jump* ini meliputi pengambilan data berat badan, tinggi lompatan, dan waktu lompatan. Untuk proses pengambilan data lompatan menggunakan

rumus seleksi nilai terbesar. Pada saat sensor mendeteksi jarak lompatan, maka nilai dari lompatan itu disimpan pada variable *array* dengan batas 100 data. Dari 100 data ini akan di seleksi untuk mendapatkan nilai terbesar, dan itu yang diambil sebagai nilai dari satu kali lompatan. Setelah ketiga data ini diperoleh, akan di lakukan proses pengolahan data untuk menghasilkan nilai *power* dan kecepatan. Pengolahan data ini menggunakan rumus $power = \text{berat badan} * \text{tinggi lompatan} / \text{waktu lompatan}$ dan untuk perhitungan kecepatan adalah $\text{kecepatan} = \text{tinggi lompatan} / \text{waktu lompatan}$. Data hasil perhitungan ini akan di proses lagi untuk di kategorikan berdasarkan umur dan jenis kelamin dengan standar yang telah ada. Kategori untuk penilaian ini dibagi menjadi tujuh klasifikasi yaitu istimewa, baik sekali, baik, cukup, kurang, kurang sekali, dan gagal.

2) *Horizontal Jump*

Pengukuran *power* tungkai *horizontal jump* ini meliputi pengambilan data berat badan, panjang lompatan, dan waktu lompatan. dari ketiga data yang diperoleh ini, akan di lakukan proses pengolahan data untuk menghasilkan nilai *power* dan kecepatan. Pengolahan data ini menggunakan rumus $power = \text{berat badan} * \text{panjang lompatan} / \text{waktu lompatan}$ dan untuk perhitungan kecepatan adalah $\text{kecepatan} = \text{panjang lompatan} / \text{waktu lompatan}$. Data hasil perhitungan ini akan di proses lagi untuk di kategorikan berdasarkan umur dan jenis kelamin dengan standar yang telah ada.

Kategori untuk penilaian ini dibagi menjadi tujuh klasifikasi yaitu istimewa, baik sekali, baik, cukup, kurang, kurang sekali, dan gagal.

b. Uji Unjuk Kerja Pengukuran *Whole Body Reaction*

Pengukuran *Whole Body Reaction* dimaksudkan untuk mengukur kecepatan reaksi atau reflek dari seseorang. Tes ini biasa digunakan untuk melatih para atlet seperti atlet karate, pencak silat, taekwondo, dan bahkan untuk melatih kiper sepak bola. Hasil pengukuran dari alat ini cukup memuaskan. Perbedaan dari alat yang sudah ada, alat ini mampu mengukur kecepatan reaksi lalu langsung menentukan klasifikasi dari kategori jenis kelamin. Cara kerja alat ini adalah mengukur tinggi badan testi terlebih dahulu, kemudian melakukan pengukuran. Testi memilih jenis kelamin terlebih dahulu kemudian memilih jenis pengukuran yang akan dilakukan apakah menggunakan pengukuran Audio atau Visual. Setelah jenis kelamin dan jenis pengukuran sudah dipilih, maka testi akan naik pada papan tumpuan pengukuran yang posisinya diatur supaya berada didepan media reflector cahaya dan suara dengan ketentuan jarak tumpuan dan media pengukuran adalah 3 meter. Setelah testi siap maka pengukuran akan dimulai tanpa aba-aba, dan testi harus siap dan fokus. Pengukuran dilakukan kemudian hasilnya dikategorikan menjadi 6 klasifikasi yaitu istimewa, bagus sekali, bagus, sedang, kurang, kurang sekali dan akan tampil pada LCD box kontrol.

c. Uji Unjuk Kerja Pengukuran *Power Endurance*

Pengukuran *Power Endurance* disini dilakukan untuk mengetahui seberapa kekuatan maksimal atlet dalam melakukan lompatan secara vertikal sebanyak 10 kali. Proses pengambilan data dilakukan secara sekaligus dengan melakukan 10 kali lompatan. Data hasil dari pengukuran yang berjumlah 10 ini akan di seleksi menggunakan rumus seleksi nilai terbesar dan terkecil. Prinsip pengambilan data sama seperti *vertical jump*, yaitu pada saat lompatan dimulai maka data akan disimpan di variabel *array* dengan batas 100 data. Setelah itu dari 100 data yang diambil tadi di seleksi untuk mencari nilai yang paling besar. Proses ini dilakukan sebanyak 10 kali lompatan. Setelah 10 kali lompatan, data akan dikirim ke *Master Controller*. Data yang diterima oleh *Master Controller* akan di proses lagi menggunakan rumus *power Tungkai vertical* $\text{berat} \times \text{jarak} / \text{waktu}$ kemudian dapat menentukan nilai terbesar, terkecil, dan rata-rata dari 10 *power endurance* dari data lompatan yang dilakukan.

d. Uji Unjuk Kerja Tampilan Data

Pengujian ini dilakukan untuk memastikan transmisi data mulai dari proses pengukuran sampai proses penampilan data tidak terjadi kendala.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan terhadap Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* Pada Atlet dapat disimpulkan sebagai berikut :

1. Untuk membangun Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* Pada Atlet, dibutuhkan perancangan perangkat keras dengan menggabungkan antara mekanik dan elektronik. Mekanik dari alat ini dibuat menggunakan rangka berbahan *aluminium profile*. Elektronik dari alat ini merupakan gabungan beberapa komponen dan rangkaian, diantaranya yaitu: timbangan digital yang sudah di oprek dan ditambahkan IC HX711, sensor Ultrasonik, *Rotary Encoder*, *Proximity*, rangkaian *driver motor H-Bridge*, LCD, *buzzer*, motor Servo, motor DC *power window*, motor DC *gearbox* 400 rpm, dan *Bluetooth* HC-05. Mempunyai empat buah sistem kontrol antara lain Arduino Mega (*Master Controller*), Arduino NANO 1 (*Slave 1 Controller*), Arduino NANO 2 (*Slave 2 Controller*), dan Wemos D1 (*Slave 3 Controller*).
2. Dalam proses pembuatan *software*, perangkat lunak yang diaplikasikan dalam sistem ini adalah program yang dibangun dengan *software* pemrograman Arduino IDE. Berdasarkan pengujian yang telah dilakukan, perangkat lunak ini dapat bekerja dengan baik untuk memprogram empat buah kontroller yang digunakan. Kontrol *Master* mampu mengendalikan *Slave 1*, *Slave 2*, dan juga mengirim data ke Wemos untuk di *upload* ke

database website secara *online*. Selain itu perangkat lunak lain yang digunakan adalah Notepad++. *Software* ini berfungsi sebagai *text editor* digunakan untuk memprogram *website*.

3. Unjuk kerja Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* pada Atlet secara keseluruhan telah berjalan sesuai dengan yang diharapkan yaitu:

a. *Master* Kontroler

1) Saat box kontrol utama dinyalakan dan memori *micro SD card* dipasang maka sistem utama akan berjalan, begitu pula sebaliknya jika ketika box kontrol utama dinyalakan dan memori *micro SD card* tidak dipasang maka sistem tidak berjalan dan akan memberi peringatan 'masukkan *micro SD card*'.

2) Ketika masuk proses *booting* dan tiang pengukur tinggi badan masih dalam keadaan dibawah, maka tiang pengukur tinggi badan akan naik ke posisi *stanby* atas.

3) Masuk menu *stanby*.

4) Jika tombol OK ditekan maka akan masuk menu utama yang isinya *Scan Tinggi dan Berat*, *Power Tungkai [AUTO]* atau *Power Tungkai [MANU]* tergantung mana yang dipilih dengan menekan tombol *BACK*, *W. Body Reaction*, *Power Endurance*, Turunkan Tiang, Lihat Data Kelompok, Informasi *micro SD*, dan Pengaturan Penyimpanan.

5) Jika melakukan

b. *Slave 1* Kontroler

- 1) Ketika saklar dinyalakan dan sensor menghadap kebawah maka sensor akan otomatis kalibrasi kemudian buzzer dan led indikator akan nyala tiga kali.
- 2) Ketika mendapat perintah dari *Master* Kontroler dengan kode perintah “1” maka sistem akan memulai pengukuran *Power Endurance*.
- 3) Ketika mendapat perintah dari *Master* Kontroler dengan kode perintah “5” maka sistem akan memulai pengukuran *Vertikal Jump*.
- 4) Ketika mendapat perintah dari *Master* Kontroler dengan kode perintah “3” maka sistem akan melakukan kalibrasi sensor.

c. *Slave 2* Kontroler

- 1) Ketika mendapat perintah dari *Master* Kontroler dengan kode perintah “1” maka sistem akan memulai pengukuran *Power Endurance*.
- 2) Saat pengukuran dilakukan dan perangkat pengukur panjang lompatan mendeteksi kaki maka keadaan menjadi berhenti dan membaca nilai sensor *rotary encoder* untuk kemudian di konversi menjadi satuan jarak (Cm). Setelah itu data dikirimkan ke *Master Controll* kemudian perangkat pengukur panjang lompatan kembali ke posisi *default*.
- 3) Ketika pengukuran panjang lompatan dimulai dan perangkat tidak mendeteksi kaki, maka mekanik *slider* akan kembali ke posisi *default* secara otomatis.

d. *Slave 3* Kontroler

Ketika ada data pengukuran yang masuk dari *Master Controller* maka secara otomatis perangkat ini mem parsing data untuk kemudian dikirimkan ke *database* secara *online* dengan syarat harus ada koneksi internet.

e. Sistem *Database Online*

Ketika ada data pengukuran yang masuk dari *Slave 3 Controller* maka secara otomatis perangkat ini mem parsing data untuk kemudian disimpan pada *database* tabel yang telah ada. Kemudian dari *database* tabel yang telah terisi data tadi di tampilkan pada halaman website www.smartech515.xyz

B. Keterbatasan Alat

Alat Pengukur *Power Tungkai*, *Power Endurance*, dan *Whole Body Reaction* pada Atlet yang telah dibuat memiliki keterbatasan, antara lain :

1. Alat ini masih bersifat parsial, yaitu satu alat untuk satu pengguna dan dilakukan secara bergantian.
2. Alat ini bersifat fix, setiap kali ingin menggunakan harus melakukan beberapa setting dan susah kalau mau dibawa. Dikarenakan dimensi dari alat ini yang cukup besar.
3. Sistem pengukuran *vertikal jump* masih kurang maksimal masih ada *error* beberapa kali, karena belum menemukan sensor yang pas dan akurat untuk mengukur Tinggi Lompatan.
4. Hasil pengukuran masih berbentuk data digital, belum ada hasil cetak diatas kertas.

C. Saran

Pembuatan proyek akhir ini ternyata terdapat beberapa kekurangan sehingga diperlukan pengembangan lebih lanjut. Saran yang membangun dibutuhkan untuk menyempurnakan proyek akhir ini, antara lain sebagai berikut :

1. Sistem pengukuran *power tungkai*, *Power Endurance*, dan *Whole Body Reaction* dapat dikembangkan untuk pengukuran secara parallel agar jumlah atlet yang melakukan pengukuran dalam sekali kerja alat bisa bertambah. Hal ini sangat membantu saat proses seleksi atlet.
2. Mendesain ulang alat dan memperhitungkan aspek-aspek keolahragaan agar lebih praktis untuk dibawa tidak menyalahi norma-norma pengukuran pada atlet.
3. Harus melakukan riset lagi dan lagi supaya dapat menemukan sensor yang cocok untuk pengukuran *vertical jump*.
4. Hasil *output* data sebaiknya langsung dicetak menggunakan printer termal agar pengguna bisa langsung mengetahui hasil pengukuran secara langsung.

DAFTAR PUSTAKA

- Anugrah, M. D. (2015). Aplikasi Sensor Ultrasonik SRF05 Pada Robot Vacuum Cleaner Menggunakan Kendali Android Berbasis Mikrokontroller Atmega 8535. Diakses pada tanggal 30 April 2017 pukul 12.00 WIB dari <http://eprints.polsri.ac.id/2782/>
- Arduino. (2013). Arduino Mega 2560. Diakses pada tanggal 28 April 2017 pukul 14.23 WIB, dari <http://arduino.cc/en/Main/ArduinoBoardMega2560/>
- Arduino. (2013). Arduino Nano. Diakses pada tanggal 28 April 2017 pukul 14.47 WIB, dari <https://arduino.cc/en/Main/ArduinoBoardNano/>
- Azlina, Maya. (2015). Pembuatan Alat Ukur Kecepatan Angin Dan Penunjuk Arah Angin Berbasis Mikrokontroller At-Mega8535. *Skripsi*, tidak dipublikasikan. Universitas Pendidikan Indonesia.
- Bejo, Agus. (2008). C dan AVR Rahasia kemudahan bahasa C dalam mikrokontroler. Yogyakarta: graha ilmu
- Buaya-Instrument. (2016). Panduan Menggunakan Baterai Lipo. Diakses pada tanggal 30 April 2017 pukul 23.11, dari <http://buaya-instrument.com/blog-buaya-instrument/cara-menggunakan-baterai-lipo>
- Firmansyah, Dhea. (2013). Simulasi Pintu Pagar Berbasis Mikrokontroller Arduino. *Skripsi*, tidak dipublikasikan. Universitas Pendidikan Indonesia.
- Elektronika Dasar. (2012). Motor Servo. Diakses pada tanggal 29 April 2017 pukul 01.37 WIB dari <http://elektronika-dasar.web.id/motor-servo/>
- Pratama, Fery. (2013). Robot Pemadam Api Berkaki Enam Berbasis Mikrokontroller ATmega32 Dan ATmega128. *Proyek Akhir*, Universitas Negeri Yogyakarta.
- Hutapea, N. S. (2008). Aplikasi Sensor Pada Robot Penampung Sampah. Diakses pada tanggal 30 April 2017 pukul 15.42 WIB dari digilib.mercubuana.ac.id/manager/n!@file_skripsi/files133719009763.pdf
- Indo-ware. (2013). Manual MicroSD Card Adapter. Diakses pada tanggal 14 Mei 2017 pukul 23.39 WIB dari <http://www.indo-ware.com/produk-2735-microsd-card-modul-spi-antarmuka-mini-card-reader-tf.html>
- Irvan, H. (2014). Hubungan *Whole Body Reaction Time* Dengan Antisipasi Penjaga Gawang Pada saat Tendangan *Penalty* Dalam Olahraga Sepakbola. *Skripsi*, tidak dipublikasikan. Universitas Pendidikan Indonesia.
- Jonath, U. (1985). *Atletik*. Jakarta: Rusda Jaya Putra.
- Maulana, Ridwan. (2010). Hubungan *Power Tungkai*, Kekuatan Otot Lengan dan Tinggi Badan Terhadap Kemampuan *Lay Up Shoot* Pada Siswa Putra SLTP

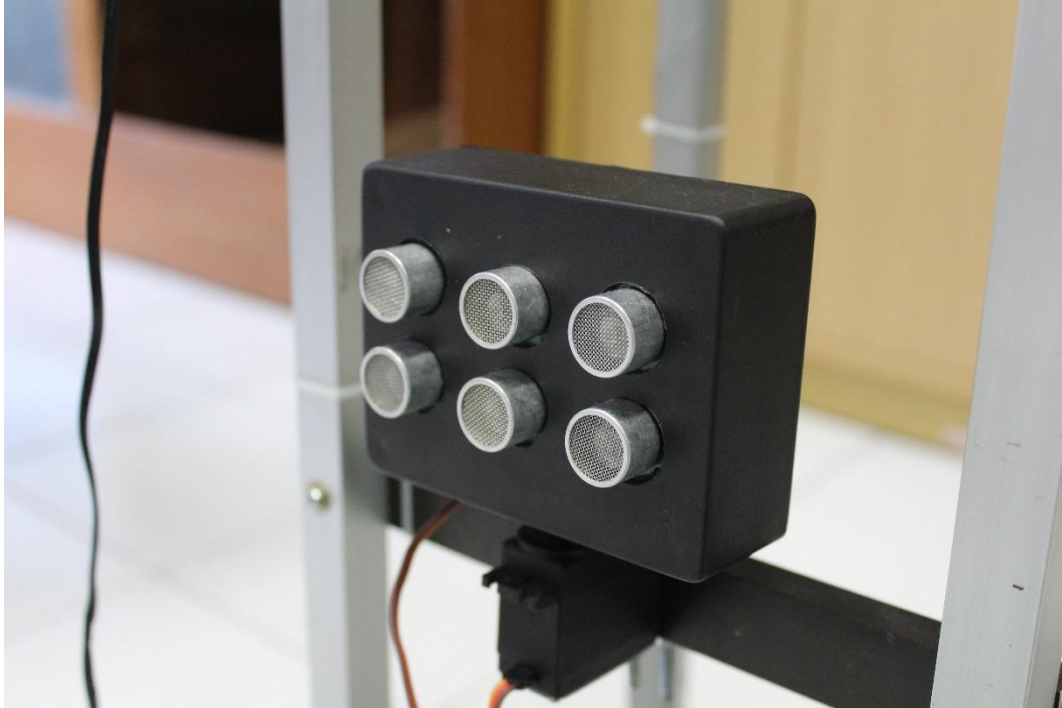
N II Arjosari Kabupaten Pacitan Jawa Timur Yang Mengikuti Ekstrakurikuler Bolabasket. *Skripsi*. Universitas Negeri Yogyakarta.

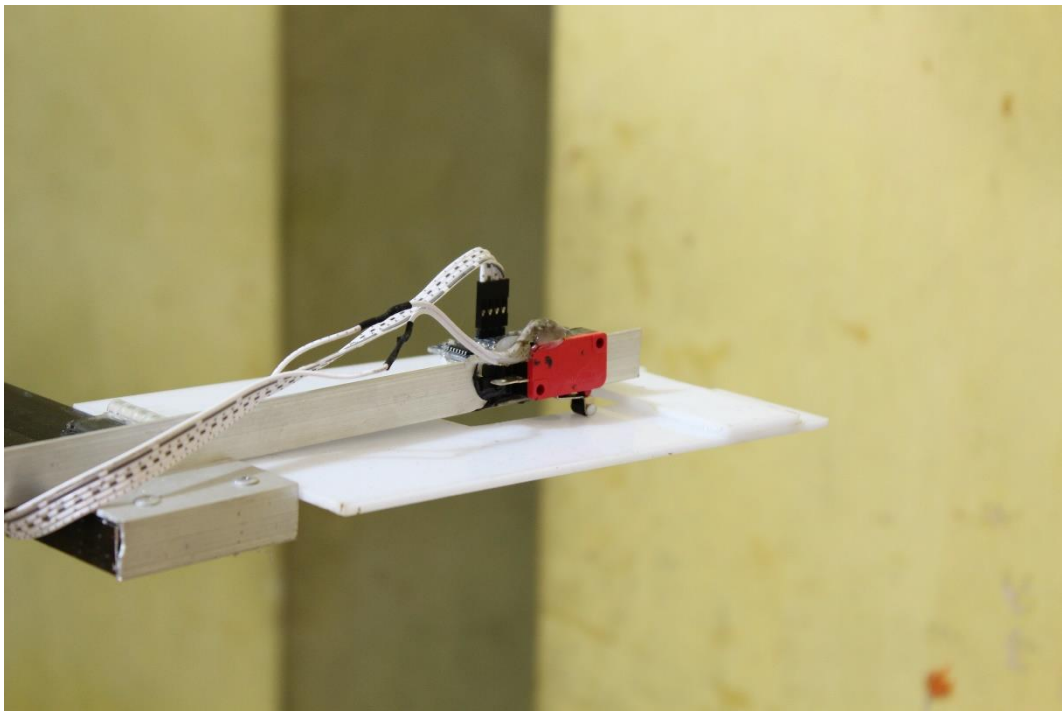
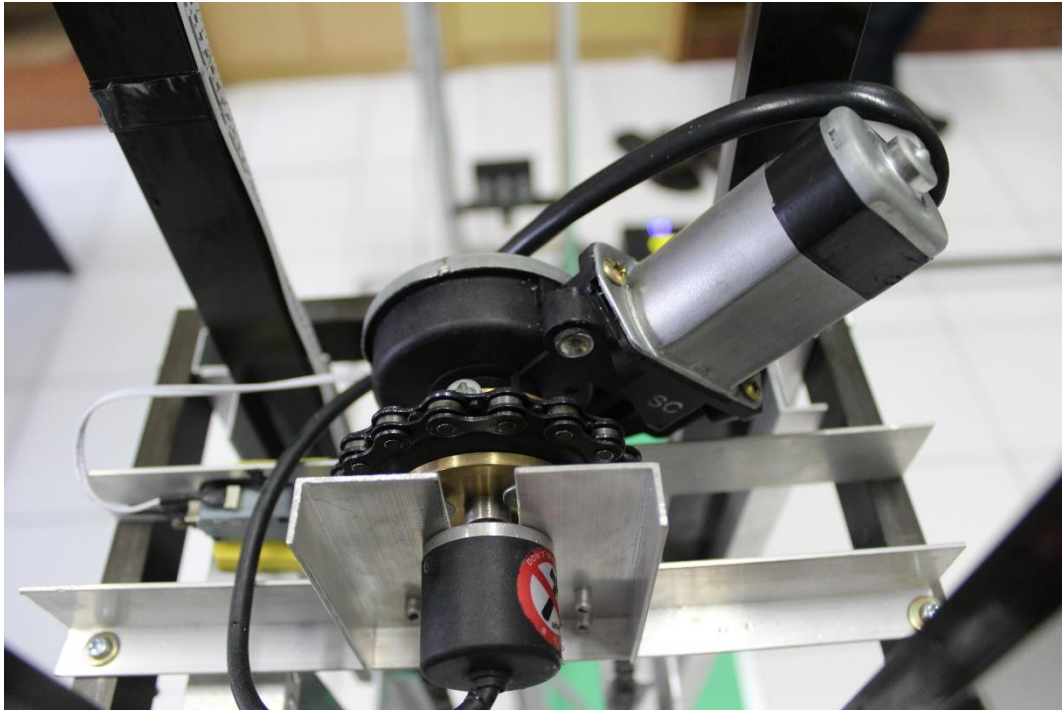
- Miller, D.K. (2002). *Measurement by the Physical Educator: Why and How*. New York: The McGraw-Hill Companies, Inc. Phillips , D. A. (1942). *Measurement And Evaluation In Physical Education*. Canada: United states of America.
- Ridwan Maulana. (2010). Hubungan Power Tungkai, Kekuatan Otot Lengan Dan Tinggi Badan Terhadap Kemampuan Lay Up Shoot Pada Siswa Putra SLTP N II Arjosari Kabupaten Pacitan Jawa Timur Yang Mengikuti Ekstrakurikuler Bolabasket. *Skripsi*. Yogyakarta: UNY.
- Sajoto M. (1988). *Pembinaan Kondisi Fisik Dalam Olahraga*. Jakarta: Departemen Pendidikan dan Kebudayaan Direktorat Jendral Pendidikan Tinggi Proyek Pembinaan Tenaga Kependidikan.
- _____. (1998). *Peningkatan dan Pembinaan Kekuatan dan Kondisi Fisik dalam Olahraga*. Semarang: Dahara Prize.
- Sasongko, B. H. (2012). *Pemrograman Mikrokontroler Dengan Bahasa C*. Yogyakarta: Andi
- Shobirin, Muhannad. (2016). *Pembuatan Sistem Kendali Robot Bulutangkis Wireless*. *Proyek Akhir*, Universitas Negeri Yogyakarta.
- Simanungkalit, P. A. (2016). *Pembuatan Prototype Monitoring Volume dan Suhu Oli Menggunakan Handphone Android*. Diakses pada tanggal 29 April 2017 pukul 22.00 WIB dari <http://repository.usu.ac.id/handle/123456789/62690>
- Sinauarduino. (2016). *Mengenal Arduino Software (IDE)*. Diakses pada tanggal 28 April 2017 pukul 23.51 WIB dari <http://www.sinauarduino.com/artikel/mengenal-arduino-software-ide/>
- Sinuhaji, Perdinan. (2015). *Sitem Perancangan Pengendalian Tekanan Digital Menggunakan IC HX711 Berbasis Mikrokontroller Atmega32 Pada Cetakan Komposit*. Diakses pada tanggal 30 April 2017 pukul 14.18 dari <http://repository.usu.ac.id/handle/123456789/52952>
- Wemos. (2016). *Wemos D1 mini*. Diakses pada tanggal 28 April 2017 pukul 15.02 WIB, dari <https://www.wemos.cc/product/d1-mini>
- Yahya, Sofian. (2009). *Kendali Putaran Motor DC dengan PWM*. Diakses pada tanggal 28 April 2017 pukul 13.54 WIB, dari <http://www.academia.edu/>

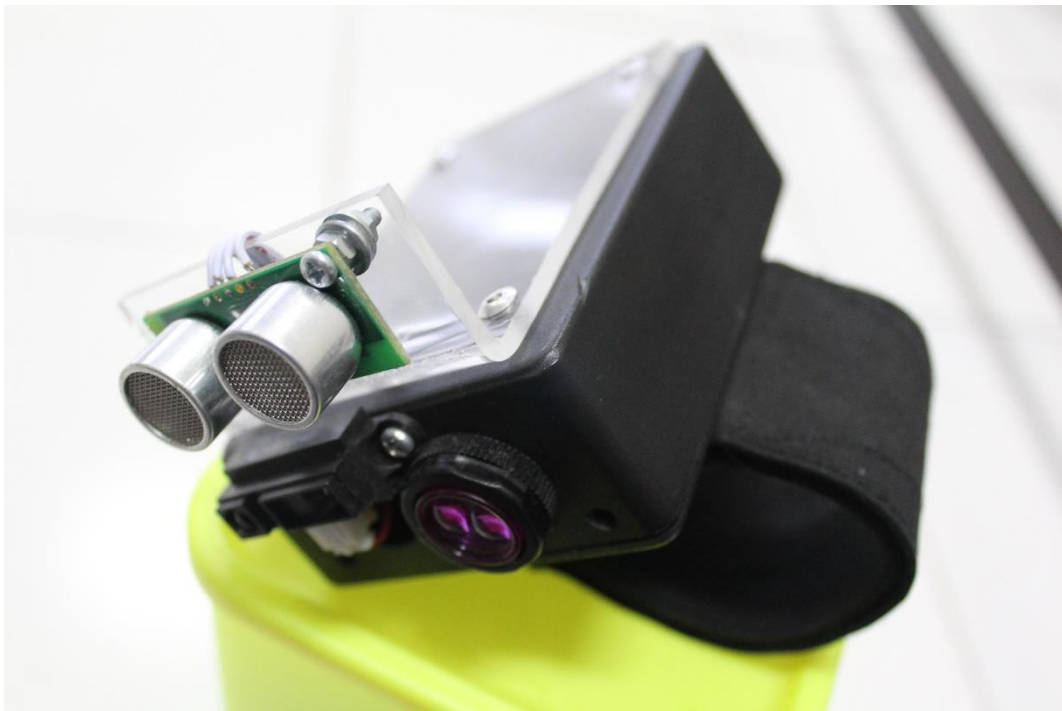
LAMPIRAN

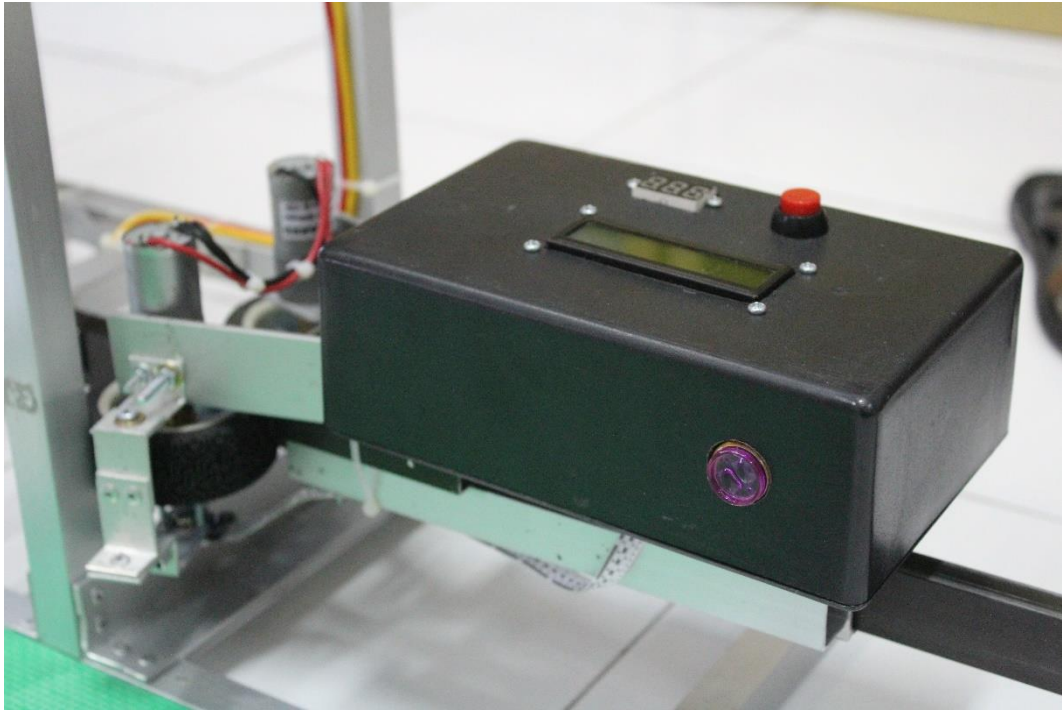
Lampiran 1. Gambar Alat











Lampiran 2. Program *Master Controller* (Arduino Mega)

```

/*****
This program was produced by the
Arduino 1.6.5
Automatic Program Generator
http://www.tokosik.esy.es

Project : TUGAS AKHIR
Version :
Date    : 21/03/2017
Author  : Rizky Ziky
Company : Microsoft
Comments:

Chip type           : ATmega2560
Program type        : Application
AVR Core Clock frequency: 16,000000 MHz
Memory model        : Small
External RAM size   : 0
Data Stack size     : 256
*****/

#include <SPI.h>
#include <SD.h>
#include <Servo.h>
#include <LiquidCrystal.h>
#include <EEPROM.h>
#include <StopWatch.h>
#include "HX711.h"
#define button_down  bDown<100
#define button_up    bUp<100
#define button_ok    bOk<100
#define button_SAVE  bSAVE<100
#define button_BACK  bBACK<100
#define ada_orang    tumpuan <20
#define power        power_fix
#define echo         31
#define triger       33
#define echo1        26
#define triger1      28
#define echo2        24
#define triger2      22
#define echo3        30
#define triger3      32
HX711 scale(12,13);
StopWatch sw_millis; // MILLIS (default)
StopWatch sw_micros(StopWatch::MICROS);
StopWatch sw_secs(StopWatch::SECONDS);
Sd2Card card;
SdVolume volume;
SdFile root;
LiquidCrystal lcd_ziky(2,3,4,5,6,7);
Servo zios_servo;
File grp1;
File grp2;
File grp3;
File grp4;
File grp5;

int posisi=30;

//VARIABEL ALAMAT EEPROM
int addr_TB = 1;
int addr_BB = 2;
int addr_VJ = 3;
int addr_HJ = 4;
int addr_kal= 5;
int addr_kiriman_client1 depan koma=6;

```

```

int addr_kiriman_client1_belakang_koma=7;
int addr_HJ_plus = 8;
int addr_HJ_belakang_koma = 9;
int addr_kiriman_client2_depan_koma=10;
int addr_kiriman_client2_belakang_koma=11;
int addr_TB_belakang_koma = 12;
int addr_BB_belakang_koma = 13;
int addr_mPower = 14;
int addr_WBR = 15;
int addr_WBR_belakang_koma = 16;
int addr_PE1 = 17;
int addr_PE2 = 18;
int addr_PE3 = 19;
int addr_PE4 = 20;
int addr_PE5 = 21;
int addr_PE6 = 22;
int addr_PE7 = 23;
int addr_PE8 = 24;
int addr_PE9 = 25;
int addr_PE10 = 26;
int addr_rata2_PE = 27;
int addr_min_PE = 28;
int addr_max_PE = 29;

//VARIABEL BACA EEPROM
int read_eeprom_TB=0;
int read_eeprom_BB=0;
int read_eeprom_VJ=0;
int read_eeprom_HJ=0;
int read_eeprom_kal=0;
int read_eeprom_kiriman_client1_depan_koma=0;
int read_eeprom_kiriman_client1_belakang_koma=0;
int read_eeprom_HJ_plus = 0;
int read_eeprom_HJ_belakang_koma = 0;
int read_eeprom_kiriman_client2_depan_koma=0;
int read_eeprom_kiriman_client2_belakang_koma=0;
int read_eeprom_TB_belakang_koma = 0;
int read_eeprom_BB_belakang_koma = 0;
int read_eeprom_mPower = 0;
int read_eeprom_WBR=0;
int read_eeprom_WBR_belakang_koma=0;
int read_eeprom_PE[10];
int read_eeprom_rata2_PE=0;
int read_eeprom_min_PE=0;
int read_eeprom_max_PE=0;

String group_test_vertikal_jump="group_test_vertikal_jump.txt";
String group_test_horizontal_jump="group_test_horizontal_jump.txt";
String group_test_power_endurance="group_test_power_endurance.txt";
String group_test_whole_body_reaction="group_test_whole_body_reaction.txt";
String dataIn_kal_PE;
String dataIn_kal_VJ;
String dataIn1;
String dataIn2;
String dataIn3;
String dataIn4;
String dataIn5;
String dt1[10];
String dt2[10];
String dt3[10];
int rata2tinggi_lompat;
int terbesar,terkecil,terbesar_loncat1,terbesar_loncat2;
int i;
boolean parsing=false;

String sTB="";
String sBB="";
String sVJ="";
String s_T_LOM_VJ="";
String s_waktu_VJ="";
String s_kec_VJ="";
String s_power_VJ="";

```

```

String s_P_LOM_HJ="";
String s_waktu_HJ="";
String s_kec_HJ="";
String s_power_HJ="";
String sWBR="";
String sPE1="";
String sPE2="";
String sPE3="";
String sPE4="";
String sPE5="";
String sPE6="";
String sPE7="";
String sPE8="";
String sPE9="";
String sPE10="";
String s_rata2PE="";
String s_min_PE="";
String s_max_PE="";

byte cinta[8] =
{0b00000,0b01010,0b11111,0b11111,0b11111,0b01110,0b00100,0b00000};
byte booting[8] =
{0b00000,0b01110,0b11111,0b11111,0b11111,0b11111,0b01110,0b00000};
byte lineup[8] =
{0b11111,0b00000,0b00000,0b00000,0b00000,0b00000,0b00000,0b00000};
byte cursor_segitiga[8] =
{0b01000,0b01100,0b01110,0b01111,0b01110,0b01100,0b01000,0b00000};
byte maxim[8]
={0b00100,0b01110,0b11111,0b01110,0b01110,0b01110,0b01110,0b00000};
byte minim[8]
={0b01110,0b01110,0b01110,0b01110,0b11111,0b01110,0b00100,0b00000};
byte ratarata[8]
={0b11011,0b11011,0b11011,0b11011,0b11011,0b11011,0b11011,0b11011};
byte batas[8]
={0b00100,0b00100,0b00100,0b00100,0b00100,0b00100,0b00100,0b00100};

const int pin_button = A0;
const int proximity = A2;
const int sensorPin = A3;
const int senka = A6;
const int senki = A7;
const int buzzer = 47;
const int buzzer_WBR = 37;
const int visual_WBR = 39;

//KONFIGURASI PIN DRIVER MOTOR
const int dir_motor = 11; // inisiasi pin untuk motor kiri
const int pwm_motor = 10;

//KONFIGURASI ENCODER
int encoderPin1 = 21;
int encoderPin2 = 20;
volatile int lastEncoded = 0;
volatile long encoderValue = 0;
int lastMSB = 0;
int lastLSB = 0;
int data_enc=0;
long lastencoderValue = 0;

//LIMIT SWITCH REL MOTOR
const int limit_atas=43;
const int limit_bawah=45;
const int limit_kepala=8;
int val_bawah=0;
int val_atas=0;
int val_kepala=0;

//DEKLARASI VARIABLE
float sensor_jarak, Hasil_read_ADC_Jarak;
float sensor_jarak1, Hasil_read_ADC_Jarak1;
float sensor_jarak2, Hasil_read_ADC_Jarak2;

```

```

float Analog_read_ADC_Jarak, hasil_sensor_jarak, hasil_jarak,
sampling_data_jarak;
float Analog_read_ADC_Jarak1, hasil_sensor_jarak1, hasil_jarak1,
sampling_data_jarak1;
float Analog_read_ADC_Jarak2, hasil_sensor_jarak2, hasil_jarak2,
sampling_data_jarak2;

//VARIABEL TINGGI BADAN 2
float jarak_fix_TB;
float jarak_loncat;
float tiang1,tiang2,H1; //variabel integer H2,HT,H1
float TB=0;
float TB_enc;
int pembagi_TB;
float hasil_bagi_TB;
float hasil_belakang_koma_TB;
float TB_dijumlah;
float TB_fix;

//VARIABEL PEMBAGI WAKTU WBR
int pembagi_WBR;
float hasil_bagi_WBR;
float hasil_belakang_koma_WBR;
float WBR_dijumlah;
float WBR_fix;

//VARIABEL BERAT BADAN
int pembagi_BB;
float hasil_bagi_BB;
float hasil_belakang_koma_BB;
float BB_dijumlah;
float BB_fix;

//VARIABEL PROXIMITY
int jarak_fix;
int prox, hasil_proximity;

//VARIABEL SENSOR DETEKSI ORANG
int jarakD1=0,jarakD2=0,jarakD3=0;

//VARIABEL BUTTON PEMBAGI TEGANGAN
unsigned int bDown,bUp,bOk,bSAVE,bBACK,tumpuan;
int menu=0;
int mPower=1;
int kelompok=1;
int format=1;
int data_kelompok=1;
int user=1;

//VARIABEL RUMUS POWER VERTIKAL JUMP & HORIZONTAL JUMP
float jarak_HJ;
float jarak_HJ_fix;
float waktu_dijumlah_VJ;
float waktu_dijumlah_HJ;
float waktu_fix_VJ;
float waktu_fix_HJ;
float kecepatan_fix_VJ;
float kecepatan_fix_HJ;
float teta_power1_VJ=0, teta_power2_VJ=0, teta_power3_VJ=0, power_fix_VJ=0,
horse_power_VJ=0;
float teta_power1_HJ=0, teta_power2_HJ=0, teta_power3_HJ=0, power_fix_HJ=0,
horse_power_HJ=0;
float v1,v11,v2,v22,v3=0,v4=0,v5=0,v6=0,v7=0,v8=0,v9=0;
float h1=0,h2=0,h3=0,h4=0,h5=0,h6=0,h7=0,h8=0;
float g = 9.8;

//BOOLEAN
byte bool_BMI=1;
byte bool_PT=1;
byte bool_WBR=1;
byte bool_PE=1;
byte bool_menu=0;

```



```

    while (grp1.available()) { Serial.write(grp1.read()); }
  grp1.close(); Serial.println();
  while (grp2.available()) { Serial.write(grp2.read()); }
  grp2.close(); Serial.println();
  while (grp3.available()) { Serial.write(grp3.read()); }
  grp3.close(); Serial.println();
  while (grp4.available()) { Serial.write(grp4.read()); }
  grp4.close(); Serial.println();
  while (grp5.available()) { Serial.write(grp5.read()); }
  grp5.close(); Serial.println();
  Serial.println("SELESAI MEMBACA
FILE=====
==");
}
else
{
  Serial.println("gagal membuka file");
}

//=====
=====

setPWMfrequency(0x01);
pinMode(echo, INPUT);
pinMode(triger, OUTPUT);
pinMode(echo1, INPUT);
pinMode(triger1, OUTPUT);
pinMode(echo2, INPUT);
pinMode(triger2, OUTPUT);
pinMode(echo3, INPUT);
pinMode(triger3, OUTPUT);
pinMode(sensorPin, INPUT);
pinMode(senka, INPUT);
pinMode(senki, INPUT);
pinMode(proximity, INPUT);
pinMode(buzzer, OUTPUT);
pinMode(buzzer_WBR, OUTPUT);
pinMode(visual_WBR, OUTPUT);
pinMode(dir_motor, OUTPUT);
pinMode(pwm_motor, OUTPUT);
pinMode(A8, INPUT_PULLUP);
pinMode(A9, INPUT_PULLUP);
pinMode(A10, INPUT_PULLUP);
pinMode(A11, INPUT_PULLUP);
pinMode(A12, INPUT_PULLUP);
pinMode(A14, INPUT_PULLUP);
pinMode(limit_atas, INPUT_PULLUP);
pinMode(limit_bawah, INPUT_PULLUP);
pinMode(limit_kepala, INPUT_PULLUP);
pinMode(encoderPin1, INPUT_PULLUP);
pinMode(encoderPin2, INPUT_PULLUP);
digitalWrite(encoderPin1, HIGH); //turn pullup resistor on
digitalWrite(encoderPin2, HIGH); //turn pullup resistor on
digitalWrite(buzzer_WBR, HIGH); //turn pullup resistor on
digitalWrite(visual_WBR, HIGH); //turn pullup resistor on
zios_servo.attach(9);

lcd_ziky.createChar(8, batas);
lcd_ziky.createChar(1, cinta);
lcd_ziky.createChar(2, booting);
lcd_ziky.createChar(3, lineup);
lcd_ziky.createChar(4, cursor_segitiga);
lcd_ziky.createChar(5, maxim);
lcd_ziky.createChar(6, minim);
lcd_ziky.createChar(7, ratarata);
tiang1=205;
tiang2=129;
scale.set_scale(2280.f);
scale.tare(); // reset the scale to 0
silahkanTunggu();
hasil_proximity=0;
timbangan=0;

```

```

dataIn_kal_PE="";
dataIn_kal_VJ="";
dataIn1="";
dataIn2="";
dataIn3="";
dataIn4="";
dataIn5="";
zios_servo.write(95);
delay(100);
attachInterrupt(0, updateEncoder, CHANGE);
attachInterrupt(1, updateEncoder, CHANGE);
read_eeprom_mPower=EEPROM.read(addr_mPower);
if(read_eeprom_mPower==1)mPower=1;
else if(read_eeprom_mPower==2)mPower=2;
while(1)
{
  val_atas=digitalRead(limit_atas);
  if(val_atas==HIGH){lcd_ziky.clear();goto tiang_naik;}
  else{lcd_ziky.clear();break;}
}
tiang_naik:
while(1)
{
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Reset Posisi Tiang");
  motor_naik();
  val_atas=digitalRead(limit_atas);
  if(val_atas==LOW)
  {
    digitalWrite(buzzer,HIGH);
    motor_stop();
    encoderValue=0;
    delay(100);
    digitalWrite(buzzer,LOW);
    lcd_ziky.clear();
    break;
  }
}
}

int pembagi_waktu;
float hasil_bagi_waktu;
float hasil_belakang_koma;

void wektu()
{
  waktu=sw_micros.elapsed();
  teta_waktu=waktu/1000000;
  pembagi_waktu=teta_waktu;
  hasil_bagi_waktu=teta_waktu-pembagi_waktu;
  hasil_belakang_koma=hasil_bagi_waktu*100;
}

void loop()
{
  tampilan_stanBy();
  utama();

  // paket_data_nodeMcu();
  // zios_servo.write(65);
  // zios_servo.write(115);
  // deteksi1();
  // deteksi2();
  // deteksi3();
  // Serial.println();
  //delay(100);
}

void utama()
{
  kalibrasi_button();
  if(button_ok)

```

```

{
  lcd_ziky.clear();
  delay(200);
  atas:
  while(1)
  {
    read_eeprom_mPower=EEPROM.read(addr_mPower);
    kalibrasi_button();
    if(button_down)
    {
      menu++;
      if(menu>7)menu=0;
      if(menu==4)lcd_ziky.clear();
      delay(100);
    }
    if(button_up)
    {
      menu--;
      if(menu<0)menu=7;
      if(menu==7)lcd_ziky.clear();
      delay(100);
    }
    if(menu==0)
    {
      lcd_ziky.setCursor(0,0);
      lcd_ziky.write(1);
      lcd_ziky.setCursor(1,0);
      lcd_ziky.print("Scan Berat & Tinggi");
      lcd_ziky.setCursor(0,1);
      lcd_ziky.print(" Power Tungkai");
      lcd_ziky.setCursor(0,2);
      lcd_ziky.print(" W Body Reaction  ");
      lcd_ziky.setCursor(0,3);
      lcd_ziky.print(" Power Endurance  ");
      if(read_eeprom_mPower==1)
      {
        lcd_ziky.setCursor(14,1);
        lcd_ziky.print("[MANU]");
      }
      if(read_eeprom_mPower==2)
      {
        lcd_ziky.setCursor(14,1);
        lcd_ziky.print("[AUTO]");
      }
      if(button_ok)
      {
        lcd_ziky.clear();
        delay(200);
        scanTBBB();
      }
    }
    if(menu==1)
    {
      lcd_ziky.setCursor(0,0);
      lcd_ziky.print(" Scan Berat & Tinggi");
      lcd_ziky.setCursor(0,1);
      lcd_ziky.write(1);
      lcd_ziky.setCursor(1,1);
      lcd_ziky.print("Power Tungkai");
      lcd_ziky.setCursor(0,2);
      lcd_ziky.print(" W Body Reaction  ");
      lcd_ziky.setCursor(0,3);
      lcd_ziky.print(" Power Endurance  ");
      if(button_BACK)
      {
        mPower++;
        if(mPower>2)mPower=1;
        delay(130);
      }
      if(mPower==1)
      {
        lcd_ziky.setCursor(14,1);

```

```

    lcd_ziky.print("[MANU]");
    EEPROM.write(addr_mPower, 1);
    if(button_ok)
    {
        lcd_ziky.clear();
        delay(300);
        power_tungkai();
    }
}
if(mPower==2)
{
    lcd_ziky.setCursor(14,1);
    lcd_ziky.print("[AUTO]");
    EEPROM.write(addr_mPower, 2);
    if(button_ok)
    {
        lcd_ziky.clear();
        delay(300);
        power_tungkai_mode_otomatis();
    }
}
}
if(menu==2)
{
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print(" Scan Berat & Tinggi");
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print(" Power Tungkai");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.write(1);
    lcd_ziky.setCursor(1,2);
    lcd_ziky.print("W Body Reaction  ");
    lcd_ziky.setCursor(0,3);
    lcd_ziky.print(" Power Endurance  ");
    if(read_eeprom_mPower==1)
    {
        lcd_ziky.setCursor(14,1);
        lcd_ziky.print("[MANU]");
    }
    if(read_eeprom_mPower==2)
    {
        lcd_ziky.setCursor(14,1);
        lcd_ziky.print("[AUTO]");
    }
    if(button_ok)
    {
        lcd_ziky.clear();
        delay(300);
        jenis_kelamin_manual();
    }
}
if(menu==3)
{
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print(" Scan Berat & Tinggi");
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print(" Power Tungkai");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print(" W Body Reaction  ");
    lcd_ziky.setCursor(0,3);
    lcd_ziky.write(1);
    lcd_ziky.setCursor(1,3);
    lcd_ziky.print("Power Endurance  ");
    if(read_eeprom_mPower==1)
    {
        lcd_ziky.setCursor(14,1);
        lcd_ziky.print("[MANU]");
    }
    if(read_eeprom_mPower==2)
    {
        lcd_ziky.setCursor(14,1);
        lcd_ziky.print("[AUTO]");
    }
}

```

```

    }
    if(button_ok)
    {
        lcd_ziky.clear();
        delay(200);
        power_endurance();
    }
}
if(menu==4)
{
    lcd_ziky.setCursor(0,0);
    lcd_ziky.write(1);
    lcd_ziky.print("Turunkan Tiang");
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print(" Lihat Data kelompok");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print(" Informasi microSD");
    lcd_ziky.setCursor(0,3);
    lcd_ziky.print(" Pengaturan microSD");
    if(button_ok)
    {
        lcd_ziky.clear();
        delay(200);
        while(1)
        {
            lcd_ziky.setCursor(0,0);
            lcd_ziky.print("Tiang Turun...");
            motor_turun();
            val_bawah=digitalRead(limit_bawah);
            if(val_bawah==LOW)
            {
                digitalWrite(buzzer,HIGH);
                motor_stop();
                lcd_ziky.clear();
                lcd_ziky.setCursor(0,0);
                lcd_ziky.print("Selesai...");
                encoderValue=0;
                delay(300);
                digitalWrite(buzzer,LOW);
                lcd_ziky.clear();
                goto atas;
            }
        }
    }
}
if(menu==5)
{
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print(" Turunkan Tiang");
    lcd_ziky.setCursor(0,1);
    lcd_ziky.write(1);
    lcd_ziky.print("Lihat Data kelompok");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print(" Informasi microSD");
    lcd_ziky.setCursor(0,3);
    lcd_ziky.print(" Pengaturan microSD");
    if(button_ok)
    {
        lcd_ziky.clear();
        delay(200);
        while(1)
        {
            kalibrasi_button();
            lcd_ziky.setCursor(0,0);
            lcd_ziky.print("data:kelompok");
            lcd_ziky.print(data_kelompok);
            lcd_ziky.setCursor(0,1);
            lcd_ziky.print(">Tekan<OK>utk lihat");
            if(button_up)
            {
                data_kelompok++;
                if(data_kelompok>5)data_kelompok=5;
            }
        }
    }
}

```

```

        delay(80);
    }
    if(button_down)
    {
        data_kelompok--;
        if(data_kelompok<1) data_kelompok=1;
        delay(80);
    }
    if(button_BACK)
    {
        lcd_ziky.clear();
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("Selesai...");
        delay(300);
        lcd_ziky.clear();
        break;
    }
    if(button_ok)
    {
        lcd_ziky.setCursor(0,2);
        lcd_ziky.print("Sending data...");
        lcd_ziky.setCursor(0,3);
        lcd_ziky.print("Lihat Serial Monitor");

if (data_kelompok==1) {grp1=SD.open("grp1.txt");delay(50);while
(grp1.available()) {Serial.write(grp1.read());}grp1.close();Serial.println();}
    else
if (data_kelompok==2) {grp2=SD.open("grp2.txt");delay(50);while
(grp2.available()) {Serial.write(grp2.read());}grp2.close();Serial.println();}
    else
if (data_kelompok==3) {grp3=SD.open("grp3.txt");delay(50);while
(grp3.available()) {Serial.write(grp3.read());}grp3.close();Serial.println();}
    else
if (data_kelompok==4) {grp4=SD.open("grp4.txt");delay(50);while
(grp4.available()) {Serial.write(grp4.read());}grp4.close();Serial.println();}
    else
if (data_kelompok==5) {grp5=SD.open("grp5.txt");delay(50);while
(grp5.available()) {Serial.write(grp5.read());}grp5.close();Serial.println();}
    /*else
if (data_kelompok==6) {grp6=SD.open("grp6.txt");delay(50);while
(grp6.available()) {Serial.write(grp6.read());}grp6.close();Serial.println();}
    else
if (data_kelompok==7) {grp7=SD.open("grp7.txt");delay(50);while
(grp7.available()) {Serial.write(grp7.read());}grp7.close();Serial.println();}
    else
if (data_kelompok==8) {grp8=SD.open("grp8.txt");delay(50);while
(grp8.available()) {Serial.write(grp8.read());}grp8.close();Serial.println();}
    else
if (data_kelompok==9) {grp9=SD.open("grp9.txt");delay(50);while
(grp9.available()) {Serial.write(grp9.read());}grp9.close();Serial.println();}
    else
if (data_kelompok==10) {grp10=SD.open("grp10.txt");delay(50);while
(grp10.available()) {Serial.write(grp10.read());}grp10.close();Serial.println();}
*/

        delay(1500);
        lcd_ziky.setCursor(0,2);
        lcd_ziky.print("                ");
        lcd_ziky.setCursor(0,3);
        lcd_ziky.print("                ");
    }
}
}
if(menu==6)
{
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print(" Turunkan Tiang");
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print(" Lihat Data kelompok");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.write(1);
    lcd_ziky.print("Informasi microSD");
}

```



```

lcd_ziky.setCursor(0,2);
lcd_ziky.print(" Informasi microSD");
lcd_ziky.setCursor(0,3);
lcd_ziky.write(1);
lcd_ziky.print("Pengaturan microSD");
if(button_ok)
{
  lcd_ziky.clear();
  delay(200);
  while(1)
  {
    kalibrasi_button();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("Hapus :kelompok");
    lcd_ziky.print(format);
    lcd_ziky.print(" <OK>");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print("format<SAVE><UP><OK>");
    if(button_up)
    {
      format++;
      if(format>5) format=5;
      delay(80);
    }
    if(button_down)
    {
      format--;
      if(format<1) format=1;
      delay(80);
    }
    if(button_BACK)
    {
      lcd_ziky.clear();
      lcd_ziky.setCursor(0,0);
      lcd_ziky.print("Selesai...");
      delay(300);
      lcd_ziky.clear();
      break;
    }
    if(button_SAVE&&button_up&&button_ok)
    {
      lcd_ziky.setCursor(0,3);
      lcd_ziky.print("formatting");

if(SD.exists("grp1.txt")) {SD.remove("grp1.txt");Serial.println("file grp1
dihapus");}delay(100);lcd_ziky.write(2);

if(SD.exists("grp2.txt")) {SD.remove("grp2.txt");Serial.println("file grp2
dihapus");}delay(100);lcd_ziky.write(2);

if(SD.exists("grp3.txt")) {SD.remove("grp3.txt");Serial.println("file grp3
dihapus");}delay(100);lcd_ziky.write(2);

if(SD.exists("grp4.txt")) {SD.remove("grp4.txt");Serial.println("file grp4
dihapus");}delay(100);lcd_ziky.write(2);

if(SD.exists("grp5.txt")) {SD.remove("grp5.txt");Serial.println("file grp5
dihapus");}delay(100);lcd_ziky.write(2);

/*if(SD.exists("grp6.txt")) {SD.remove("grp6.txt");Serial.println("file grp6
dihapus");}delay(100);lcd_ziky.write(2);

if(SD.exists("grp7.txt")) {SD.remove("grp7.txt");Serial.println("file grp7
dihapus");}delay(100);lcd_ziky.write(2);

if(SD.exists("grp8.txt")) {SD.remove("grp8.txt");Serial.println("file grp8
dihapus");}delay(100);lcd_ziky.write(2);

if(SD.exists("grp9.txt")) {SD.remove("grp9.txt");Serial.println("file grp9
dihapus");}delay(100);lcd_ziky.write(2);

```

```

if(SD.exists("grp10.txt")){SD.remove("grp10.txt");Serial.println("file grp10
dihapus");}delay(100);lcd_ziky.write(2);*/
    delay(1000);
    lcd_ziky.clear();
    goto atas;
}
else if(button_ok)
{
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print("deleting");

if(format==1){if(SD.exists("grp1.txt")){SD.remove("grp1.txt");Serial.println("fi
le grp1 dihapus");}delay(200);}
    else
if(format==2){if(SD.exists("grp2.txt")){SD.remove("grp2.txt");Serial.println("fi
le grp2 dihapus");}delay(200);}
    else
if(format==3){if(SD.exists("grp3.txt")){SD.remove("grp3.txt");Serial.println("fi
le grp3 dihapus");}delay(200);}
    else
if(format==4){if(SD.exists("grp4.txt")){SD.remove("grp4.txt");Serial.println("fi
le grp4 dihapus");}delay(200);}
    else
if(format==5){if(SD.exists("grp5.txt")){SD.remove("grp5.txt");Serial.println("fi
le grp5 dihapus");}delay(200);}
    /*else
if(format==6){if(SD.exists("grp6.txt")){SD.remove("grp6.txt");Serial.println("fi
le grp6 dihapus");}delay(200);}
    else
if(format==7){if(SD.exists("grp7.txt")){SD.remove("grp7.txt");Serial.println("fi
le grp7 dihapus");}delay(200);}
    else
if(format==8){if(SD.exists("grp8.txt")){SD.remove("grp8.txt");Serial.println("fi
le grp8 dihapus");}delay(200);}
    else
if(format==9){if(SD.exists("grp9.txt")){SD.remove("grp9.txt");Serial.println("fi
le grp9 dihapus");}delay(200);}
    else
if(format==10){if(SD.exists("grp10.txt")){SD.remove("grp10.txt");Serial.println(
"file grp10 dihapus");}delay(200);}*/
        for(int del=9;del<20;del++)
        {
            lcd_ziky.setCursor(del,1);
            lcd_ziky.write(2);
            delay(100);
        }
        lcd_ziky.setCursor(0,1);
        lcd_ziky.print("                ");
    }
}
}
if(button_SAVE)
{
    lcd_ziky.clear();
    delay(100);
    while(1)
    {
        kalibrasi_button();
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("Kelompok:");
        lcd_ziky.print(kelompok);
        lcd_ziky.print(" tekan<OK>");
        if(button_up)
        {
            kelompok++;
            if(kelompok>5)kelompok=5;
            delay(80);
        }
        if(button_down)
        {

```

```

kelompok--;
if(kelompok<1) kelompok=1;
delay(80);
}
if(button_BACK)
{
  lcd_ziky.clear();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Batal Simpan");
  delay(300);
  lcd_ziky.clear();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Selesai...");
  delay(300);
  lcd_ziky.clear();
  break;
}
if(button_ok)
{
  delay(50);
  while(1)
  {
    kalibrasi_button();
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print("User:");
    lcd_ziky.print(user);
    lcd_ziky.print(" tekan<SAVE>  ");
    if(button_up)
    {
      user++;
      if(user>200)user=1;
      delay(80);
    }
    if(button_down)
    {
      user--;
      if(user<1)user=200;
      delay(80);
    }
    if(button_BACK)
    {
      lcd_ziky.clear();
      lcd_ziky.setCursor(0,0);
      lcd_ziky.print("Batal Simpan");
      delay(300);
      lcd_ziky.clear();
      lcd_ziky.setCursor(0,0);
      lcd_ziky.print("Selesai...");
      delay(300);
      lcd_ziky.clear();
      goto akhir;
    }
    if(button_SAVE){goto lanjut;}
  }
  lanjut:
  while(1)
  {
    simpan_TB_koma();
    simpan_BB_koma();
    rumus_powerDankecepatan_VJ();
    rumus_powerDankecepatan_HJ();
    simpan_WBR_koma();
    read_eeprom_PE[1]=EEPROM.read(addr_PE1);
    read_eeprom_PE[2]=EEPROM.read(addr_PE2);
    read_eeprom_PE[3]=EEPROM.read(addr_PE3);
    read_eeprom_PE[4]=EEPROM.read(addr_PE4);
    read_eeprom_PE[5]=EEPROM.read(addr_PE5);
    read_eeprom_PE[6]=EEPROM.read(addr_PE6);
    read_eeprom_PE[7]=EEPROM.read(addr_PE7);
    read_eeprom_PE[8]=EEPROM.read(addr_PE8);
    read_eeprom_PE[9]=EEPROM.read(addr_PE9);
    read_eeprom_PE[10]=EEPROM.read(addr_PE10);
  }
}

```



```

grp2.println();
grp2.println();
grp2.println("=====");
grp2.println("VERTIKAL JUMP");
grp2.println("=====");
grp2.print("Tinggi Lompatan:");
grp2.println(s_T_LOM_VJ);
grp2.print("Waktu Lompatan:");
grp2.println(s_waktu_VJ);
grp2.print("Kecepatan Lompatan:");
grp2.println(s_kec_VJ);
grp2.print("Power Lompatan:");
grp2.println(s_power_VJ);
grp2.println();
grp2.println();
grp2.println("=====");
grp2.println("HORIZONTAL JUMP");
grp2.println("=====");
grp2.print("Panjang Lompatan:");
grp2.println(s_P_LOM_HJ);
grp2.print("Waktu Lompatan:");
grp2.println(s_waktu_HJ);
grp2.print("Kecepatan Lompatan:");
grp2.println(s_kec_HJ);
grp2.print("Power Lompatan:");
grp2.println(s_power_HJ);
grp2.println();
grp2.println();
grp2.println("=====");
grp2.println("WHOLE BODY REACTION");
grp2.println("=====");
grp2.print("Reaction:");
grp2.println(sWBR);
grp2.println();
grp2.println();
grp2.println("=====");
grp2.println("POWER ENDURANCE");
grp2.println("=====");
grp2.print("Tinggi Lompatan1:");
grp2.println(sPE1);
grp2.print("Tinggi Lompatan2:");
grp2.println(sPE2);
grp2.print("Tinggi Lompatan3:");
grp2.println(sPE3);
grp2.print("Tinggi Lompatan4:");
grp2.println(sPE4);
grp2.print("Tinggi Lompatan5:");
grp2.println(sPE5);
grp2.print("Tinggi Lompatan6:");
grp2.println(sPE6);
grp2.print("Tinggi Lompatan7:");
grp2.println(sPE7);
grp2.print("Tinggi Lompatan8:");
grp2.println(sPE8);
grp2.print("Tinggi Lompatan9:");
grp2.println(sPE9);
grp2.print("Tinggi Lompatan10:");
grp2.println(sPE10);
grp2.print("Rata-rata Lompatan:");
grp2.println(s_rata2PE);
grp2.print("Maksimal Lompatan:");
grp2.println(s_max_PE);
grp2.print("Minimal Lompatan:");
grp2.println(s_min_PE);
grp2.println();
grp2.println();
grp2.close();
delay(300);
goto sukses;
}
else
{

```



```

grp5.println("VERTIKAL JUMP");
grp5.println("=====");
grp5.print("Tinggi Lompatan:");
grp5.println(s_T_LOM_VJ);
grp5.print("Waktu Lompatan:");
grp5.println(s_waktu_VJ);
grp5.print("Kecepatan Lompatan:");
grp5.println(s_kec_VJ);
grp5.print("Power Lompatan:");
grp5.println(s_power_VJ);
grp5.println();
grp5.println();
grp5.println("=====");
grp5.println("HORIZONTAL JUMP");
grp5.println("=====");
grp5.print("Panjang Lompatan:");
grp5.println(s_P_LOM_HJ);
grp5.print("Waktu Lompatan:");
grp5.println(s_waktu_HJ);
grp5.print("Kecepatan Lompatan:");
grp5.println(s_kec_HJ);
grp5.print("Power Lompatan:");
grp5.println(s_power_HJ);
grp5.println();
grp5.println();
grp5.println("=====");
grp5.println("WHOLE BODY REACTION");
grp5.println("=====");
grp5.print("Reaction:");
grp5.println(sWBR);
grp5.println();
grp5.println();
grp5.println("=====");
grp5.println("POWER ENDURANCE");
grp5.println("=====");
grp5.print("Tinggi Lompatan1:");
grp5.println(sPE1);
grp5.print("Tinggi Lompatan2:");
grp5.println(sPE2);
grp5.print("Tinggi Lompatan3:");
grp5.println(sPE3);
grp5.print("Tinggi Lompatan4:");
grp5.println(sPE4);
grp5.print("Tinggi Lompatan5:");
grp5.println(sPE5);
grp5.print("Tinggi Lompatan6:");
grp5.println(sPE6);
grp5.print("Tinggi Lompatan7:");
grp5.println(sPE7);
grp5.print("Tinggi Lompatan8:");
grp5.println(sPE8);
grp5.print("Tinggi Lompatan9:");
grp5.println(sPE9);
grp5.print("Tinggi Lompatan10:");
grp5.println(sPE10);
grp5.print("Rata-rata Lompatan:");
grp5.println(s_rata2PE);
grp5.print("Maksimal Lompatan:");
grp5.println(s_max_PE);
grp5.print("Minimal Lompatan:");
grp5.println(s_min_PE);
grp5.println();
grp5.println();
grp5.close();
delay(300);
goto sukses;
}
else
{
    goto gagal;
}
}

```



```

    lcd_ziky.print("Kalibrasi");
    for(int x=0;x<20;x++)
    {
        lcd_ziky.setCursor(x,1);
        lcd_ziky.write(2);
        lcd_ziky.display();
        Serial1.println(3);
        delay(100);
    }
    for(int z=0;z<30;z++)
    {
        Serial1.println(4);
        delay(10);
    }
    lcd_ziky.clear();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("Selesai...");
    delay(200);
    lcd_ziky.clear();
    break;
}
}
if(button_ok)
{
    lcd_ziky.clear();
    Serial1.println(1);
    delay(10);
    Serial1.println(9);
    goto eksekusi;
}
if(button_BACK)
{
    lcd_ziky.clear();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("Selesai...");
    delay(300);
    lcd_ziky.clear();
    goto akhir;
}
}
eksekusi:
while(1)
{
    kalibrasi_button();
    parsing_data_PE();
    rumus_nilai_terkecil_PE();
    rumus_nilai_terbesar_PE();
    read_eeprom_PE[1]=EEPROM.read(addr_PE1);
    read_eeprom_PE[2]=EEPROM.read(addr_PE2);
    read_eeprom_PE[3]=EEPROM.read(addr_PE3);
    read_eeprom_PE[4]=EEPROM.read(addr_PE4);
    read_eeprom_PE[5]=EEPROM.read(addr_PE5);
    read_eeprom_PE[6]=EEPROM.read(addr_PE6);
    read_eeprom_PE[7]=EEPROM.read(addr_PE7);
    read_eeprom_PE[8]=EEPROM.read(addr_PE8);
    read_eeprom_PE[9]=EEPROM.read(addr_PE9);
    read_eeprom_PE[10]=EEPROM.read(addr_PE10);
    read_eeprom_rata2_PE=EEPROM.read(addr_rata2_PE);
    read_eeprom_min_PE=EEPROM.read(addr_min_PE);
    read_eeprom_max_PE=EEPROM.read(addr_max_PE);

    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("1:");
    lcd_ziky.print(read_eeprom_PE[1]);
    lcd_ziky.write(8);
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print("2:");
    lcd_ziky.print(read_eeprom_PE[2]);
    lcd_ziky.write(8);
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print("3:");
    lcd_ziky.print(read_eeprom_PE[3]);
}

```

```

lcd_ziky.write(8);
lcd_ziky.setCursor(0,3);
lcd_ziky.print("4:");
lcd_ziky.print(read_eeprom_PE[4]);
lcd_ziky.write(8);
lcd_ziky.setCursor(5,0);
lcd_ziky.print("5:");
lcd_ziky.print(read_eeprom_PE[5]);
lcd_ziky.write(8);
lcd_ziky.setCursor(5,1);
lcd_ziky.print("6:");
lcd_ziky.print(read_eeprom_PE[6]);
lcd_ziky.write(8);
lcd_ziky.setCursor(5,2);
lcd_ziky.print("7:");
lcd_ziky.print(read_eeprom_PE[7]);
lcd_ziky.write(8);
lcd_ziky.setCursor(5,3);
lcd_ziky.print("8:");
lcd_ziky.print(read_eeprom_PE[8]);
lcd_ziky.write(8);
lcd_ziky.setCursor(10,0);
lcd_ziky.print("9 :");
lcd_ziky.print(read_eeprom_PE[9]);
lcd_ziky.setCursor(10,1);
lcd_ziky.print("10:");
lcd_ziky.print(read_eeprom_PE[10]);

lcd_ziky.setCursor(16,0);
lcd_ziky.write(7);
lcd_ziky.print(read_eeprom_rata2_PE);
lcd_ziky.print(" ");
lcd_ziky.setCursor(16,1);
lcd_ziky.write(6);
lcd_ziky.print(read_eeprom_min_PE);
lcd_ziky.setCursor(16,2);
lcd_ziky.write(5);
lcd_ziky.print(read_eeprom_max_PE);
lcd_ziky.setCursor(16,3);
lcd_ziky.write(2);
lcd_ziky.setCursor(17,3);
lcd_ziky.print("Cm");
if(button_BACK)
{
  lcd_ziky.clear();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Kirim data ke web...");
  paket_data_nodeMcu();
  lcd_ziky.clear();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Selesai...");
  rata2tinggi_lompat=0;
  for(int reset_data_PE=0;reset_data_PE<10;reset_data_PE++)
  {
    dt1[reset_data_PE]="";
    delay(20);
  }
  lcd_ziky.clear();
  goto mulai_lagi;
}
}
akhir:
while(1){break;}
}

void power_tungkai()
{
  while(1)
  {
    kalibrasi_button();
    zios_servo.write(115);
    lcd_ziky.setCursor(0,0);

```

```

lcd_ziky.print("POWER TUNGKAI [MANU]");
lcd_ziky.setCursor(0,3);
lcd_ziky.print("Tekan <BACK> utk bck");
if(button_BACK)
{
  lcd_ziky.clear();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Selesai...");
  zios_servo.write(95);
  delay(300);
  lcd_ziky.clear();
  break;
}
if(button_down)
{
  menu_PT++;
  if(menu_PT>1)menu_PT=0;
  delay(120);
}
if(button_up)
{
  menu_PT--;
  if(menu_PT<0)menu_PT=2;
  delay(120);
}
if(menu_PT==0)
{
  lcd_ziky.setCursor(0,1);
  lcd_ziky.write(1);
  lcd_ziky.print("Vertikal Jump");
  lcd_ziky.setCursor(0,2);
  lcd_ziky.print(" Horizontal Jump");
  if(button_ok)
  {
    lcd_ziky.clear();
    delay(300);
    jenis_kelamin_khusus_VJHJ();
  }
}
if(menu_PT==1)
{
  lcd_ziky.setCursor(0,1);
  lcd_ziky.print(" Vertikal Jump");
  lcd_ziky.setCursor(0,2);
  lcd_ziky.write(1);
  lcd_ziky.print("Horizontal Jump");
  if(button_ok)
  {
    lcd_ziky.clear();
    delay(300);
    jenis_kelamin_khusus_VJHJ();
  }
}
}
}

void power_tungkai_mode_otomatis()
{
  while(1)
  {
    kalibrasi_button();
    zios_servo.write(70);
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("POWER TUNGKAI [AUTO]");
    lcd_ziky.setCursor(0,3);
    lcd_ziky.print("Tekan <BACK> utk bck");
    if(button_BACK)
    {
      lcd_ziky.clear();
      lcd_ziky.setCursor(0,0);
      lcd_ziky.print("Selesai...");
      zios_servo.write(95);
    }
  }
}

```

```

        delay(300);
        lcd_ziky.clear();
        break;
    }
    if(button_down)
    {
        menu_PT++;
        if(menu_PT>1)menu_PT=0;
        delay(120);
    }
    if(button_up)
    {
        menu_PT--;
        if(menu_PT<0)menu_PT=2;
        delay(120);
    }
    if(menu_PT==0)
    {
        lcd_ziky.setCursor(0,1);
        lcd_ziky.write(1);
        lcd_ziky.print("Vertikal Jump");
        lcd_ziky.setCursor(0,2);
        lcd_ziky.print(" Horizontal Jump");
        if(button_ok)
        {
            lcd_ziky.clear();
            delay(300);
            jenis_kelamin_khusus_VJHJ();
        }
    }
    if(menu_PT==1)
    {
        lcd_ziky.setCursor(0,1);
        lcd_ziky.print(" Vertikal Jump");
        lcd_ziky.setCursor(0,2);
        lcd_ziky.write(1);
        lcd_ziky.print("Horizontal Jump");
        if(button_ok)
        {
            lcd_ziky.clear();
            delay(300);
            jenis_kelamin_khusus_VJHJ();
        }
    }
}

void simpan_TB_koma()
{
    read_eeprom_TB=EEPROM.read(addr_TB);
    read_eeprom_TB_belakang_koma=EEPROM.read(addr_TB_belakang_koma);
    TB_dijumlah=(read_eeprom_TB*10)+read_eeprom_TB_belakang_koma;
    TB_fix=TB_dijumlah/10;
}

void simpan_BB_koma()
{
    read_eeprom_BB=EEPROM.read(addr_BB);
    read_eeprom_BB_belakang_koma=EEPROM.read(addr_BB_belakang_koma);
    BB_dijumlah=(read_eeprom_BB*10)+read_eeprom_BB_belakang_koma;
    BB_fix=BB_dijumlah/10;
}

void scanTBBB()
{
    while(1)
    {
        simpan_TB_koma();
        simpan_BB_koma();
        lcd_ziky.setCursor(2, 0);
        lcd_ziky.print("SCANNING TB & BB");
        lcd_ziky.setCursor(0,1);
        lcd_ziky.print("Tekan <OK> utk mulai");
        lcd_ziky.setCursor(0, 2);
    }
}

```

```

lcd_ziky.print("Tinggi: ");
lcd_ziky.print(TB_fix);
lcd_ziky.print(" Cm");
lcd_ziky.setCursor(0, 3);
lcd_ziky.print("Berat : ");
lcd_ziky.print(BB_fix);
lcd_ziky.print(" Kg");
kalibrasi_button();
zios_servo.write(70);
if(button_BACK)
{
  lcd_ziky.clear();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Selesai...");
  zios_servo.write(90);
  delay(500);
  lcd_ziky.clear();
  break;
}
if(button_ok)
{
  lcd_ziky.clear();
  encoderValue=0;
  delay(100);
  while(1)
  {
    motor_turun();
    TB_enc=tiang1-encoderValue*0.0558;
    pembagi_TB=TB_enc;
    hasil_bagi_TB=TB_enc-pembagi_TB;
    hasil_belakang_koma_TB=hasil_bagi_TB*10;
    lcd_ziky.setCursor(0, 0);
    lcd_ziky.print("Scan.....");
    lcd_ziky.setCursor(0, 1);
    lcd_ziky.print("Tinggi :");
    lcd_ziky.print(TB_enc,1);
    Serial.println(TB_enc,1);
    lcd_ziky.setCursor(0, 2);
    lcd_ziky.print("Berat :0.0");
    val_kepala=digitalRead(limit_kepala);
    if(val_kepala==LOW)
    {
      digitalWrite(buzzer,HIGH);
      EEPROM.write(addr_TB, TB_enc);
      EEPROM.write(addr_TB_belakang_koma, hasil_belakang_koma_TB);
      delay(50);
      digitalWrite(buzzer,LOW);
      motor_naik();
      delay(700);
      goto berat_scan;
    }
    val_bawah=digitalRead(limit_bawah);
    if(val_bawah==LOW)
    {
      digitalWrite(buzzer,HIGH);
      EEPROM.write(addr_TB, TB_enc);
      EEPROM.write(addr_TB_belakang_koma, hasil_belakang_koma_TB);
      motor_stop();
      delay(200);
      digitalWrite(buzzer,LOW);
      goto pengukuran_bawah;
    }
  }
}
berat_scan:
while(1)
{
  motor_stop();
  timer();
  sw_millis.start();
  sw_micros.start();
  sw_secs.start();
  berat_badan();
}

```

```

simpan_TB_koma();
lcd_ziky.setCursor(0, 0);
lcd_ziky.print("Scan.....");
lcd_ziky.setCursor(0, 1);
lcd_ziky.print("Tinggi :");
lcd_ziky.print(TB_fix);
lcd_ziky.print(" ");
lcd_ziky.setCursor(0, 2);
lcd_ziky.print("Berat :");
lcd_ziky.print(timbangan,1);
lcd_ziky.print(" ");
if(teta_waktu>5)
{
  sw_millis.stop();
  sw_micros.stop();
  sw_secs.stop();
  pembagi_BB=timbangan;
  hasil_bagi_BB=timbangan-pembagi_BB;
  hasil_belakang_koma_BB=hasil_bagi_BB*10;
  EEPROM.write(addr_BB, timbangan);
  EEPROM.write(addr_BB_belakang_koma, hasil_belakang_koma_BB);
  for(int a=0; a<5;a++)
  {
    lcd_ziky.setCursor(0, 0);
    lcd_ziky.print("Selesai...");
    lcd_ziky.setCursor(0, 1);
    lcd_ziky.print("Tinggi :");
    lcd_ziky.print(TB_fix);
    lcd_ziky.print("Cm");
    lcd_ziky.setCursor(0, 2);
    lcd_ziky.print("Berat :");
    lcd_ziky.print(timbangan,1);
    lcd_ziky.print("Kg");
    digitalWrite(buzzer,HIGH);
    delay(200);
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print(" ");
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print(" ");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print(" ");
    digitalWrite(buzzer,LOW);
    delay(100);
  }
  lcd_ziky.clear();
  goto selesai;
}
}
pengukuran_bawah:
while(1)
{
  sw_millis.start();
  sw_micros.start();
  sw_secs.start();
  tinggi_badan();
  timer();
  berat_badan();
  tinggi_badan();
  //Simpan TB dengan koma
  pembagi_TB=TB;
  hasil_bagi_TB=TB-pembagi_TB;
  hasil_belakang_koma_TB=hasil_bagi_TB*10;
  //Simpan BB dengan koma
  pembagi_BB=timbangan;
  hasil_bagi_BB=timbangan-pembagi_BB;
  hasil_belakang_koma_BB=hasil_bagi_BB*10;
  lcd_ziky.setCursor(0, 0);
  lcd_ziky.print("Scan.....");
  lcd_ziky.setCursor(0, 1);
  lcd_ziky.print("Tinggi :");
  lcd_ziky.print(TB);
  lcd_ziky.print(" ");
}

```

```

lcd_ziky.setCursor(0, 2);
lcd_ziky.print("Berat :");
lcd_ziky.print(timbangan,1);
lcd_ziky.print(" ");
val_kepala=digitalRead(limit_kepala);
if(val_kepala==LOW)
{
  goto selesai;
}
if(teta_waktu>5)
{
  sw_millis.stop();
  sw_micros.stop();
  sw_secs.stop();
  EEPROM.write(addr_BB_belakang_koma, hasil_belakang_koma_BB);
  EEPROM.write(addr_TB, TB);
  EEPROM.write(addr_TB_belakang_koma, hasil_belakang_koma_TB);
  for(int a=0; a<5;a++)
  {
    lcd_ziky.setCursor(0, 0);
    lcd_ziky.print("Selesai...");
    lcd_ziky.setCursor(0, 1);
    lcd_ziky.print("Tinggi :");
    lcd_ziky.print(TB);
    lcd_ziky.print("Cm");
    lcd_ziky.setCursor(0, 2);
    lcd_ziky.print("Berat :");
    lcd_ziky.print(timbangan,1);
    lcd_ziky.print("Kg");
    digitalWrite(buzzer,HIGH);
    delay(200);
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("                ");
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print("                ");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print("                ");
    digitalWrite(buzzer,LOW);
    delay(100);
  }
  goto selesai;
}
}
selesai:
while(1)
{
  simpan_TB_koma();
  simpan_BB_koma();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Simpan...");
  motor_naik();
  Serial.print("TB=");
  Serial.println(TB_fix);
  Serial.print("BB=");
  Serial.println(BB_fix);
  val_atas=digitalRead(limit_atas);
  if(val_atas==LOW)
  {
    digitalWrite(buzzer,HIGH);
    motor_stop();
    sw_millis.reset();
    sw_micros.reset();
    sw_secs.reset();
    encoderValue=0;
    zios_servo.write(90);
    lcd_ziky.clear();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("Kirim data ke web...");
    paket_data_nodeMcu();
    delay(50);
    digitalWrite(buzzer,LOW);
  }
}

```

```

        lcd_ziky.clear();
        break;
    }
}
}
}

void simpan_WBR_koma()
{
    read_eeprom_WBR=EEPROM.read(addr_WBR);
    read_eeprom_WBR_belakang_koma=EEPROM.read(addr_WBR_belakang_koma);
    WBR_dijumlah=(read_eeprom_WBR*100)+read_eeprom_WBR_belakang_koma;
    WBR_fix=WBR_dijumlah/100;
}

void WBR_setting()
{
    while(1)
    {
        kalibrasi_button();
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("JENIS PENGUKURAN WBR");
        if(button_BACK)
        {
            lcd_ziky.clear();
            lcd_ziky.setCursor(0,0);
            lcd_ziky.print("Selesai...");
            delay(300);
            lcd_ziky.clear();
            break;
        }
        if(button_down)
        {
            jenis_pengukuran_WBR++;
            if(jenis_pengukuran_WBR>2) jenis_pengukuran_WBR=1;
            delay(140);
        }
        if(button_up)
        {
            jenis_pengukuran_WBR--;
            if(jenis_pengukuran_WBR<1) jenis_pengukuran_WBR=2;
            delay(140);
        }
        if(jenis_pengukuran_WBR==1)
        {
            lcd_ziky.setCursor(0,2);
            lcd_ziky.print("[Audio] <<");
            lcd_ziky.setCursor(10,2);
            lcd_ziky.print(" Visual ");
            if(button_ok)
            {
                lcd_ziky.clear();
                delay(300);
                encoderValue=0;
                whole_body_reaction();
            }
        }
        if(jenis_pengukuran_WBR==2)
        {
            lcd_ziky.setCursor(0,2);
            lcd_ziky.print(" Audio ");
            lcd_ziky.setCursor(8,2);
            lcd_ziky.print(">> [Visual]");
            if(button_ok)
            {
                lcd_ziky.clear();
                delay(300);
                encoderValue=0;
                whole_body_reaction();
            }
        }
    }
}

```

```

}
}

void whole_body_reaction()
{
  atas:
  while(1)
  {
    simpan_TB_koma();
    kalibrasi_button();
    motor_turun();
    TB_enc=tiang1-encoderValue*0.0540;
    if(TB_enc<TB_fix){motor_stop();}
    lcd_ziky.setCursor(0,3);
    lcd_ziky.print("Tekan <BACK> utk bck");
    if(jenis_kelamin==1)
    {
      lcd_ziky.setCursor(0,0);
      lcd_ziky.print("W BODY REACTION [L]");
    }
    else if(jenis_kelamin==2)
    {
      lcd_ziky.setCursor(0,0);
      lcd_ziky.print("W BODY REACTION [P]");
    }
    if(tumpuan==LOW)
    {
      lcd_ziky.setCursor(0,1);
      lcd_ziky.print("      SUDAH SIAP      ");
      lcd_ziky.setCursor(0,2);
      lcd_ziky.print("Tekan <OK> utk mulai");
      if(button_ok)
      {
        lcd_ziky.clear();
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("_____");
        lcd_ziky.setCursor(0,1);
        lcd_ziky.print("      FOKUS...!!!      ");
        for(int a=0;a<20;a++)
        {
          lcd_ziky.setCursor(a,2);
          lcd_ziky.write(3);
        }
        delay(5000);
        lcd_ziky.clear();
        if(jenis_pengukuran_WBR==1){digitalWrite(buzzer_WBR,LOW);}
        else if(jenis_pengukuran_WBR==2){digitalWrite(visual_WBR,LOW);}
        goto eksekusi;
      }
    }
    if(button_BACK)
    {
      lcd_ziky.clear();
      goto kembali;
    }
    else if(tumpuan==HIGH)
    {
      lcd_ziky.setCursor(0,1);
      lcd_ziky.print("      BELUM SIAP      ");
      lcd_ziky.setCursor(0,2);
      lcd_ziky.print("Ambil Ancang Ancang ");
    }
  }
  kembali:
  while(1)
  {
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("Kembali...");
    motor_naik();
    val_atas=digitalRead(limit_atas);
    if(val_atas==LOW)
    {

```

```

        digitalWrite(buzzer,HIGH);
        motor_stop();
        lcd_ziky.clear();
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("Selesai...");
        encoderValue=0;
        zios_servo.write(95);
        sw_millis.reset();
        sw_micros.reset();
        sw_secs.reset();
        delay(100);
        digitalWrite(buzzer,LOW);
        lcd_ziky.clear();
        goto akhir;
    }
}
eksekusi:
while(1)
{
    sw_millis.start();
    sw_micros.start();
    sw_secs.start();
    kalibrasi_button();
    rumus_powerDankecepatan_WBR();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("WHOLE BODY REACTION");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print("Reaction:");
    lcd_ziky.setCursor(9,2);
    lcd_ziky.print(teta_waktu,3);
    lcd_ziky.print(" detik");
    if(teta_waktu>0.300)
    {
        if(jenis_pengukuran_WBR==1){digitalWrite(buzzer_WBR,HIGH);}
        else if(jenis_pengukuran_WBR==2){digitalWrite(visual_WBR,HIGH);}
    }
    if(tumpuan==HIGH)
    {
        sw_millis.stop();
        sw_micros.stop();
        sw_secs.stop();
        if(jenis_pengukuran_WBR==1){digitalWrite(buzzer_WBR,HIGH);}
        else if(jenis_pengukuran_WBR==2){digitalWrite(visual_WBR,HIGH);}
        pembagi_WBR=teta_waktu;
        hasil_bagi_WBR=teta_waktu-pembagi_WBR;
        hasil_belakang_koma_WBR=hasil_bagi_WBR*100;
        EEPROM.write(addr_WBR, teta_waktu);
        EEPROM.write(addr_WBR_belakang_koma, hasil_belakang_koma_WBR);
        delay(150);
        if(teta_waktu>0)teta_waktu=0;
        lcd_ziky.setCursor(8,3);
        lcd_ziky.print("Stop");
        delay(150);
        lcd_ziky.clear();
        goto tampil_WBR;
    }
}
tampil_WBR:
while(1)
{
    kalibrasi_button();
    simpan_WBR_koma();
    rumus_powerDankecepatan_WBR();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("W.BODY REACTION");
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print("  Reaction:");
    lcd_ziky.setCursor(2,2);
    lcd_ziky.print(teta_waktu,3);
    lcd_ziky.print(" detik");
    lcd_ziky.setCursor(16,0);
    lcd_ziky.print("BACK");
}

```

```

lcd_ziky.setCursor(16,1);
lcd_ziky.print("PRES");
lcd_ziky.setCursor(16,2);
lcd_ziky.write(6);
lcd_ziky.write(6);
lcd_ziky.write(6);
lcd_ziky.write(6);
lcd_ziky.setCursor(16,3);
lcd_ziky.print("BACK");

if(jenis_kelamin==1)
{
  if(teta_waktu<=0.100){lcd_ziky.setCursor(0,3);lcd_ziky.print("
(ISTIMEWA  ");}
  else if(teta_waktu>=0.101 &&
teta_waktu<=0.200){lcd_ziky.setCursor(0,3);lcd_ziky.print(" (BAGUS SEKALI) ");}
  else if(teta_waktu>=0.201 &&
teta_waktu<=0.300){lcd_ziky.setCursor(0,3);lcd_ziky.print("      (BAGUS)      ");}
  else if(teta_waktu>=0.301 &&
teta_waktu<=0.400){lcd_ziky.setCursor(0,3);lcd_ziky.print("      (SEDANG)      ");}
  else if(teta_waktu>=0.401 &&
teta_waktu<=0.500){lcd_ziky.setCursor(0,3);lcd_ziky.print("      (KURANG)      ");}
  else if(teta_waktu>0.501){lcd_ziky.setCursor(0,3);lcd_ziky.print(" (KURANG
SEKALI) ");}
}

if(jenis_kelamin==2)
{
  if(teta_waktu<=0.100){lcd_ziky.setCursor(0,3);lcd_ziky.print("
(ISTIMEWA  ");}
  else if(teta_waktu>=0.101 &&
teta_waktu<=0.200){lcd_ziky.setCursor(0,3);lcd_ziky.print(" (BAGUS SEKALI) ");}
  else if(teta_waktu>=0.201 &&
teta_waktu<=0.300){lcd_ziky.setCursor(0,3);lcd_ziky.print("      (BAGUS)      ");}
  else if(teta_waktu>=0.301 &&
teta_waktu<=0.400){lcd_ziky.setCursor(0,3);lcd_ziky.print("      (SEDANG)      ");}
  else if(teta_waktu>=0.401 &&
teta_waktu<=0.500){lcd_ziky.setCursor(0,3);lcd_ziky.print("      (KURANG)      ");}
  else if(teta_waktu>0.501){lcd_ziky.setCursor(0,3);lcd_ziky.print(" (KURANG
SEKALI) ");}
}

if(button_BACK)
{
  lcd_ziky.clear();
  goto simpan_keluar;
}
}
simpan_keluar:
while(1)
{
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Kirim data ke web...");
  motor_naik();
  val_atas=digitalRead(limit_atas);
  if(val_atas==LOW)
  {
    digitalWrite(buzzer,HIGH);
    motor_stop();
    paket_data_nodeMcu();
    lcd_ziky.clear();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("Selesai...");
    encoderValue=0;
    zios_servo.write(95);
    sw_millis.reset();
    sw_micros.reset();
    sw_secs.reset();
    delay(100);
    digitalWrite(buzzer,LOW);
    lcd_ziky.clear();
    zios_servo.write(90);
  }
}

```

```

        break;
    }
}
akhir:
while(1){zios_servo.write(90);break;}
}

void bersih2_layar()
{
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("                ");
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print("                ");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print("                ");
    lcd_ziky.setCursor(0,3);
    lcd_ziky.print("                ");
    delay(100);
}

void tampilan_stanBy()
{
    lcd_ziky.setCursor(2,0);
    lcd_ziky.write(2);
    lcd_ziky.write(2);
    lcd_ziky.setCursor(2,1);
    lcd_ziky.print("UP");
    lcd_ziky.setCursor(9,0);
    lcd_ziky.write(2);
    lcd_ziky.write(2);
    lcd_ziky.setCursor(6,1);
    lcd_ziky.print("OK/MENU");
    lcd_ziky.setCursor(16,0);
    lcd_ziky.write(2);
    lcd_ziky.write(2);
    lcd_ziky.setCursor(15,1);
    lcd_ziky.print("SAVE");
    lcd_ziky.setCursor(2,2);
    lcd_ziky.write(2);
    lcd_ziky.write(2);
    lcd_ziky.setCursor(1,3);
    lcd_ziky.print("DOWN");
    lcd_ziky.setCursor(9,2);
    lcd_ziky.write(2);
    lcd_ziky.write(2);
    lcd_ziky.setCursor(8,3);
    lcd_ziky.print("RESET");
    lcd_ziky.setCursor(16,2);
    lcd_ziky.write(2);
    lcd_ziky.write(2);
    lcd_ziky.setCursor(15,3);
    lcd_ziky.print("BACK");
}

void silahkanTunggu()
{
    int i;
    for (i = 0 ; i < 1; i ++ )
    {
        bersih2_layar();
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("                ");
        lcd_ziky.setCursor(0,1);
        lcd_ziky.print("        ZIOS V.57");
        for(int a=0;a<20;a++)
        {
            lcd_ziky.setCursor(a,2);
            lcd_ziky.write(3);
        }
        for(int b=0;b<20;b++)
        {
            lcd_ziky.setCursor(b,3);

```

```

        lcd_ziky.write(2);
        delay(80);
    }
}
lcd_ziky.clear();
}

void kalibrasi_PE()
{
    int i;
    for (i = 0 ; i < 1; i ++ )
    {
        bersih2_layar();
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("Kalibrasi");
        for(int c=0;c<20;c++)
        {
            lcd_ziky.setCursor(c,1);
            lcd_ziky.write(3);
        }
        for(int d=0;d<20;d++)
        {
            lcd_ziky.setCursor(0,3);
            lcd_ziky.write(2);
            delay(80);
        }
    }
    lcd_ziky.clear();
}

void simpan_data()
{
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print("Simpan");
    for(int d=0;d<20;d++)
    {
        lcd_ziky.setCursor(d,3);
        lcd_ziky.write(2);
        delay(50);
    }
}

void kalibrasi_button()
{
    bDown =map(analogRead(A8),0,1023,0,1023);
    bUp =map(analogRead(A9),0,1023,0,1023);
    bOk =map(analogRead(A10),0,1023,0,1023);
    bSAVE =map(analogRead(A11),0,1023,0,1023);
    bBACK =map(analogRead(A12),0,1023,0,1023);
    tumpuan=digitalRead(A14);
}

void istimewa() { lcd_ziky.setCursor(0,1);lcd_ziky.print(" (ISTIMEWA)
");}
void baik_sekali() { lcd_ziky.setCursor(0,1);lcd_ziky.print(" (BAIK
SEKALI) ");}
void baik() { lcd_ziky.setCursor(0,1);lcd_ziky.print(" (BAIK)
");}
void cukup() { lcd_ziky.setCursor(0,1);lcd_ziky.print(" (CUKUP)
");}
void kurang() { lcd_ziky.setCursor(0,1);lcd_ziky.print(" (KURANG)
");}
void kurang_sekali() { lcd_ziky.setCursor(0,1);lcd_ziky.print(" (KURANG
SEKALI) ");}
void gagal() { lcd_ziky.setCursor(0,1);lcd_ziky.print(" (GAGAL)
");}

void horizontal_jump_manual()
{
    mulai_lagi:
    while(1)
    {

```

```

kalibrasi_button();
deteksi1();
deteksi2();
deteksi3();
if(jenis_kelamin==1)
{
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("HORIZONTAL JUMP [L]");
}
else if(jenis_kelamin==2)
{
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("HORIZONTAL JUMP [P]");
}
if(jarakD1<=70&&jarakD2<=70&&jarakD3<=70)
{
  lcd_ziky.setCursor(0,1);
  lcd_ziky.print("      SUDAH SIAP      ");
  lcd_ziky.setCursor(0,2);
  lcd_ziky.print("Tekan <OK> utk mulai");
  lcd_ziky.setCursor(0,3);
  lcd_ziky.print("Tekan <BACK> utk bck");
  if(button_ok)
  {
    lcd_ziky.clear();
    Serial2.println(1);
    delay(10);
    goto siapsiap;
  }
}
if(button_BACK)
{
  lcd_ziky.clear();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Selesai...");
  delay(300);
  lcd_ziky.clear();
  goto akhir;
}
else if(jarakD1>70&&jarakD2>70&&jarakD3>70)
{
  lcd_ziky.setCursor(0,1);
  lcd_ziky.print("      BELUM SIAP      ");
  lcd_ziky.setCursor(0,2);
  lcd_ziky.print("Ambil Ancang Ancang ");
  lcd_ziky.setCursor(0,3);
  lcd_ziky.print("Tekan <BACK> utk bck");
}
}
siapsiap:
while(1)
{
  deteksi1();
  deteksi2();
  deteksi3();
  if(jarakD1<=70&&jarakD2<=70&&jarakD3<=70)
  {
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("      ready      ");
  }
  else if(jarakD1>70&&jarakD2>70&&jarakD3>70)
  {
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("      GO!!!      ");
    lcd_ziky.clear();
    Serial2.println(7);
    goto eksekusi;
  }
}
eksekusi:
while(1)
{

```

```

kalibrasi_button();
rumus_powerDankecepatan_HJ();
lcd_ziky.setCursor(0,0);
lcd_ziky.print("P.lom :");
lcd_ziky.print(jarak_HJ_fix);
lcd_ziky.print(" Cm");
lcd_ziky.setCursor(0,1);
lcd_ziky.print("Kec  :");
lcd_ziky.print(kecepatan_fix_HJ);
lcd_ziky.print(" Cm/s");
lcd_ziky.setCursor(0,2);
lcd_ziky.print("Power :");
lcd_ziky.print(power_fix_HJ);
lcd_ziky.print(" Kgm/s");
lcd_ziky.setCursor(0,3);
lcd_ziky.print("      :");
lcd_ziky.print(horse_power_HJ);
lcd_ziky.print(" Hp");
if(button_BACK)
{
  lcd_ziky.clear();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Kirim data ke web...");
  paket_data_nodeMcu();
  lcd_ziky.clear();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Selesai...");
  waktu_fix_HJ=0;
  kecepatan_fix_HJ=0;
  for(int
reset_data_HJ_manual=0;reset_data_HJ_manual<10;reset_data_HJ_manual++)
  {
    dt3[reset_data_HJ_manual]="";
    delay(20);
  }
  lcd_ziky.clear();
  goto mulai_lagi;
}
delay(2000);
lcd_ziky.clear();
goto gantian1;
}

gantian1:
while(1)
{
  kalibrasi_button();
  rumus_powerDankecepatan_HJ();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("_____");
  for(int a=0;a<20;a++){lcd_ziky.setCursor(a,2);lcd_ziky.write(3);}
  if(jenis_kelamin==1)
  {
    if(umur==1)
    {
      if(jarak_HJ_fix>196){istimewa();}
      else if(jarak_HJ_fix>168 && jarak_HJ_fix<=196){baik_sekali();}
      else if(jarak_HJ_fix>155 && jarak_HJ_fix<=168){baik();}
      else if(jarak_HJ_fix>145 && jarak_HJ_fix<=155){cukup();}
      else if(jarak_HJ_fix>135 && jarak_HJ_fix<=145){kurang();}
      else if(jarak_HJ_fix>115 && jarak_HJ_fix<=135){kurang_sekali();}
      else if(jarak_HJ_fix<=115){gagal();}
    }
    if(umur==2)
    {
      if(jarak_HJ_fix>226){istimewa();}
      else if(jarak_HJ_fix>175 && jarak_HJ_fix<=226){baik_sekali();}
      else if(jarak_HJ_fix>168 && jarak_HJ_fix<=175){baik();}
      else if(jarak_HJ_fix>152 && jarak_HJ_fix<=168){cukup();}
      else if(jarak_HJ_fix>140 && jarak_HJ_fix<=152){kurang();}
      else if(jarak_HJ_fix>122 && jarak_HJ_fix<=140){kurang_sekali();}
      else if(jarak_HJ_fix<=122){gagal();}
    }
  }
}

```

```

}
if (umur==3)
{
    if(jarak_HJ_fix>256){istimewa();}
    else if(jarak_HJ_fix>183 && jarak_HJ_fix<=256){baik_sekali();}
    else if(jarak_HJ_fix>170 && jarak_HJ_fix<=183){baik();}
    else if(jarak_HJ_fix>160 && jarak_HJ_fix<=170){cukup();}
    else if(jarak_HJ_fix>147 && jarak_HJ_fix<=160){kurang();}
    else if(jarak_HJ_fix>127 && jarak_HJ_fix<=147){kurang_sekali();}
    else if(jarak_HJ_fix<=127){gagal();}
}
if (umur==4)
{
    if(jarak_HJ_fix>259){istimewa();}
    else if(jarak_HJ_fix>196 && jarak_HJ_fix<=259){baik_sekali();}
    else if(jarak_HJ_fix>173 && jarak_HJ_fix<=196){baik();}
    else if(jarak_HJ_fix>168 && jarak_HJ_fix<=173){cukup();}
    else if(jarak_HJ_fix>152 && jarak_HJ_fix<=168){kurang();}
    else if(jarak_HJ_fix>132 && jarak_HJ_fix<=152){kurang_sekali();}
    else if(jarak_HJ_fix<=132){gagal();}
}
if (umur==5)
{
    if(jarak_HJ_fix>274){istimewa();}
    else if(jarak_HJ_fix>211 && jarak_HJ_fix<=274){baik_sekali();}
    else if(jarak_HJ_fix>193 && jarak_HJ_fix<=211){baik();}
    else if(jarak_HJ_fix>180 && jarak_HJ_fix<=193){cukup();}
    else if(jarak_HJ_fix>163 && jarak_HJ_fix<=180){kurang();}
    else if(jarak_HJ_fix>142 && jarak_HJ_fix<=163){kurang_sekali();}
    else if(jarak_HJ_fix<=142){gagal();}
}
if (umur==6)
{
    if(jarak_HJ_fix>274){istimewa();}
    else if(jarak_HJ_fix>221 && jarak_HJ_fix<=274){baik_sekali();}
    else if(jarak_HJ_fix>208 && jarak_HJ_fix<=221){baik();}
    else if(jarak_HJ_fix>196 && jarak_HJ_fix<=208){cukup();}
    else if(jarak_HJ_fix>180 && jarak_HJ_fix<=196){kurang();}
    else if(jarak_HJ_fix>157 && jarak_HJ_fix<=180){kurang_sekali();}
    else if(jarak_HJ_fix<=157){gagal();}
}
if (umur==7)
{
    if(jarak_HJ_fix>279){istimewa();}
    else if(jarak_HJ_fix>229 && jarak_HJ_fix<=279){baik_sekali();}
    else if(jarak_HJ_fix>218 && jarak_HJ_fix<=229){baik();}
    else if(jarak_HJ_fix>205 && jarak_HJ_fix<=218){cukup();}
    else if(jarak_HJ_fix>193 && jarak_HJ_fix<=205){kurang();}
    else if(jarak_HJ_fix>160 && jarak_HJ_fix<=193){kurang_sekali();}
    else if(jarak_HJ_fix<=160){gagal();}
}
if (umur==8)
{
    if(jarak_HJ_fix>300){istimewa();}
    else if(jarak_HJ_fix>239 && jarak_HJ_fix<=300){baik_sekali();}
    else if(jarak_HJ_fix>226 && jarak_HJ_fix<=239){baik();}
    else if(jarak_HJ_fix>213 && jarak_HJ_fix<=226){cukup();}
    else if(jarak_HJ_fix>193 && jarak_HJ_fix<=213){kurang();}
    else if(jarak_HJ_fix>165 && jarak_HJ_fix<=193){kurang_sekali();}
    else if(jarak_HJ_fix<=165){gagal();}
}
}

if(jenis_kelamin==2)
{
    if (umur==1)
    {
        if(jarak_HJ_fix>241){istimewa();}
        else if(jarak_HJ_fix>158 && jarak_HJ_fix<=241){baik_sekali();}
        else if(jarak_HJ_fix>147 && jarak_HJ_fix<=158){baik();}
        else if(jarak_HJ_fix>137 && jarak_HJ_fix<=147){cukup();}
        else if(jarak_HJ_fix>122 && jarak_HJ_fix<=137){kurang();}
    }
}

```

```

else if(jarak_HJ_fix>104 && jarak_HJ_fix<=122){kurang_sekali();}
else if(jarak_HJ_fix<=104){gagal();}
}
if(umur==2)
{
if(jarak_HJ_fix>213){istimewa();}
else if(jarak_HJ_fix>165 && jarak_HJ_fix<=213){baik_sekali();}
else if(jarak_HJ_fix>155 && jarak_HJ_fix<=165){baik();}
else if(jarak_HJ_fix>142 && jarak_HJ_fix<=155){cukup();}
else if(jarak_HJ_fix>130 && jarak_HJ_fix<=142){kurang();}
else if(jarak_HJ_fix>112 && jarak_HJ_fix<=130){kurang_sekali();}
else if(jarak_HJ_fix<=112){gagal();}
}
if(umur==3)
{
if(jarak_HJ_fix>256){istimewa();}
else if(jarak_HJ_fix>173 && jarak_HJ_fix<=213){baik_sekali();}
else if(jarak_HJ_fix>157 && jarak_HJ_fix<=173){baik();}
else if(jarak_HJ_fix>147 && jarak_HJ_fix<=157){cukup();}
else if(jarak_HJ_fix>135 && jarak_HJ_fix<=147){kurang();}
else if(jarak_HJ_fix>117 && jarak_HJ_fix<=135){kurang_sekali();}
else if(jarak_HJ_fix<=117){gagal();}
}
if(umur==4)
{
if(jarak_HJ_fix>244){istimewa();}
else if(jarak_HJ_fix>178 && jarak_HJ_fix<=244){baik_sekali();}
else if(jarak_HJ_fix>165 && jarak_HJ_fix<=178){baik();}
else if(jarak_HJ_fix>155 && jarak_HJ_fix<=165){cukup();}
else if(jarak_HJ_fix>142 && jarak_HJ_fix<=155){kurang();}
else if(jarak_HJ_fix>122 && jarak_HJ_fix<=142){kurang_sekali();}
else if(jarak_HJ_fix<=122){gagal();}
}
if(umur==5)
{
if(jarak_HJ_fix>226){istimewa();}
else if(jarak_HJ_fix>183 && jarak_HJ_fix<=226){baik_sekali();}
else if(jarak_HJ_fix>170 && jarak_HJ_fix<=183){baik();}
else if(jarak_HJ_fix>157 && jarak_HJ_fix<=170){cukup();}
else if(jarak_HJ_fix>145 && jarak_HJ_fix<=157){kurang();}
else if(jarak_HJ_fix>122 && jarak_HJ_fix<=145){kurang_sekali();}
else if(jarak_HJ_fix<=122){gagal();}
}
if(umur==6)
{
if(jarak_HJ_fix>243){istimewa();}
else if(jarak_HJ_fix>183 && jarak_HJ_fix<=243){baik_sekali();}
else if(jarak_HJ_fix>168 && jarak_HJ_fix<=183){baik();}
else if(jarak_HJ_fix>157 && jarak_HJ_fix<=168){cukup();}
else if(jarak_HJ_fix>145 && jarak_HJ_fix<=157){kurang();}
else if(jarak_HJ_fix>127 && jarak_HJ_fix<=145){kurang_sekali();}
else if(jarak_HJ_fix<=127){gagal();}
}
if(umur==7)
{
if(jarak_HJ_fix>231){istimewa();}
else if(jarak_HJ_fix>180 && jarak_HJ_fix<=231){baik_sekali();}
else if(jarak_HJ_fix>168 && jarak_HJ_fix<=180){baik();}
else if(jarak_HJ_fix>155 && jarak_HJ_fix<=168){cukup();}
else if(jarak_HJ_fix>140 && jarak_HJ_fix<=155){kurang();}
else if(jarak_HJ_fix>122 && jarak_HJ_fix<=140){kurang_sekali();}
else if(jarak_HJ_fix<=122){gagal();}
}
if(umur==8)
{
if(jarak_HJ_fix>228){istimewa();}
else if(jarak_HJ_fix>188 && jarak_HJ_fix<=228){baik_sekali();}
else if(jarak_HJ_fix>175 && jarak_HJ_fix<=188){baik();}
else if(jarak_HJ_fix>160 && jarak_HJ_fix<=175){cukup();}
else if(jarak_HJ_fix>145 && jarak_HJ_fix<=160){kurang();}
else if(jarak_HJ_fix>124 && jarak_HJ_fix<=145){kurang_sekali();}
else if(jarak_HJ_fix<=124){gagal();}
}

```

```

    }
}
if(button_BACK)
{
    lcd_ziky.clear();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("Kirim data ke web...");
    paket_data_nodeMcu();
    lcd_ziky.clear();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("Selesai...");
    waktu_fix_HJ=0;
    kecepatan_fix_HJ=0;
    for(int
reset_data_HJ_manual=0;reset_data_HJ_manual<10;reset_data_HJ_manual++)
    {
        dt3[reset_data_HJ_manual]="";
        delay(20);
    }
    lcd_ziky.clear();
    goto mulai_lagi;
}
delay(1000);
lcd_ziky.clear();
goto eksekusi;
}
akhir:
while(1){break;}
}

void horizontal_jump_otomatis()
{
    mulai_lagi:
    while(1)
    {
        zios_servo.write(70);
        kalibrasi_button();
        deteksi1();
        deteksi2();
        deteksi3();
        if(jenis_kelamin==1)
        {
            lcd_ziky.setCursor(0,0);
            lcd_ziky.print("HORIZONTAL JUMP [L]");
        }
        else if(jenis_kelamin==2)
        {
            lcd_ziky.setCursor(0,0);
            lcd_ziky.print("HORIZONTAL JUMP [P]");
        }
        if(jarakD1<=40&&jarakD2<=40&&jarakD3<=40)
        {
            lcd_ziky.setCursor(0,1);
            lcd_ziky.print("    SUDAH SIAP    ");
            lcd_ziky.setCursor(0,2);
            lcd_ziky.print("    mulai...    ");
            lcd_ziky.setCursor(0,3);
            lcd_ziky.print("Tekan <BACK> utk bck");
            encoderValue=0;
            delay(4000);
            lcd_ziky.clear();
            goto scanTBBB;
        }
    }
    if(button_BACK)
    {
        lcd_ziky.clear();
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("Selesai...");
        delay(300);
        lcd_ziky.clear();
        goto akhir;
    }
}

```

```

else if(jarakD1>40&&jarakD2>40&&jarakD3>40)
{
  lcd_ziky.setCursor(0,1);
  lcd_ziky.print("      BELUM SIAP      ");
  lcd_ziky.setCursor(0,2);
  lcd_ziky.print("Belum Naik Timbangan");
  lcd_ziky.setCursor(0,3);
  lcd_ziky.print("Tekan <BACK> utk bck");
}
}

//=====================================================
scanTBBB:
while(1)
{
  motor_turun();
  TB_enc=tiang1-encoderValue*0.0558;
  pembagi_TB=TB_enc;
  hasil_bagi_TB=TB_enc-pembagi_TB;
  hasil_belakang_koma_TB=hasil_bagi_TB*10;
  lcd_ziky.setCursor(0, 0);
  lcd_ziky.print("Scan.....");
  lcd_ziky.setCursor(0, 1);
  lcd_ziky.print("Tinggi :");
  lcd_ziky.print(TB_enc,1);
  Serial.println(TB_enc,1);
  lcd_ziky.setCursor(0, 2);
  lcd_ziky.print("Berat :0.0");
  val_kepala=digitalRead(limit_kepala);
  if(val_kepala==LOW)
  {
    digitalWrite(buzzer,HIGH);
    EEPROM.write(addr_TB, TB_enc);
    EEPROM.write(addr_TB_belakang_koma, hasil_belakang_koma_TB);
    delay(50);
    digitalWrite(buzzer,LOW);
    motor_naik();
    delay(700);
    goto berat_scan;
  }
  val_bawah=digitalRead(limit_bawah);
  if(val_bawah==LOW)
  {
    digitalWrite(buzzer,HIGH);
    EEPROM.write(addr_TB, TB_enc);
    EEPROM.write(addr_TB_belakang_koma, hasil_belakang_koma_TB);
    motor_stop();
    delay(200);
    digitalWrite(buzzer,LOW);
    goto pengukuran_bawah;
  }
}
berat_scan:
while(1)
{
  motor_stop();
  timer();
  sw_millis.start();
  sw_micros.start();
  sw_secs.start();
  berat_badan();
  simpan_TB_koma();
  lcd_ziky.setCursor(0, 0);
  lcd_ziky.print("Scan.....");
  lcd_ziky.setCursor(0, 1);
  lcd_ziky.print("Tinggi :");
  lcd_ziky.print(TB_fix);
  lcd_ziky.print(" ");
  lcd_ziky.setCursor(0, 2);
  lcd_ziky.print("Berat :");
  lcd_ziky.print(timbangan,1);
  lcd_ziky.print(" ");
}

```

```

if(teta_waktu>5)
{
  sw_millis.stop();
  sw_micros.stop();
  sw_secs.stop();
  pembagi_BB=timbangan;
  hasil_bagi_BB=timbangan-pembagi_BB;
  hasil_belakang_koma_BB=hasil_bagi_BB*10;
  EEPROM.write(addr_BB, timbangan);
  EEPROM.write(addr_BB_belakang_koma, hasil_belakang_koma_BB);
  for(int a=0; a<5;a++)
  {
    lcd_ziky.setCursor(0, 0);
    lcd_ziky.print("Selesai...");
    lcd_ziky.setCursor(0, 1);
    lcd_ziky.print("Tinggi :");
    lcd_ziky.print(TB_fix);
    lcd_ziky.print("Cm");
    lcd_ziky.setCursor(0, 2);
    lcd_ziky.print("Berat :");
    lcd_ziky.print(timbangan,1);
    lcd_ziky.print("Kg");
    digitalWrite(buzzer,HIGH);
    delay(200);
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("                ");
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print("                ");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print("                ");
    digitalWrite(buzzer,LOW);
    delay(100);
  }
  lcd_ziky.clear();
  goto selesai;
}
}
pengukuran_bawah:
while(1)
{
  sw_millis.start();
  sw_micros.start();
  sw_secs.start();
  tinggi_badan();
  timer();
  berat_badan();
  tinggi_badan();
  //Simpan TB dengan koma
  pembagi_TB=TB;
  hasil_bagi_TB=TB-pembagi_TB;
  hasil_belakang_koma_TB=hasil_bagi_TB*10;
  //Simpan BB dengan koma
  pembagi_BB=timbangan;
  hasil_bagi_BB=timbangan-pembagi_BB;
  hasil_belakang_koma_BB=hasil_bagi_BB*10;
  lcd_ziky.setCursor(0, 0);
  lcd_ziky.print("Scan.....");
  lcd_ziky.setCursor(0, 1);
  lcd_ziky.print("Tinggi :");
  lcd_ziky.print(TB);
  lcd_ziky.print(" ");
  lcd_ziky.setCursor(0, 2);
  lcd_ziky.print("Berat :");
  lcd_ziky.print(timbangan,1);
  lcd_ziky.print(" ");
  val_kepala=digitalRead(limit_kepala);
  if(val_kepala==LOW)
  {
    goto selesai;
  }
  if(teta_waktu>5)
  {

```

```

sw_millis.stop();
sw_micros.stop();
sw_secs.stop();
EEPROM.write(addr_BB, timbangan);
EEPROM.write(addr_BB_belakang_koma, hasil_belakang_koma_BB);
EEPROM.write(addr_TB, TB);
EEPROM.write(addr_TB_belakang_koma, hasil_belakang_koma_TB);
for(int a=0; a<5;a++)
{
  lcd_ziky.setCursor(0, 0);
  lcd_ziky.print("Selesai...");
  lcd_ziky.setCursor(0, 1);
  lcd_ziky.print("Tinggi :");
  lcd_ziky.print(TB);
  lcd_ziky.print("Cm");
  lcd_ziky.setCursor(0, 2);
  lcd_ziky.print("Berat :");
  lcd_ziky.print(timbangan,1);
  lcd_ziky.print("Kg");
  digitalWrite(buzzer,HIGH);
  delay(200);
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("                ");
  lcd_ziky.setCursor(0,1);
  lcd_ziky.print("                ");
  lcd_ziky.setCursor(0,2);
  lcd_ziky.print("                ");
  digitalWrite(buzzer,LOW);
  delay(100);
}
goto selesai;
}
}
selesai:
while(1)
{
  simpan_TB_koma();
  simpan_BB_koma();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Simpan...");
  motor_naik();
  Serial.print("TB=");
  Serial.println(TB_fix);
  Serial.print("BB=");
  Serial.println(BB_fix);
  val_atas=digitalRead(limit_atas);
  if(val_atas==LOW)
  {
    digitalWrite(buzzer,HIGH);
    motor_stop();
    sw_millis.reset();
    sw_micros.reset();
    sw_secs.reset();
    encoderValue=0;
    zios_servo.write(115);
    delay(50);
    digitalWrite(buzzer,LOW);
    delay(1000);
    lcd_ziky.clear();
    goto siapsiap1;
  }
}

//=====
siapsiap1:
while(1)
{
  deteksi1();
  deteksi2();
  deteksi3();
  if(jarakD1<=70&&jarakD2<=70&&jarakD3<=70)
  {

```

```

        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("        ready        ");
        Serial2.println(1);
        goto siapsiap2;
    }
    else if(jarakD1>70&&jarakD2>70&&jarakD3>70)
    {
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("Ambil Ancang-ancang ");
        delay(250);
        lcd_ziky.clear();
        Serial2.println(7);
    }
}

siapsiap2:
while(1)
{
    deteksi1();
    deteksi2();
    deteksi3();
    if(jarakD1<=70&&jarakD2<=70&&jarakD3<=70)
    {
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("        ready        ");
    }
    else if(jarakD1>70&&jarakD2>70&&jarakD3>70)
    {
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("        GO!!!        ");
        delay(250);
        lcd_ziky.clear();
        Serial2.println(7);
        delay(10);
        Serial2.println(9);
        goto eksekusi;
    }
}
eksekusi:
while(1)
{
    kalibrasi_button();
    rumus_powerDankecepatan_HJ();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("P.lom :");
    lcd_ziky.print(jarak_HJ_fix);
    lcd_ziky.print(" Cm");
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print("Kec    :");
    lcd_ziky.print(kecepatan_fix_HJ);
    lcd_ziky.print(" m/s");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print("Power :");
    lcd_ziky.print(power_fix_HJ);
    lcd_ziky.print(" Kgm/s");
    lcd_ziky.setCursor(0,3);
    lcd_ziky.print("        :");
    lcd_ziky.print(horse_power_HJ);
    lcd_ziky.print(" Hp");
    if(button_BACK)
    {
        lcd_ziky.clear();
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("Kirim data ke web...");
        paket_data_nodeMcu();
        lcd_ziky.clear();
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("Selesai...");
        waktu_fix_HJ=0;
        kecepatan_fix_HJ=0;
        for(int
reset_data_HJ manual=0;reset_data_HJ manual<10;reset_data_HJ manual++)

```

```

    {
        dt3[reset_data_HJ_manual]="";
        delay(20);
    }
    lcd_ziky.clear();
    goto mulai_lagi;
}
delay(2000);
lcd_ziky.clear();
goto gantian1;
}

gantian1:
while(1)
{
    kalibrasi_button();
    rumus_powerDankecepatan_HJ();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("_____");
    for(int a=0;a<20;a++){lcd_ziky.setCursor(a,2);lcd_ziky.write(3);}
    if(jenis_kelamin==1)
    {
        if(umur==1)
        {
            if(jarak_HJ_fix>196){istimewa();}
            else if(jarak_HJ_fix>168 && jarak_HJ_fix<=196){baik_sekali();}
            else if(jarak_HJ_fix>155 && jarak_HJ_fix<=168){baik();}
            else if(jarak_HJ_fix>145 && jarak_HJ_fix<=155){cukup();}
            else if(jarak_HJ_fix>135 && jarak_HJ_fix<=145){kurang();}
            else if(jarak_HJ_fix>115 && jarak_HJ_fix<=135){kurang_sekali();}
            else if(jarak_HJ_fix<=115){gagal();}
        }
        if(umur==2)
        {
            if(jarak_HJ_fix>226){istimewa();}
            else if(jarak_HJ_fix>175 && jarak_HJ_fix<=226){baik_sekali();}
            else if(jarak_HJ_fix>168 && jarak_HJ_fix<=175){baik();}
            else if(jarak_HJ_fix>152 && jarak_HJ_fix<=168){cukup();}
            else if(jarak_HJ_fix>140 && jarak_HJ_fix<=152){kurang();}
            else if(jarak_HJ_fix>122 && jarak_HJ_fix<=140){kurang_sekali();}
            else if(jarak_HJ_fix<=122){gagal();}
        }
        if(umur==3)
        {
            if(jarak_HJ_fix>256){istimewa();}
            else if(jarak_HJ_fix>183 && jarak_HJ_fix<=256){baik_sekali();}
            else if(jarak_HJ_fix>170 && jarak_HJ_fix<=183){baik();}
            else if(jarak_HJ_fix>160 && jarak_HJ_fix<=170){cukup();}
            else if(jarak_HJ_fix>147 && jarak_HJ_fix<=160){kurang();}
            else if(jarak_HJ_fix>127 && jarak_HJ_fix<=147){kurang_sekali();}
            else if(jarak_HJ_fix<=127){gagal();}
        }
        if(umur==4)
        {
            if(jarak_HJ_fix>259){istimewa();}
            else if(jarak_HJ_fix>196 && jarak_HJ_fix<=259){baik_sekali();}
            else if(jarak_HJ_fix>173 && jarak_HJ_fix<=196){baik();}
            else if(jarak_HJ_fix>168 && jarak_HJ_fix<=173){cukup();}
            else if(jarak_HJ_fix>152 && jarak_HJ_fix<=168){kurang();}
            else if(jarak_HJ_fix>132 && jarak_HJ_fix<=152){kurang_sekali();}
            else if(jarak_HJ_fix<=132){gagal();}
        }
        if(umur==5)
        {
            if(jarak_HJ_fix>274){istimewa();}
            else if(jarak_HJ_fix>211 && jarak_HJ_fix<=274){baik_sekali();}
            else if(jarak_HJ_fix>193 && jarak_HJ_fix<=211){baik();}
            else if(jarak_HJ_fix>180 && jarak_HJ_fix<=193){cukup();}
            else if(jarak_HJ_fix>163 && jarak_HJ_fix<=180){kurang();}
            else if(jarak_HJ_fix>142 && jarak_HJ_fix<=163){kurang_sekali();}
            else if(jarak_HJ_fix<=142){gagal();}
        }
    }
}

```

```

if (umur==6)
{
    if(jarak_HJ_fix>274){istimewa();}
    else if(jarak_HJ_fix>221 && jarak_HJ_fix<=274){baik_sekali();}
    else if(jarak_HJ_fix>208 && jarak_HJ_fix<=221){baik();}
    else if(jarak_HJ_fix>196 && jarak_HJ_fix<=208){cukup();}
    else if(jarak_HJ_fix>180 && jarak_HJ_fix<=196){kurang();}
    else if(jarak_HJ_fix>157 && jarak_HJ_fix<=180){kurang_sekali();}
    else if(jarak_HJ_fix<=157){gagal();}
}
if (umur==7)
{
    if(jarak_HJ_fix>279){istimewa();}
    else if(jarak_HJ_fix>229 && jarak_HJ_fix<=279){baik_sekali();}
    else if(jarak_HJ_fix>218 && jarak_HJ_fix<=229){baik();}
    else if(jarak_HJ_fix>205 && jarak_HJ_fix<=218){cukup();}
    else if(jarak_HJ_fix>193 && jarak_HJ_fix<=205){kurang();}
    else if(jarak_HJ_fix>160 && jarak_HJ_fix<=193){kurang_sekali();}
    else if(jarak_HJ_fix<=160){gagal();}
}
if (umur==8)
{
    if(jarak_HJ_fix>300){istimewa();}
    else if(jarak_HJ_fix>239 && jarak_HJ_fix<=300){baik_sekali();}
    else if(jarak_HJ_fix>226 && jarak_HJ_fix<=239){baik();}
    else if(jarak_HJ_fix>213 && jarak_HJ_fix<=226){cukup();}
    else if(jarak_HJ_fix>193 && jarak_HJ_fix<=213){kurang();}
    else if(jarak_HJ_fix>165 && jarak_HJ_fix<=193){kurang_sekali();}
    else if(jarak_HJ_fix<=165){gagal();}
}
}

if(jenis_kelamin==2)
{
    if (umur==1)
    {
        if(jarak_HJ_fix>241){istimewa();}
        else if(jarak_HJ_fix>158 && jarak_HJ_fix<=241){baik_sekali();}
        else if(jarak_HJ_fix>147 && jarak_HJ_fix<=158){baik();}
        else if(jarak_HJ_fix>137 && jarak_HJ_fix<=147){cukup();}
        else if(jarak_HJ_fix>122 && jarak_HJ_fix<=137){kurang();}
        else if(jarak_HJ_fix>104 && jarak_HJ_fix<=122){kurang_sekali();}
        else if(jarak_HJ_fix<=104){gagal();}
    }
    if (umur==2)
    {
        if(jarak_HJ_fix>213){istimewa();}
        else if(jarak_HJ_fix>165 && jarak_HJ_fix<=213){baik_sekali();}
        else if(jarak_HJ_fix>155 && jarak_HJ_fix<=165){baik();}
        else if(jarak_HJ_fix>142 && jarak_HJ_fix<=155){cukup();}
        else if(jarak_HJ_fix>130 && jarak_HJ_fix<=142){kurang();}
        else if(jarak_HJ_fix>112 && jarak_HJ_fix<=130){kurang_sekali();}
        else if(jarak_HJ_fix<=112){gagal();}
    }
    if (umur==3)
    {
        if(jarak_HJ_fix>256){istimewa();}
        else if(jarak_HJ_fix>173 && jarak_HJ_fix<=213){baik_sekali();}
        else if(jarak_HJ_fix>157 && jarak_HJ_fix<=173){baik();}
        else if(jarak_HJ_fix>147 && jarak_HJ_fix<=157){cukup();}
        else if(jarak_HJ_fix>135 && jarak_HJ_fix<=147){kurang();}
        else if(jarak_HJ_fix>117 && jarak_HJ_fix<=135){kurang_sekali();}
        else if(jarak_HJ_fix<=117){gagal();}
    }
    if (umur==4)
    {
        if(jarak_HJ_fix>244){istimewa();}
        else if(jarak_HJ_fix>178 && jarak_HJ_fix<=244){baik_sekali();}
        else if(jarak_HJ_fix>165 && jarak_HJ_fix<=178){baik();}
        else if(jarak_HJ_fix>155 && jarak_HJ_fix<=165){cukup();}
        else if(jarak_HJ_fix>142 && jarak_HJ_fix<=155){kurang();}
        else if(jarak_HJ_fix>122 && jarak_HJ_fix<=142){kurang_sekali();}
    }
}

```

```

else if(jarak_HJ_fix<=122){gagal();}
}
if(umur==5)
{
if(jarak_HJ_fix>226){istimewa();}
else if(jarak_HJ_fix>183 && jarak_HJ_fix<=226){baik_sekali();}
else if(jarak_HJ_fix>170 && jarak_HJ_fix<=183){baik();}
else if(jarak_HJ_fix>157 && jarak_HJ_fix<=170){cukup();}
else if(jarak_HJ_fix>145 && jarak_HJ_fix<=157){kurang();}
else if(jarak_HJ_fix>122 && jarak_HJ_fix<=145){kurang_sekali();}
else if(jarak_HJ_fix<=122){gagal();}
}
if(umur==6)
{
if(jarak_HJ_fix>243){istimewa();}
else if(jarak_HJ_fix>183 && jarak_HJ_fix<=243){baik_sekali();}
else if(jarak_HJ_fix>168 && jarak_HJ_fix<=183){baik();}
else if(jarak_HJ_fix>157 && jarak_HJ_fix<=168){cukup();}
else if(jarak_HJ_fix>145 && jarak_HJ_fix<=157){kurang();}
else if(jarak_HJ_fix>127 && jarak_HJ_fix<=145){kurang_sekali();}
else if(jarak_HJ_fix<=127){gagal();}
}
if(umur==7)
{
if(jarak_HJ_fix>231){istimewa();}
else if(jarak_HJ_fix>180 && jarak_HJ_fix<=231){baik_sekali();}
else if(jarak_HJ_fix>168 && jarak_HJ_fix<=180){baik();}
else if(jarak_HJ_fix>155 && jarak_HJ_fix<=168){cukup();}
else if(jarak_HJ_fix>140 && jarak_HJ_fix<=155){kurang();}
else if(jarak_HJ_fix>122 && jarak_HJ_fix<=140){kurang_sekali();}
else if(jarak_HJ_fix<=122){gagal();}
}
if(umur==8)
{
if(jarak_HJ_fix>228){istimewa();}
else if(jarak_HJ_fix>188 && jarak_HJ_fix<=228){baik_sekali();}
else if(jarak_HJ_fix>175 && jarak_HJ_fix<=188){baik();}
else if(jarak_HJ_fix>160 && jarak_HJ_fix<=175){cukup();}
else if(jarak_HJ_fix>145 && jarak_HJ_fix<=160){kurang();}
else if(jarak_HJ_fix>124 && jarak_HJ_fix<=145){kurang_sekali();}
else if(jarak_HJ_fix<=124){gagal();}
}
}
if(button_BACK)
{
lcd_ziky.clear();
lcd_ziky.setCursor(0,0);
lcd_ziky.print("Kirim data ke web...");
paket_data_nodeMcu();
lcd_ziky.clear();
lcd_ziky.setCursor(0,0);
lcd_ziky.print("Selesai...");
waktu_fix_HJ=0;
kecepatan_fix_HJ=0;
for(int
reset_data_HJ_manual=0;reset_data_HJ_manual<10;reset_data_HJ_manual++)
{
dt3[reset_data_HJ_manual]="";
delay(20);
}
lcd_ziky.clear();
goto mulai_lagi;
}
delay(1000);
lcd_ziky.clear();
goto eksekusi;
}
akhir:
while(1){break;}
}

void jenis_kelamin_manual()

```

```

{
while(1)
{
  read_eeprom_mPower=EEPROM.read(addr_mPower);
  kalibrasi_button();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("  JENIS KELAMIN");
  if(button_BACK)
  {
    lcd_ziky.clear();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("Selesai...");
    delay(300);
    lcd_ziky.clear();
    break;
  }
  if(button_down)
  {
    jenis_kelamin++;
    if(jenis_kelamin>2) jenis_kelamin=1;
    delay(120);
  }
  if(button_up)
  {
    jenis_kelamin--;
    if(jenis_kelamin<1) jenis_kelamin=2;
    delay(120);
  }
  if(jenis_kelamin==1)
  {
    lcd_ziky.setCursor(0,1);
    lcd_ziky.write(1);
    lcd_ziky.print("Laki-laki");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print(" Perempuan");
    if(button_ok)
    {
      lcd_ziky.clear();
      delay(300);
      if(menu==2){WBR_setting();}
      else if(menu_PT==0 && read_eeprom_mPower==1){vertikal_jump();}
      else if(menu_PT==0 && read_eeprom_mPower==2){vertikal_jump_otomatis();}
      else if(menu_PT==1 && read_eeprom_mPower==1){horizontal_jump_manual();}
      else if(menu_PT==1 &&
read_eeprom_mPower==2){horizontal_jump_otomatis();}
    }
  }
  if(jenis_kelamin==2)
  {
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print(" Laki-laki");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.write(1);
    lcd_ziky.print("Perempuan");
    if(button_ok)
    {
      lcd_ziky.clear();
      delay(300);
      if(menu==2){WBR_setting();}
      else if(menu_PT==0 && read_eeprom_mPower==1){vertikal_jump();}
      else if(menu_PT==0 && read_eeprom_mPower==2){vertikal_jump_otomatis();}
      else if(menu_PT==1 && read_eeprom_mPower==1){horizontal_jump_manual();}
      else if(menu_PT==1 &&
read_eeprom_mPower==2){horizontal_jump_otomatis();}
    }
  }
}
}

void jenis_kelamin_khusus_VJHJ()
{
  atas:

```

```

while(1)
{
  read_eeprom_mPower=EEPROM.read(addr_mPower);
  kalibrasi_button();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("  KELAMIN      USIA");
  for(int i=0;i<20;i++)
  {
    lcd_ziky.setCursor(i,1);
    lcd_ziky.write(3);
    lcd_ziky.setCursor(11,0);
    lcd_ziky.write(8);
    lcd_ziky.setCursor(11,1);
    lcd_ziky.write(8);
    lcd_ziky.setCursor(11,2);
    lcd_ziky.write(8);
    lcd_ziky.setCursor(11,3);
    lcd_ziky.write(8);
  }
  if(button_BACK)
  {
    lcd_ziky.clear();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("Selesai...");
    delay(300);
    lcd_ziky.clear();
    break;
  }
  if(button_down)
  {
    jenis_kelamin++;
    if(jenis_kelamin>2) jenis_kelamin=1;
    delay(120);
  }
  if(button_up)
  {
    jenis_kelamin--;
    if(jenis_kelamin<1) jenis_kelamin=2;
    delay(120);
  }
  if(jenis_kelamin==1)
  {
    lcd_ziky.setCursor(0,2);
    lcd_ziky.write(1);
    lcd_ziky.print("Laki-laki");
    lcd_ziky.setCursor(0,3);
    lcd_ziky.print(" Perempuan");
    if(button_ok)
    {
      delay(100);
      while(1)
      {
        kalibrasi_button();
        lcd_ziky.setCursor(12,2);
        lcd_ziky.write(4);
        if(menu_PT==0)
        {
          if(button_up){umur++;if(umur>10)umur=1; delay(130);}
          if(button_down){umur--;if(umur<1)umur=10; delay(130);}
          if(umur==1){lcd_ziky.setCursor(13,2);lcd_ziky.print("9 th ");}
          if(umur==2){lcd_ziky.setCursor(13,2);lcd_ziky.print("10 th ");}
          if(umur==3){lcd_ziky.setCursor(13,2);lcd_ziky.print("11 th ");}
          if(umur==4){lcd_ziky.setCursor(13,2);lcd_ziky.print("12 th ");}
          if(umur==5){lcd_ziky.setCursor(13,2);lcd_ziky.print("13 th ");}
          if(umur==6){lcd_ziky.setCursor(13,2);lcd_ziky.print("14 th ");}
          if(umur==7){lcd_ziky.setCursor(13,2);lcd_ziky.print("15 th ");}
          if(umur==8){lcd_ziky.setCursor(13,2);lcd_ziky.print("16 th");}
          if(umur==9){lcd_ziky.setCursor(13,2);lcd_ziky.print("17 th");}
          if(umur==10){lcd_ziky.setCursor(13,2);lcd_ziky.print(">18 th");}
        }
        if(menu_PT==1)
        {

```

```

        if(button_up){umur++;if(umur>8)umur=1; delay(130);}
        if(button_down){umur--;if(umur<1)umur=8; delay(130);}
        if(umur==1){lcd_ziky.setCursor(13,2);lcd_ziky.print("10 th ");}
        if(umur==2){lcd_ziky.setCursor(13,2);lcd_ziky.print("11 th ");}
        if(umur==3){lcd_ziky.setCursor(13,2);lcd_ziky.print("12 th ");}
        if(umur==4){lcd_ziky.setCursor(13,2);lcd_ziky.print("13 th ");}
        if(umur==5){lcd_ziky.setCursor(13,2);lcd_ziky.print("14 th ");}
        if(umur==6){lcd_ziky.setCursor(13,2);lcd_ziky.print("15 th ");}
        if(umur==7){lcd_ziky.setCursor(13,2);lcd_ziky.print("16 th ");}
        if(umur==8){lcd_ziky.setCursor(13,2);lcd_ziky.print(">17 th");}
    }
    if(button_ok)
    {
        lcd_ziky.clear();
        delay(300);
        if(menu_PT==0 && read_eeprom_mPower==1){vertikal_jump();}
        else if(menu_PT==0 &&
read_eeprom_mPower==2){vertikal_jump_otomatis();}
        else if(menu_PT==1 &&
read_eeprom_mPower==1){horizontal_jump_manual();}
        else if(menu_PT==1 &&
read_eeprom_mPower==2){horizontal_jump_otomatis();}
    }
    if(button_BACK)
    {
        lcd_ziky.setCursor(12,2);lcd_ziky.print("      ");
        delay(200);
        goto atas;
    }
}
}
}
if(jenis_kelamin==2)
{
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print(" Laki-laki");
    lcd_ziky.setCursor(0,3);
    lcd_ziky.write(1);
    lcd_ziky.print("Perempuan");
    if(button_ok)
    {
        delay(100);
        while(1)
        {
            kalibrasi_button();
            lcd_ziky.setCursor(12,3);
            lcd_ziky.write(4);
            if(menu_PT==0)
            {
                if(button_up){umur++;if(umur>10)umur=1; delay(130);}
                if(button_down){umur--;if(umur<1)umur=10; delay(130);}
                if(umur==1){lcd_ziky.setCursor(13,3);lcd_ziky.print("9 th ");}
                if(umur==2){lcd_ziky.setCursor(13,3);lcd_ziky.print("10 th ");}
                if(umur==3){lcd_ziky.setCursor(13,3);lcd_ziky.print("11 th ");}
                if(umur==4){lcd_ziky.setCursor(13,3);lcd_ziky.print("12 th ");}
                if(umur==5){lcd_ziky.setCursor(13,3);lcd_ziky.print("13 th ");}
                if(umur==6){lcd_ziky.setCursor(13,3);lcd_ziky.print("14 th ");}
                if(umur==7){lcd_ziky.setCursor(13,3);lcd_ziky.print("15 th ");}
                if(umur==8){lcd_ziky.setCursor(13,3);lcd_ziky.print("16 th");}
                if(umur==9){lcd_ziky.setCursor(13,3);lcd_ziky.print("17 th");}
                if(umur==10){lcd_ziky.setCursor(13,3);lcd_ziky.print(">18 th");}
            }
        }
        if(menu_PT==1)
        {
            if(button_up){umur++;if(umur>8)umur=1; delay(130);}
            if(button_down){umur--;if(umur<1)umur=8; delay(130);}
            if(umur==1){lcd_ziky.setCursor(13,3);lcd_ziky.print("10 th ");}
            if(umur==2){lcd_ziky.setCursor(13,3);lcd_ziky.print("11 th ");}
            if(umur==3){lcd_ziky.setCursor(13,3);lcd_ziky.print("12 th ");}
            if(umur==4){lcd_ziky.setCursor(13,3);lcd_ziky.print("13 th ");}
            if(umur==5){lcd_ziky.setCursor(13,3);lcd_ziky.print("14 th ");}
            if(umur==6){lcd_ziky.setCursor(13,3);lcd_ziky.print("15 th ");}
        }
    }
}
}

```

```

        if (umur==7) {lcd_ziky.setCursor(13,3);lcd_ziky.print("16 th ");}
        if (umur==8) {lcd_ziky.setCursor(13,3);lcd_ziky.print(">17 th");}
    }
    if (button_ok)
    {
        lcd_ziky.clear();
        delay(300);
        if (menu_PT==0 && read_eeeprom_mPower==1) {vertikal_jump();}
        else if (menu_PT==0 &&
read_eeeprom_mPower==2) {vertikal_jump_otomatis();}
        else if (menu_PT==1 &&
read_eeeprom_mPower==1) {horizontal_jump_manual();}
        else if (menu_PT==1 &&
read_eeeprom_mPower==2) {horizontal_jump_otomatis();}
    }
    if (button_BACK)
    {
        lcd_ziky.setCursor(12,3);lcd_ziky.print("      ");
        delay(200);
        goto atas;
    }
}
}
}
}

void paket_data_nodeMcu()
{
    simpan_TB_koma();
    simpan_BB_koma();
    rumus_powerDankecepatan_VJ();
    rumus_powerDankecepatan_HJ();
    simpan_WBR_koma();
    read_eeeprom_PE[1]=EEPROM.read(addr_PE1);
    read_eeeprom_PE[2]=EEPROM.read(addr_PE2);
    read_eeeprom_PE[3]=EEPROM.read(addr_PE3);
    read_eeeprom_PE[4]=EEPROM.read(addr_PE4);
    read_eeeprom_PE[5]=EEPROM.read(addr_PE5);
    read_eeeprom_PE[6]=EEPROM.read(addr_PE6);
    read_eeeprom_PE[7]=EEPROM.read(addr_PE7);
    read_eeeprom_PE[8]=EEPROM.read(addr_PE8);
    read_eeeprom_PE[9]=EEPROM.read(addr_PE9);
    read_eeeprom_PE[10]=EEPROM.read(addr_PE10);
    read_eeeprom_rata2_PE=EEPROM.read(addr_rata2_PE);
    read_eeeprom_min_PE=EEPROM.read(addr_min_PE);
    read_eeeprom_max_PE=EEPROM.read(addr_max_PE);
    int ePE1=read_eeeprom_PE[1];
    int ePE2=read_eeeprom_PE[2];
    int ePE3=read_eeeprom_PE[3];
    int ePE4=read_eeeprom_PE[4];
    int ePE5=read_eeeprom_PE[5];
    int ePE6=read_eeeprom_PE[6];
    int ePE7=read_eeeprom_PE[7];
    int ePE8=read_eeeprom_PE[8];
    int ePE9=read_eeeprom_PE[9];
    int ePE10=read_eeeprom_PE[10];
    sTB=String(TB_fix);
    sBB=String(BB_fix);
    s_T_LOM_VJ=String(read_eeeprom_VJ);
    s_waktu_VJ=String(waktu_fix_VJ);
    s_kec_VJ=String(kecepatan_fix_VJ);
    s_power_VJ=String(power_fix_VJ);
    s_P_LOM_HJ=String(read_eeeprom_HJ);
    s_waktu_HJ=String(waktu_fix_HJ);
    s_kec_HJ=String(kecepatan_fix_HJ);
    s_power_HJ=String(power_fix_HJ);
    sWBR=String(WBR_fix);
    sPE1=String(ePE1);
    sPE2=String(ePE2);
    sPE3=String(ePE3);
    sPE4=String(ePE4);
}

```

```

sPE5=String(ePE5);
sPE6=String(ePE6);
sPE7=String(ePE7);
sPE8=String(ePE8);
sPE9=String(ePE9);
sPE10=String(ePE10);
s_rata2PE=String(read_eeeprom_rata2_PE);
s_min_PE=String(read_eeeprom_min_PE);
s_max_PE=String(read_eeeprom_max_PE);

Serial.println("=====");
Serial.println("TINGGI BADAN");
Serial.println("=====");
Serial.print("Tinggi Badan:");
Serial.println(sTB);
Serial.println();
Serial.println();
Serial.println("=====");
Serial.println("BERAT BADAN");
Serial.println("=====");
Serial.print("Berat Badan:");
Serial.println(sBB);
Serial.println();
Serial.println();
Serial.println("=====");
Serial.println("VERTIKAL JUMP");
Serial.println("=====");
Serial.print("Tinggi Lomptan:");
Serial.println(s_T_LOM_VJ);
Serial.print("Waktu Lompatan:");
Serial.println(s_waktu_VJ);
Serial.print("Kecepatan Lompatan:");
Serial.println(s_kec_VJ);
Serial.print("Power Lompatan:");
Serial.println(s_power_VJ);
Serial.println();
Serial.println();
Serial.println("=====");
Serial.println("HORIZONTAL JUMP");
Serial.println("=====");
Serial.print("Panjang Lomptan:");
Serial.println(s_P_LOM_HJ);
Serial.print("Waktu Lompatan:");
Serial.println(s_waktu_HJ);
Serial.print("Kecepatan Lompatan:");
Serial.println(s_kec_HJ);
Serial.print("Power Lompatan:");
Serial.println(s_power_HJ);
Serial.println();
Serial.println();
Serial.println("=====");
Serial.println("WHOLE BODY REACTION");
Serial.println("=====");
Serial.print("Reaction:");
Serial.println(sWBR);
Serial.println();
Serial.println();
Serial.println("=====");
Serial.println("POWER ENDURANCE");
Serial.println("=====");
Serial.print("Tinggi Lompatan1:");
Serial.println(sPE1);
Serial.print("Tinggi Lompatan2:");
Serial.println(sPE2);
Serial.print("Tinggi Lompatan3:");
Serial.println(sPE3);
Serial.print("Tinggi Lompatan4:");
Serial.println(sPE4);
Serial.print("Tinggi Lompatan5:");
Serial.println(sPE5);
Serial.print("Tinggi Lompatan6:");
Serial.println(sPE6);

```

```

Serial.print("Tinggi Lompatan7:");
Serial.println(sPE7);
Serial.print("Tinggi Lompatan8:");
Serial.println(sPE8);
Serial.print("Tinggi Lompatan9:");
Serial.println(sPE9);
Serial.print("Tinggi Lompatan10:");
Serial.println(sPE10);
Serial.print("Rata-rata Lompatan:");
Serial.println(s_rata2PE);
Serial.print("Maksimal Lompatan:");
Serial.println(s_max_PE);
Serial.print("Minimal Lompatan:");
Serial.println(s_min_PE);
Serial.println();
Serial.println();

Serial3.print(sTB);
Serial3.print(",");
Serial3.print(sBB);
Serial3.print(",");
Serial3.print(s_T_LOM_VJ);
Serial3.print(",");
Serial3.print(s_waktu_VJ);
Serial3.print(",");
Serial3.print(s_kec_VJ);
Serial3.print(",");
Serial3.print(s_power_VJ);
Serial3.print(",");
Serial3.print(s_P_LOM_HJ);
Serial3.print(",");
Serial3.print(s_waktu_HJ);
Serial3.print(",");
Serial3.print(s_kec_HJ);
Serial3.print(",");
Serial3.print(s_power_HJ);
Serial3.print(",");
Serial3.print(sWBR);
Serial3.print(",");
Serial3.print(sPE1);
Serial3.print(",");
Serial3.print(sPE2);
Serial3.print(",");
Serial3.print(sPE3);
Serial3.print(",");
Serial3.print(sPE4);
Serial3.print(",");
Serial3.print(sPE5);
Serial3.print(",");
Serial3.print(sPE6);
Serial3.print(",");
Serial3.print(sPE7);
Serial3.print(",");
Serial3.print(sPE8);
Serial3.print(",");
Serial3.print(sPE9);
Serial3.print(",");
Serial3.print(sPE10);
Serial3.print(",");
Serial3.print(s_rata2PE);
Serial3.print(",");
Serial3.print(s_max_PE);
Serial3.print(",");
Serial3.println(s_min_PE);
delay(1000);
}

void parsing_data_PE()
{
  if(Serial1.available()>0)
  {
    char inChar = Serial1.read();

```

```

    dataIn1 += inChar;
    if (inChar == '\n')
    {
        parsing = true;
    }
}
if(parsing)
{
    parsingDataPE();
    parsing=false;
    dataIn1="";
}
}

void parsing_data2()
{
    if(Serial1.available()>0)
    {
        char inChar = Serial1.read();
        dataIn2 += inChar;
        if (inChar == '\n')
        {
            parsing = true;
        }
    }
    if(parsing)
    {
        parsingData2();
        parsing=false;
        dataIn2="";
    }
}

void parsing_data_horizontal_jump()
{
    if(Serial2.available()>0)
    {
        char inChar = Serial2.read();
        dataIn3 += inChar;
        if (inChar == '\n')
        {
            parsing = true;
        }
    }
    if(parsing)
    {
        parsingDataHorizontalJump();
        parsing=false;
        dataIn3="";
    }
}

void parsingDataPE()
{
    int j=0;

    Serial.print("data masuk : ");
    Serial.print(dataIn1);
    Serial.print("\n");

    dt1[j]="";
    for(i=0;i<=dataIn1.length();i++)
    {
        if ((dataIn1[i] == '#') || (dataIn1[i] == ','))
        {
            j++;
            dt1[j]="";          //inisialisasi variabel array dt[j]
        }
        else
        {
            dt1[j] = dt1[j] + dataIn1[i];
        }
    }
}

```

```

}

rata2tinggi_lompat=(dt1[0].toInt()+dt1[1].toInt()+dt1[2].toInt()+dt1[3].toInt()+
dt1[4].toInt()+dt1[5].toInt()+dt1[6].toInt()+dt1[7].toInt()+dt1[8].toInt()+dt1[9
].toInt())/10;
Serial.print("data 1 : ");
Serial.print(dt1[0].toInt());
Serial.print("\n");
Serial.print("data 2 : ");
Serial.print(dt1[1].toInt());
Serial.print("\n");
Serial.print("data 3 : ");
Serial.print(dt1[2].toInt());
Serial.print("\n");
Serial.print("data 4 : ");
Serial.print(dt1[3].toInt());
Serial.print("\n");
Serial.print("data 5 : ");
Serial.print(dt1[4].toInt());
Serial.print("\n");
Serial.print("data 6 : ");
Serial.print(dt1[5].toInt());
Serial.print("\n");
Serial.print("data 7 : ");
Serial.print(dt1[6].toInt());
Serial.print("\n");
Serial.print("data 8 : ");
Serial.print(dt1[7].toInt());
Serial.print("\n");
Serial.print("data 9 : ");
Serial.print(dt1[8].toInt());
Serial.print("\n");
Serial.print("data 10 : ");
Serial.print(dt1[9].toInt());
Serial.print("\n\n");
EEPROM.write(addr_PE1, dt1[0].toInt());
EEPROM.write(addr_PE2, dt1[1].toInt());
EEPROM.write(addr_PE3, dt1[2].toInt());
EEPROM.write(addr_PE4, dt1[3].toInt());
EEPROM.write(addr_PE5, dt1[4].toInt());
EEPROM.write(addr_PE6, dt1[5].toInt());
EEPROM.write(addr_PE7, dt1[6].toInt());
EEPROM.write(addr_PE8, dt1[7].toInt());
EEPROM.write(addr_PE9, dt1[8].toInt());
EEPROM.write(addr_PE10, dt1[9].toInt());
EEPROM.write(addr_rata2_PE, rata2tinggi_lompat);
delay(100);
}

void parsingData2()
{
    int j=0;

    Serial.print("data masuk : ");
    Serial.print(dataIn2);
    Serial.print("\n");

    dt2[j]="";
    for(i=0;i<dataIn2.length();i++)
    {
        if ((dataIn2[i] == '#') || (dataIn2[i] == ','))
        {
            j++;
            dt2[j]="";
        }
        else
        {
            dt2[j] = dt2[j] + dataIn2[i];
        }
    }
    Serial.print("data 1 : ");
    Serial.print(dt2[0].toInt());

```

```

Serial.print("data 2 : ");
Serial.print(dt2[1].toInt());
Serial.print("data 3 : ");
Serial.print(dt2[2].toInt());
Serial.print("data 4 : ");
Serial.print(dt2[3].toInt());
Serial.print("data 5 : ");
Serial.print(dt2[4].toInt());
Serial.print("\n\n");
EEPROM.write(addr_VJ, dt2[0].toInt());
EEPROM.write(addr_kiriman_client1_depan_koma, dt2[1].toInt());
EEPROM.write(addr_kiriman_client1_belakang_koma, dt2[2].toInt());
delay(100);
}

void parsingDataHorizontalJump()
{
  int j=0;

  Serial.print("data masuk : ");
  Serial.print(dataIn3);
  Serial.print("\n");

  dt3[j]="";
  for(i=0;i<dataIn3.length();i++)
  {
    if ((dataIn3[i] == '#') || (dataIn3[i] == ','))
    {
      j++;
      dt3[j]="";
    }
    else
    {
      dt3[j] = dt3[j] + dataIn3[i];
    }
  }
  Serial.print("data 1 : ");
  Serial.print(dt3[0].toInt());
  Serial.print("data 2 : ");
  Serial.print(dt3[1].toInt());
  Serial.print("data 3 : ");
  Serial.print(dt3[2].toInt());
  Serial.print("data 4 : ");
  Serial.print(dt3[3].toInt());
  Serial.print("data 5 : ");
  Serial.print(dt3[4].toInt());
  Serial.print("\n\n");
  EEPROM.write(addr_HJ, dt3[0].toInt());
  EEPROM.write(addr_HJ_plus, dt3[1].toInt());
  EEPROM.write(addr_HJ_belakang_koma, dt3[2].toInt());
  EEPROM.write(addr_kiriman_client2_depan_koma, dt3[3].toInt());
  EEPROM.write(addr_kiriman_client2_belakang_koma, dt3[4].toInt());
  delay(100);
}

void rumus_nilai_terbesar_PE()
{
  int
data[]={dt1[0].toInt(),dt1[1].toInt(),dt1[2].toInt(),dt1[3].toInt(),dt1[4].toInt
(),dt1[5].toInt(),dt1[6].toInt(),dt1[7].toInt(),dt1[8].toInt(),dt1[9].toInt()};
  int i;
  Serial.print("NILAI TERBESAR");
  Serial.println("");
  for(i=0;i<10;i++)
  {
    Serial.print(data[i]);
  }
  Serial.println(" ");
  terbesar=data[0];
  for(i=0;i<10;i++)
  {
    if(data[i]>terbesar)

```

```

    {
        terbesar=data[i];
    }
}
EEPROM.write(addr_max_PE, terbesar);
Serial.println(terbesar);
delay(50);
}

void rumus_nilai_terkecil_PE()
{
    int
data[]={dt1[0].toInt(),dt1[1].toInt(),dt1[2].toInt(),dt1[3].toInt(),dt1[4].toInt
(),dt1[5].toInt(),dt1[6].toInt(),dt1[7].toInt(),dt1[8].toInt(),dt1[9].toInt()};//
/,dt[10].toInt(),dt[11].toInt()});
    int i;
    Serial.print("NILAI TERKECIL");
    Serial.println("");
    for(i=0;i<10;i++)
    {
        Serial.print(data[i]);
    }
    Serial.println(" ");
    terkecil=data[0];
    for(i=0;i<10;i++)
    {
        if(data[i]<terkecil)
        {
            terkecil=data[i];
        }
    }
    EEPROM.write(addr_min_PE, terkecil);
    Serial.println(terkecil);
    delay(50);
}

void Proximity()
{
    prox = analogRead(proximity);
    hasil_proximity = map(prox,0,1023,0,255);
}

void tampil_proxy()
{
    Proximity();
    lcd_ziky.setCursor(0,3);
    lcd_ziky.print("Prox:");
    lcd_ziky.print(hasil_proximity);
    lcd_ziky.print("    ");
}

void rumus_powerDankecepatan_VJ()
{
    parsing_data2();
    read_eeprom_VJ=EEPROM.read(addr_VJ);

    read_eeprom_kiriman_client1_depan_koma=EEPROM.read(addr_kiriman_client1_depan_koma);

    read_eeprom_kiriman_client1_belakang_koma=EEPROM.read(addr_kiriman_client1_belakang_koma);
    simpan_BB_koma();
    simpan_TB_koma();

    waktu_dijumlah_VJ=(read_eeprom_kiriman_client1_depan_koma*100)+read_eeprom_kiriman_client1_belakang_koma;
    waktu_fix_VJ=waktu_dijumlah_VJ/100;
    kecepatan_fix_VJ=read_eeprom_VJ/waktu_fix_VJ;

    v1=read_eeprom_VJ;
    v2=BB_fix;
    v3=waktu_fix_VJ;
}

```

```

    teta_power2_VJ = v1*v2;
    teta_power3_VJ = teta_power2_VJ/v3;
    power_fix_VJ = teta_power3_VJ/100;
    horse_power_VJ= power_fix_VJ*0.013150934274927;
}

void rumus_powerDankecepatan_HJ()
{
    parsing_data_horizontal_jump();
    read_eeprom_HJ=EEPROM.read(addr_HJ);
    read_eeprom_HJ_plus=EEPROM.read(addr_HJ_plus);
    read_eeprom_HJ_belakang_koma=EEPROM.read(addr_HJ_belakang_koma);

    read_eeprom_kiriman_client2_depan_koma=EEPROM.read(addr_kiriman_client2_depan_koma);

    read_eeprom_kiriman_client2_belakang_koma=EEPROM.read(addr_kiriman_client2_belakang_koma);
    simpan_TB_koma();
    simpan_BB_koma();

    jarak_HJ=((read_eeprom_HJ+read_eeprom_HJ_plus)*100)+read_eeprom_HJ_belakang_koma;
    jarak_HJ_fix=jarak_HJ/100;

    waktu_dijumlah_HJ=(read_eeprom_kiriman_client2_depan_koma*100)+read_eeprom_kiriman_client2_belakang_koma;
    waktu_fix_HJ=waktu_dijumlah_HJ/100;
    kecepatan_fix_HJ=jarak_HJ_fix/waktu_fix_HJ;
    h1 = jarak_HJ_fix;
    h2 = BB_fix;
    h3 = waktu_fix_HJ;
    teta_power2_HJ = h1*h2;
    teta_power3_HJ = teta_power2_HJ/h3;
    power_fix_HJ = teta_power3_HJ/100;
    horse_power_HJ= power_fix_HJ*0.013150934274927;
}

void rumus_powerDankecepatan_WBR()
{
    waktu=sw_micros.elapsed();
    teta_waktu=waktu/1000000;
}

void timer()
{
    waktu=sw_micros.elapsed();
    teta_waktu=waktu/1000000;
}

void berat_badan()
{
    timbangan=((scale.get_units()*-0.1)+0.5);
    // if(timbangan<15)timbangan=0;
    scale.power_down();
    delay(30);
    scale.power_up();
}

void tampil_Bb()
{
    berat_badan();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("B.B :");
    lcd_ziky.print(timbangan,1);
    lcd_ziky.print("Kg ");
}

void deteksi1()
{
    digitalWrite(triger1, LOW); //memberi sebentar pulsa LOW ke pin untuk memastikan pulsa HIGH yang bersih
}

```

```

    delayMicroseconds(2);
    digitalWrite(triger1, HIGH);
    delayMicroseconds(10);
    digitalWrite(triger1, LOW);
    delayMicroseconds(10);
    jarakD1 = pulseIn(echo1, HIGH); //pulsa HIGH (jarak) yang berupa waktu yang
berasal dari penerima ke objek
    jarakD1 = jarakD1/58; //rumus untuk mengolah jarak
    Serial.print("DETEKSI 1:");
    Serial.print(jarakD1);
    Serial.print("\t");
    delay(10);
}
void deteksi2()
{
    digitalWrite(triger2, LOW); //memberi sebentar pulsa LOW ke pin untuk
memastikan pulsa HIGH yang bersih
    delayMicroseconds(2);
    digitalWrite(triger2, HIGH);
    delayMicroseconds(10);
    digitalWrite(triger2, LOW);
    delayMicroseconds(10);
    jarakD2 = pulseIn(echo2, HIGH); //pulsa HIGH (jarak) yang berupa waktu yang
berasal dari penerima ke objek
    jarakD2 = jarakD2/58; //rumus untuk mengolah jarak
    Serial.print("DETEKSI 2:");
    Serial.print(jarakD2);
    Serial.print("\t");
    delay(10);
}
void deteksi3()
{
    digitalWrite(triger3, LOW); //memberi sebentar pulsa LOW ke pin untuk
memastikan pulsa HIGH yang bersih
    delayMicroseconds(2);
    digitalWrite(triger3, HIGH);
    delayMicroseconds(10);
    digitalWrite(triger3, LOW);
    delayMicroseconds(10);
    jarakD3 = pulseIn(echo3, HIGH); //pulsa HIGH (jarak) yang berupa waktu yang
berasal dari penerima ke objek
    jarakD3 = jarakD3/58; //rumus untuk mengolah jarak
    Serial.print("DETEKSI 3:");
    Serial.print(jarakD3);
    Serial.println("\t");
    delay(10);
}

void coba_motor()
{
    kalibrasi_button();
    if(button_up)
    {
        motor_naik();
    }
    else if(button_down)
    {
        motor_turun();
    }
    else
    {
        motor_stop();
    }
}
/*      val_atas=digitalRead(limit_atas);
        if(val_atas==LOW)
        {
            Serial.println("atas");
        }
        val_bawah=digitalRead(limit_bawah);
        if(val_bawah==LOW)
        {
            Serial.println("bawah");
        }
*/

```

```

        }
        else
        {
            Serial.println("----");
        }*/
    }

void motor_turun()
{
    digitalWrite(dir_motor, LOW);
    analogWrite(pwm_motor, 80);
}

void motor_naik()
{
    digitalWrite(dir_motor, HIGH);
    analogWrite(pwm_motor, 30);
}

void motor_stop()
{
    digitalWrite(dir_motor, LOW);
    analogWrite(pwm_motor, LOW);
}

void updateEncoder()
{
    int MSB = digitalRead(encoderPin1); //MSB = most significant bit
    int LSB = digitalRead(encoderPin2); //LSB = least significant bit

    int encoded = (MSB << 1) | LSB; //converting the 2 pin value to single number
    int sum = (lastEncoded << 2) | encoded; //adding it to the previous encoded
    value

    if(sum == 0b1101) encoderValue++;
    if(sum == 0b1110) encoderValue--;

    lastEncoded = encoded; //store this value for next time
}

void tampil_encoder()
{
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print("encoder :");
    lcd_ziky.print(encoderValue);
    lcd_ziky.print("  ");
}

void kalibrasi_Sensor_Jarak()
{
    Analog_read_ADC_Jarak = analogRead(sensorPin);
    Hasil_read_ADC_Jarak = map(Analog_read_ADC_Jarak,0,1023,0,255);
}

void Ultrasonik()
{
    digitalWrite(triger, LOW); //memberi sebentar pulsa LOW ke pin untuk
    memastikan pulsa HIGH yang bersih
    delayMicroseconds(2);
    digitalWrite(triger, HIGH);
    delayMicroseconds(10);
    digitalWrite(triger, LOW);
    delayMicroseconds(10);
    jarak_fix_TB = pulseIn(echo, HIGH); //pulsa HIGH (jarak) yang berupa waktu
    yang berasal dari penerima ke objek
    jarak_fix_TB = jarak_fix_TB/58; //rumus untuk mengolah jarak
    // Serial.println(jarak_fix);
    delay(10);
}

void tinggi_badan()
{

```

```

Ultrasonik();
TB=tiang2-jarak_fix_TB; //rumus menghitung tinggi badan
delay(50);
}

void tampil_TB1()
{
  lcd_ziky.setCursor(0,1);
  lcd_ziky.print("T.B1:");
  lcd_ziky.print(encoderValue*0.0560);//konfersi dari data encoder ke jarak
  sesungguhnya
  lcd_ziky.print("Cm   ");
  Serial.println(encoderValue*0.0560);
}

void tampil_TB2()
{
  tinggi_badan();
  lcd_ziky.setCursor(0,2);
  lcd_ziky.print("T.B2:");
  lcd_ziky.print((TB),1);
  lcd_ziky.print("Cm   ");
}

void vertikal_jump()
{
  mulai_lagi:
  while(1)
  {
    kalibrasi_button();
    if(jenis_kelamin==1)
    {
      lcd_ziky.setCursor(0,1);
      lcd_ziky.print(" Laki-laki");
      if(umur==1){lcd_ziky.setCursor(11,1);lcd_ziky.print("9 th ");}
      if(umur==2){lcd_ziky.setCursor(11,1);lcd_ziky.print("10 th ");}
      if(umur==3){lcd_ziky.setCursor(11,1);lcd_ziky.print("11 th ");}
      if(umur==4){lcd_ziky.setCursor(11,1);lcd_ziky.print("12 th ");}
      if(umur==5){lcd_ziky.setCursor(11,1);lcd_ziky.print("13 th ");}
      if(umur==6){lcd_ziky.setCursor(11,1);lcd_ziky.print("14 th ");}
      if(umur==7){lcd_ziky.setCursor(11,1);lcd_ziky.print("15 th ");}
      if(umur==8){lcd_ziky.setCursor(11,1);lcd_ziky.print("16 th");}
      if(umur==9){lcd_ziky.setCursor(11,1);lcd_ziky.print("17 th");}
      if(umur==10){lcd_ziky.setCursor(11,1);lcd_ziky.print(">18 th");}
    }
    else if(jenis_kelamin==2)
    {
      lcd_ziky.setCursor(0,1);
      lcd_ziky.print(" Perempuan");
      if(umur==1){lcd_ziky.setCursor(11,1);lcd_ziky.print("9 th ");}
      if(umur==2){lcd_ziky.setCursor(11,1);lcd_ziky.print("10 th ");}
      if(umur==3){lcd_ziky.setCursor(11,1);lcd_ziky.print("11 th ");}
      if(umur==4){lcd_ziky.setCursor(11,1);lcd_ziky.print("12 th ");}
      if(umur==5){lcd_ziky.setCursor(11,1);lcd_ziky.print("13 th ");}
      if(umur==6){lcd_ziky.setCursor(11,1);lcd_ziky.print("14 th ");}
      if(umur==7){lcd_ziky.setCursor(11,1);lcd_ziky.print("15 th ");}
      if(umur==8){lcd_ziky.setCursor(11,1);lcd_ziky.print("16 th");}
      if(umur==9){lcd_ziky.setCursor(11,1);lcd_ziky.print("17 th");}
      if(umur==10){lcd_ziky.setCursor(11,1);lcd_ziky.print(">18 th");}
    }
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("  VERTIKAL JUMP");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print("Tekan <OK> utk mulai");
    lcd_ziky.setCursor(0,3);
    lcd_ziky.print("Tekan <BACK> utk bck");
    if(Serial1.available(>0)
    {
      char a=Serial1.read();
      dataIn_kal_VJ += a;
      EEPROM.write(addr_kal, dataIn_kal_VJ.toInt());
    }
  }
}

```

```

read_eeeprom_kal=EEPROM.read(addr_kal);
// lcd_ziky.setCursor(10,3);
// lcd_ziky.print(read_eeeprom_kal);
if(button_up)
{
  lcd_ziky.clear();
  delay(200);
  while(1)
  {
    kalibrasi_button();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("Kalibrasi");
    Serial1.println(3);
    for(int x=0;x<20;x++)
    {
      lcd_ziky.setCursor(x,1);
      lcd_ziky.write(2);
      lcd_ziky.display();
      delay(80);
    }
    lcd_ziky.clear();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("Kalibrasi");
    for(int x=0;x<20;x++)
    {
      lcd_ziky.setCursor(x,1);
      lcd_ziky.write(2);
      lcd_ziky.display();
      delay(100);
    }
    for(int z=0;z<30;z++)
    {
      Serial1.println(4);
      delay(10);
    }
    lcd_ziky.clear();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("Selesai...");
    delay(200);
    lcd_ziky.clear();
    break;
  }
}
if(button_ok)
{
  Serial1.println(5);
  Serial1.println(9);
  lcd_ziky.clear();
  goto eksekusi;
}
if(button_BACK)
{
  lcd_ziky.clear();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Selesai...");
  delay(300);
  lcd_ziky.clear();
  goto akhir;
}
}
eksekusi:
while(1)
{
  kalibrasi_button();
  rumus_powerDankecepatan_VJ();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("T.lom :");
  lcd_ziky.print(read_eeeprom_VJ);
  lcd_ziky.print(" Cm ");
  lcd_ziky.setCursor(0,1);
  lcd_ziky.print("Kec :");
  lcd_ziky.print(kecepatan_fix_VJ);
}

```

```

lcd_ziky.print(" Cm/s ");
lcd_ziky.setCursor(0,2);
lcd_ziky.print("Power :");
lcd_ziky.print(power_fix_VJ);
lcd_ziky.print(" Kgm/s");
lcd_ziky.setCursor(0,3);
lcd_ziky.print("      :");
lcd_ziky.print(horse_power_VJ);
lcd_ziky.print(" Hp      ");
if(button_BACK)
{
  lcd_ziky.clear();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Kirim data ke web...");
  paket_data_nodeMcu();
  lcd_ziky.clear();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Selesai...");
  waktu_fix_VJ=0;
  kecepatan_fix_VJ=0;
  for(int reset_data_VJ=0;reset_data_VJ<10;reset_data_VJ++)
  {
    dt2[reset_data_VJ]="";
    delay(20);
  }
  lcd_ziky.clear();
  goto mulai_lagi;
}
delay(2000);
lcd_ziky.clear();
goto gantian1;
}
gantian1:
while(1)
{
  kalibrasi_button();
  rumus_powerDankecepatan_VJ();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("_____");
  for(int a=0;a<20;a++){lcd_ziky.setCursor(a,2);lcd_ziky.write(3);}
  if(jenis_kelamin==1)
  {
    if(umur==1)
    {
      if(read_eeprom_VJ>41){istimewa();}
      else if(read_eeprom_VJ>36 && read_eeprom_VJ<=41){baik_sekali();}
      else if(read_eeprom_VJ>28 && read_eeprom_VJ<=36){baik();}
      else if(read_eeprom_VJ>23 && read_eeprom_VJ<=28){cukup();}
      else if(read_eeprom_VJ>10 && read_eeprom_VJ<=23){kurang();}
      else if(read_eeprom_VJ<=10){kurang_sekali();}
    }
    if(umur==2)
    {
      if(read_eeprom_VJ>41){istimewa();}
      else if(read_eeprom_VJ>36 && read_eeprom_VJ<=41){baik_sekali();}
      else if(read_eeprom_VJ>28 && read_eeprom_VJ<=36){baik();}
      else if(read_eeprom_VJ>23 && read_eeprom_VJ<=28){cukup();}
      else if(read_eeprom_VJ>10 && read_eeprom_VJ<=23){kurang();}
      else if(read_eeprom_VJ<=10){kurang_sekali();}
    }
    if(umur==3)
    {
      if(read_eeprom_VJ>41){istimewa();}
      else if(read_eeprom_VJ>36 && read_eeprom_VJ<=41){baik_sekali();}
      else if(read_eeprom_VJ>28 && read_eeprom_VJ<=36){baik();}
      else if(read_eeprom_VJ>23 && read_eeprom_VJ<=28){cukup();}
      else if(read_eeprom_VJ>10 && read_eeprom_VJ<=23){kurang();}
      else if(read_eeprom_VJ<=10){kurang_sekali();}
    }
    if(umur==4)
    {
      if(read_eeprom_VJ>50){istimewa();}
    }
  }
}

```

```

else if(read_eeprom_VJ>43 && read_eeprom_VJ<=50){baik_sekali();}
else if(read_eeprom_VJ>36 && read_eeprom_VJ<=43){baik();}
else if(read_eeprom_VJ>28 && read_eeprom_VJ<=36){cukup();}
else if(read_eeprom_VJ>13 && read_eeprom_VJ<=28){kurang();}
else if(read_eeprom_VJ<=13){kurang_sekali();}
}
if (umur==5)
{
if(read_eeprom_VJ>50){istimewa();}
else if(read_eeprom_VJ>43 && read_eeprom_VJ<=50){baik_sekali();}
else if(read_eeprom_VJ>36 && read_eeprom_VJ<=43){baik();}
else if(read_eeprom_VJ>28 && read_eeprom_VJ<=36){cukup();}
else if(read_eeprom_VJ>13 && read_eeprom_VJ<=28){kurang();}
else if(read_eeprom_VJ<=13){kurang_sekali();}
}
if (umur==6)
{
if(read_eeprom_VJ>50){istimewa();}
else if(read_eeprom_VJ>43 && read_eeprom_VJ<=50){baik_sekali();}
else if(read_eeprom_VJ>36 && read_eeprom_VJ<=43){baik();}
else if(read_eeprom_VJ>28 && read_eeprom_VJ<=36){cukup();}
else if(read_eeprom_VJ>13 && read_eeprom_VJ<=28){kurang();}
else if(read_eeprom_VJ<=13){kurang_sekali();}
}
if (umur==7)
{
if(read_eeprom_VJ>63){istimewa();}
else if(read_eeprom_VJ>58 && read_eeprom_VJ<=63){baik_sekali();}
else if(read_eeprom_VJ>48 && read_eeprom_VJ<=58){baik();}
else if(read_eeprom_VJ>30 && read_eeprom_VJ<=48){cukup();}
else if(read_eeprom_VJ>13 && read_eeprom_VJ<=30){kurang();}
else if(read_eeprom_VJ<=13){kurang_sekali();}
}
if (umur==8)
{
if(read_eeprom_VJ>63){istimewa();}
else if(read_eeprom_VJ>58 && read_eeprom_VJ<=63){baik_sekali();}
else if(read_eeprom_VJ>48 && read_eeprom_VJ<=58){baik();}
else if(read_eeprom_VJ>30 && read_eeprom_VJ<=48){cukup();}
else if(read_eeprom_VJ>13 && read_eeprom_VJ<=30){kurang();}
else if(read_eeprom_VJ<=13){kurang_sekali();}
}
if (umur==9)
{
if(read_eeprom_VJ>63){istimewa();}
else if(read_eeprom_VJ>58 && read_eeprom_VJ<=63){baik_sekali();}
else if(read_eeprom_VJ>48 && read_eeprom_VJ<=58){baik();}
else if(read_eeprom_VJ>30 && read_eeprom_VJ<=48){cukup();}
else if(read_eeprom_VJ>13 && read_eeprom_VJ<=30){kurang();}
else if(read_eeprom_VJ<=13){kurang_sekali();}
}
if (umur==10)
{
if(read_eeprom_VJ>66){istimewa();}
else if(read_eeprom_VJ>61 && read_eeprom_VJ<=66){baik_sekali();}
else if(read_eeprom_VJ>48 && read_eeprom_VJ<=61){baik();}
else if(read_eeprom_VJ>33 && read_eeprom_VJ<=48){cukup();}
else if(read_eeprom_VJ>20 && read_eeprom_VJ<=33){kurang();}
else if(read_eeprom_VJ<=20){kurang_sekali();}
}
}

if(jenis_kelamin==2)
{
if (umur==1)
{
if(read_eeprom_VJ>41){istimewa();}
else if(read_eeprom_VJ>36 && read_eeprom_VJ<=41){baik_sekali();}
else if(read_eeprom_VJ>28 && read_eeprom_VJ<=36){baik();}
else if(read_eeprom_VJ>23 && read_eeprom_VJ<=28){cukup();}
else if(read_eeprom_VJ>10 && read_eeprom_VJ<=23){kurang();}
else if(read_eeprom_VJ<=10){kurang_sekali();}
}
}

```



```

    }
    if (umur==10)
    {
        if (read_eeprom_VJ>66){istimewa();}
        else if (read_eeprom_VJ>61 && read_eeprom_VJ<=66){baik_sekali();}
        else if (read_eeprom_VJ>48 && read_eeprom_VJ<=61){baik();}
        else if (read_eeprom_VJ>33 && read_eeprom_VJ<=48){cukup();}
        else if (read_eeprom_VJ>20 && read_eeprom_VJ<=33){kurang();}
        else if (read_eeprom_VJ<=20){kurang_sekali();}
    }
}

if (button_BACK)
{
    lcd_ziky.clear();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("Kirim data ke web...");
    paket_data_nodeMcu();
    lcd_ziky.clear();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("Selesai...");
    waktu_fix_VJ=0;
    kecepatan_fix_VJ=0;
    for (int reset_data_VJ=0; reset_data_VJ<10; reset_data_VJ++)
    {
        dt2[reset_data_VJ]="";
        delay(20);
    }
    lcd_ziky.clear();
    goto mulai_lagi;
}
delay(1000);
lcd_ziky.clear();
goto eksekusi;
}
akhir:
while(1){break;}
}

void vertikal_jump_otomatis()
{
    mulai_lagi:
    while(1)
    {
        zios_servo.write(70);
        kalibrasi_button();
        deteksi1();
        deteksi2();
        deteksi3();
        if (jenis_kelamin==1)
        {
            lcd_ziky.setCursor(0,0);
            lcd_ziky.print(" VERTIKAL JUMP [L]");
        }
        else if (jenis_kelamin==2)
        {
            lcd_ziky.setCursor(0,0);
            lcd_ziky.print(" VERTIKAL JUMP [P]");
        }
        if (jarakD1<=40&&jarakD2<=40&&jarakD3<=40)
        {
            lcd_ziky.setCursor(0,1);
            lcd_ziky.print("          SUDAH SIAP          ");
            lcd_ziky.setCursor(0,2);
            lcd_ziky.print("          mulai...          ");
            lcd_ziky.setCursor(0,3);
            lcd_ziky.print("Tekan <BACK> utk bck");
            encoderValue=0;
            delay(4000);
            lcd_ziky.clear();
            goto scanTBBB;
        }
    }
}

```

```

if(button_BACK)
{
  lcd_ziky.clear();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Selesai...");
  delay(300);
  lcd_ziky.clear();
  goto akhir;
}
else if(jarakD1>40&&jarakD2>40&&jarakD3>40)
{
  lcd_ziky.setCursor(0,1);
  lcd_ziky.print("      BELUM SIAP      ");
  lcd_ziky.setCursor(0,2);
  lcd_ziky.print("Belum Naik Timbangan");
  lcd_ziky.setCursor(0,3);
  lcd_ziky.print("Tekan <BACK> utk bck");
}
}

//=====
scantBBB:
while(1)
{
  motor_turun();
  TB_enc=tiang1-encoderValue*0.0558;
  pembagi_TB=TB_enc;
  hasil_bagi_TB=TB_enc-pembagi_TB;
  hasil_belakang_koma_TB=hasil_bagi_TB*10;
  lcd_ziky.setCursor(0, 0);
  lcd_ziky.print("Scan.....");
  lcd_ziky.setCursor(0, 1);
  lcd_ziky.print("Tinggi :");
  lcd_ziky.print(TB_enc,1);
  Serial.println(TB_enc,1);
  lcd_ziky.setCursor(0, 2);
  lcd_ziky.print("Berat  :0.0");
  val_kepala=digitalRead(limit_kepala);
  if(val_kepala==LOW)
  {
    digitalWrite(buzzer,HIGH);
    EEPROM.write(addr_TB, TB_enc);
    EEPROM.write(addr_TB_belakang_koma, hasil_belakang_koma_TB);
    delay(50);
    digitalWrite(buzzer,LOW);
    motor_naik();
    delay(700);
    goto berat_scan;
  }
  val_bawah=digitalRead(limit_bawah);
  if(val_bawah==LOW)
  {
    digitalWrite(buzzer,HIGH);
    EEPROM.write(addr_TB, TB_enc);
    EEPROM.write(addr_TB_belakang_koma, hasil_belakang_koma_TB);
    motor_stop();
    delay(200);
    digitalWrite(buzzer,LOW);
    goto pengukuran_bawah;
  }
}
berat_scan:
while(1)
{
  motor_stop();
  timer();
  sw_millis.start();
  sw_micros.start();
  sw_secs.start();
  berat_badan();
  simpan_TB_koma();
  lcd_ziky.setCursor(0, 0);

```

```

lcd_ziky.print("Scan.....");
lcd_ziky.setCursor(0, 1);
lcd_ziky.print("Tinggi :");
lcd_ziky.print(TB_fix);
lcd_ziky.print(" ");
lcd_ziky.setCursor(0, 2);
lcd_ziky.print("Berat :");
lcd_ziky.print(timbangan,1);
lcd_ziky.print(" ");
if(teta_waktu>5)
{
  sw_millis.stop();
  sw_micros.stop();
  sw_secs.stop();
  pembagi_BB=timbangan;
  hasil_bagi_BB=timbangan-pembagi_BB;
  hasil_belakang_koma_BB=hasil_bagi_BB*10;
  EEPROM.write(addr_BB, timbangan);
  EEPROM.write(addr_BB_belakang_koma, hasil_belakang_koma_BB);
  for(int a=0; a<5;a++)
  {
    lcd_ziky.setCursor(0, 0);
    lcd_ziky.print("Selesai...");
    lcd_ziky.setCursor(0, 1);
    lcd_ziky.print("Tinggi :");
    lcd_ziky.print(TB_fix);
    lcd_ziky.print("Cm");
    lcd_ziky.setCursor(0, 2);
    lcd_ziky.print("Berat :");
    lcd_ziky.print(timbangan,1);
    lcd_ziky.print("Kg");
    digitalWrite(buzzer,HIGH);
    delay(200);
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("                ");
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print("                ");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print("                ");
    digitalWrite(buzzer,LOW);
    delay(100);
  }
  lcd_ziky.clear();
  goto selesai;
}
}
pengukuran_bawah:
while(1)
{
  sw_millis.start();
  sw_micros.start();
  sw_secs.start();
  tinggi_badan();
  timer();
  berat_badan();
  tinggi_badan();
  //Simpan TB dengan koma
  pembagi_TB=TB;
  hasil_bagi_TB=TB-pembagi_TB;
  hasil_belakang_koma_TB=hasil_bagi_TB*10;
  //Simpan BB dengan koma
  pembagi_BB=timbangan;
  hasil_bagi_BB=timbangan-pembagi_BB;
  hasil_belakang_koma_BB=hasil_bagi_BB*10;
  lcd_ziky.setCursor(0, 0);
  lcd_ziky.print("Scan.....");
  lcd_ziky.setCursor(0, 1);
  lcd_ziky.print("Tinggi :");
  lcd_ziky.print(TB);
  lcd_ziky.print(" ");
  lcd_ziky.setCursor(0, 2);
  lcd_ziky.print("Berat :");

```

```

lcd_ziky.print(timbangan,1);
lcd_ziky.print(" ");
val_kepala=digitalRead(limit_kepala);
if(val_kepala==LOW)
{
  goto selesai;
}
if(teta_waktu>5)
{
  sw_millis.stop();
  sw_micros.stop();
  sw_secs.stop();
  EEPROM.write(addr_BB, timbangan);
  EEPROM.write(addr_BB_belakang_koma, hasil_belakang_koma_BB);
  EEPROM.write(addr_TB, TB);
  EEPROM.write(addr_TB_belakang_koma, hasil_belakang_koma_TB);
  for(int a=0; a<5;a++)
  {
    lcd_ziky.setCursor(0, 0);
    lcd_ziky.print("Selesai...");
    lcd_ziky.setCursor(0, 1);
    lcd_ziky.print("Tinggi :");
    lcd_ziky.print(TB);
    lcd_ziky.print("Cm");
    lcd_ziky.setCursor(0, 2);
    lcd_ziky.print("Berat :");
    lcd_ziky.print(timbangan,1);
    lcd_ziky.print("Kg");
    digitalWrite(buzzer,HIGH);
    delay(200);
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("                ");
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print("                ");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print("                ");
    digitalWrite(buzzer,LOW);
    delay(100);
  }
  goto selesai;
}
}
selesai:
while(1)
{
  simpan_TB_koma();
  simpan_BB_koma();
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("Simpan...");
  motor_naik();
  Serial.print("TB=");
  Serial.println(TB_fix);
  Serial.print("BB=");
  Serial.println(BB_fix);
  val_atas=digitalRead(limit_atas);
  if(val_atas==LOW)
  {
    digitalWrite(buzzer,HIGH);
    motor_stop();
    sw_millis.reset();
    sw_micros.reset();
    sw_secs.reset();
    encoderValue=0;
    zios_servo.write(115);
    delay(50);
    digitalWrite(buzzer,LOW);
    delay(5000);
    Serial1.println(5);
    Serial1.println(9);
    lcd_ziky.clear();
    goto eksekusi;
  }
}

```

```

}
eksekusi:
while(1)
{
    kalibrasi_button();
    rumus_powerDankecepatan_VJ();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("T.lom :");
    lcd_ziky.print(read_eeprom_VJ);
    lcd_ziky.print(" Cm ");
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print("Kec :");
    lcd_ziky.print(kecepatan_fix_VJ);
    lcd_ziky.print(" Cm/s ");
    lcd_ziky.setCursor(0,2);
    lcd_ziky.print("Power :");
    lcd_ziky.print(power_fix_VJ);
    lcd_ziky.print(" Kgm/s");
    lcd_ziky.setCursor(0,3);
    lcd_ziky.print(" :");
    lcd_ziky.print(horse_power_VJ);
    lcd_ziky.print(" Hp ");
    if(button_BACK)
    {
        lcd_ziky.clear();
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("Kirim data ke web...");
        paket_data_nodeMcu();
        lcd_ziky.clear();
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("Selesai...");
        waktu_fix_VJ=0;
        kecepatan_fix_VJ=0;
        for(int reset_data_VJ=0;reset_data_VJ<10;reset_data_VJ++)
        {
            dt2[reset_data_VJ]="";
            delay(20);
        }
        lcd_ziky.clear();
        goto mulai_lagi;
    }
    delay(2000);
    lcd_ziky.clear();
    goto gantian1;
}
gantian1:
while(1)
{
    kalibrasi_button();
    rumus_powerDankecepatan_VJ();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("_____");
    for(int a=0;a<20;a++){lcd_ziky.setCursor(a,2);lcd_ziky.write(3);}
    if(jenis_kelamin==1)
    {
        if(umur==1)
        {
            if(read_eeprom_VJ>41){istimewa();}
            else if(read_eeprom_VJ>36 && read_eeprom_VJ<=41){baik_sekali();}
            else if(read_eeprom_VJ>28 && read_eeprom_VJ<=36){baik();}
            else if(read_eeprom_VJ>23 && read_eeprom_VJ<=28){cukup();}
            else if(read_eeprom_VJ>10 && read_eeprom_VJ<=23){kurang();}
            else if(read_eeprom_VJ<=10){kurang_sekali();}
        }
        if(umur==2)
        {
            if(read_eeprom_VJ>41){istimewa();}
            else if(read_eeprom_VJ>36 && read_eeprom_VJ<=41){baik_sekali();}
            else if(read_eeprom_VJ>28 && read_eeprom_VJ<=36){baik();}
            else if(read_eeprom_VJ>23 && read_eeprom_VJ<=28){cukup();}
            else if(read_eeprom_VJ>10 && read_eeprom_VJ<=23){kurang();}
            else if(read_eeprom_VJ<=10){kurang_sekali();}
        }
    }
}

```



```

else if(read_eeprom_VJ>33 && read_eeprom_VJ<=38){baik();}
else if(read_eeprom_VJ>20 && read_eeprom_VJ<=33){cukup();}
else if(read_eeprom_VJ>8 && read_eeprom_VJ<=20){kurang();}
else if(read_eeprom_VJ<=8){kurang_sekali();}
}
if (umur==9)
{
if(read_eeprom_VJ>43){istimewa();}
else if(read_eeprom_VJ>38 && read_eeprom_VJ<=43){baik_sekali();}
else if(read_eeprom_VJ>33 && read_eeprom_VJ<=38){baik();}
else if(read_eeprom_VJ>20 && read_eeprom_VJ<=33){cukup();}
else if(read_eeprom_VJ>8 && read_eeprom_VJ<=20){kurang();}
else if(read_eeprom_VJ<=8){kurang_sekali();}
}
if (umur==10)
{
if(read_eeprom_VJ>66){istimewa();}
else if(read_eeprom_VJ>61 && read_eeprom_VJ<=66){baik_sekali();}
else if(read_eeprom_VJ>48 && read_eeprom_VJ<=61){baik();}
else if(read_eeprom_VJ>33 && read_eeprom_VJ<=48){cukup();}
else if(read_eeprom_VJ>20 && read_eeprom_VJ<=33){kurang();}
else if(read_eeprom_VJ<=20){kurang_sekali();}
}
}

if(button_BACK)
{
lcd_ziky.clear();
lcd_ziky.setCursor(0,0);
lcd_ziky.print("Kirim data ke web...");
paket_data_nodeMcu();
lcd_ziky.clear();
lcd_ziky.setCursor(0,0);
lcd_ziky.print("Selesai...");
waktu_fix_VJ=0;
kecepatan_fix_VJ=0;
for(int reset_data_VJ=0;reset_data_VJ<10;reset_data_VJ++)
{
dt2[reset_data_VJ]="";
delay(20);
}
lcd_ziky.clear();
goto mulai_lagi;
}
delay(1000);
lcd_ziky.clear();
goto eksekusi;
}
akhir:
while(1){break;}
}

```

Lampiran 3. Program *Slave 1 Controller* (Arduino NANO 1)

```

#include <EEPROM.h>
#include <StopWatch.h>
#include <SoftwareSerial.h>
#define naik hasil_proximity > 100
SoftwareSerial kirim_server(9,10); // RX, TX
StopWatch sw_millis; // MILLIS (default)
StopWatch sw_micros(StopWatch::MICROS);
StopWatch sw_secs(StopWatch::SECONDS);
#define echo 7
#define triger 8

const int sjarak=A0;
const int proximity=A1;
const int buzzer=2;
const int led_merah=4;
const int led_hijau=5;

float sensor_jarak, Hasil_read_ADC_Jarak;
float Analog_read_ADC_Jarak, hasil_sensor_jarak, hasil_jarak,
sampling_data_jarak;
unsigned int jarak=0;
int jarak_fix=0;
int prox, hasil_proximity;

int b;
int terbesar[15];
int terbesar_Vjump;
unsigned int tinggi_Vjump[100];
unsigned int tinggi1[10];
unsigned int tinggi2[10];
unsigned int tinggi3[10];
unsigned int tinggi4[10];
unsigned int tinggi5[10];
unsigned int tinggi6[10];
unsigned int tinggi7[10];
unsigned int tinggi8[10];
unsigned int tinggi9[10];
unsigned int tinggi10[10];
unsigned int counter=0;

int erata2tinggi1=1;
int erata2tinggi2=2;
int erata2tinggi3=3;
int erata2tinggi4=4;
int erata2tinggi5=5;
int erata2tinggi6=6;
int erata2tinggi7=7;
int erata2tinggi8=8;
int erata2tinggi9=9;
int erata2tinggi10=10;
int ekalibrasi=16;
int erata2tinggi_Vjump=17;
int ewaktu_depan_koma=18;
int ewaktu_belakang_koma=19;

int rata2tinggi1=0;
int rata2tinggi2=0;
int rata2tinggi3=0;
int rata2tinggi4=0;
int rata2tinggi5=0;
int rata2tinggi6=0;
int rata2tinggi7=0;
int rata2tinggi8=0;
int rata2tinggi9=0;
int rata2tinggi10=0;
int kalibrasi=0;
int rata2tinggi_Vjump=0;
int waktu_depan_koma=0;

```

```

int waktu_belakang_koma=0;

float waktu=0;
float teta_waktu=0;
int pembagi_waktu;
float hasil_bagi_waktu;
float hasil_belakang_koma;

void timer()
{
    waktu=sw_micros.elapsed();
    teta_waktu=waktu/1000000;
    pembagi_waktu=teta_waktu;
    hasil_bagi_waktu=teta_waktu-pembagi_waktu;
    hasil_belakang_koma=hasil_bagi_waktu*100;
}

void ADC_Jarak()
{
    Analog_read_ADC_Jarak = analogRead(sjarak);
    Hasil_read_ADC_Jarak = map(Analog_read_ADC_Jarak,0,1023,0,255);
}

void Proximity()
{
    prox = analogRead(proximity);
    hasil_proximity = map(prox,0,1023,0,255);
}

void kalibrasi_jarak()
{
    ADC_Jarak();
    sensor_jarak = Hasil_read_ADC_Jarak/51;
    hasil_jarak = ((1/((0.04*sensor_jarak)-0.006))-0.4));
    jarak=hasil_jarak; //rumus jarak
    if(jarak>90)
    {
        jarak=90;
    }
    delay(40);
}

void Ultrasonik()
{
    digitalWrite(triger, LOW); //memberi sebentar pulsa LOW ke pin untuk
    memastikan pulsa HIGH yang bersih
    delayMicroseconds(2);
    digitalWrite(triger, HIGH);
    delayMicroseconds(10);
    digitalWrite(triger, LOW);
    delayMicroseconds(10);
    jarak_fix = pulseIn(echo, HIGH); //pulsa HIGH (jarak) yang berupa waktu yang
    berasal dari penerima ke objek
    jarak_fix = jarak_fix/58; //rumus untuk mengolah jarak
    Serial.println(jarak_fix);
    delay(5);
}

long ultrasonic_distance()
{
    long jarak_long;
    digitalWrite(triger, LOW); //memberi sebentar pulsa LOW ke pin untuk
    memastikan pulsa HIGH yang bersih
    delayMicroseconds(2);
    digitalWrite(triger, HIGH);
    delayMicroseconds(10);
    digitalWrite(triger, LOW);
    delayMicroseconds(10);
    jarak_long = pulseIn(echo, HIGH); //pulsa HIGH (jarak) yang berupa waktu yang
    berasal dari penerima ke objek
    jarak_long = jarak_long/58; //rumus untuk mengolah jarak
    delay(30);
    return jarak_long;
}

```

```

}

void setup()
{
  pinMode(echo, INPUT);
  pinMode(trigger, OUTPUT);
  pinMode(sjarak, INPUT);
  pinMode(proximity, INPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(led_merah, OUTPUT);
  pinMode(led_hijau, OUTPUT);
  Serial.begin(57600);
  kirim_server.begin(9600);
  kalibrasi_jarak();
  Ultrasonik();
  EEPROM.write(ekalibrasi, jarak_fix);
  digitalWrite(led_merah, HIGH);
  digitalWrite(led_hijau, HIGH);
  delay(1000);
  digitalWrite(led_merah, LOW);
  digitalWrite(led_hijau, LOW);
  digitalWrite(buzzer, HIGH);
  delay(50);
  digitalWrite(buzzer, LOW);
  delay(50);
  digitalWrite(led_merah, HIGH);
  digitalWrite(led_hijau, HIGH);
  digitalWrite(buzzer, HIGH);
  delay(50);
  digitalWrite(buzzer, LOW);
  delay(50);
  digitalWrite(led_merah, LOW);
  digitalWrite(led_hijau, LOW);
  digitalWrite(buzzer, HIGH);
  delay(50);
  digitalWrite(buzzer, LOW);
  delay(50);
}

void loop()
{
  // kalibrasi_jarak();
  // Serial.println(jarak);
  // Proximity();
  // Serial.println(hasil_proximity);
  // Serial.println(jarak);
  // if(jarak
  // power_endurance();
  // Vjump();
  terima_data();
  // Ultrasonik();
  // long distance=ultrasonic_distance();
  // Serial.println(distance);
}

void terima_data()
{
  kalibrasi=EEPROM.read(ekalibrasi);
  // Serial.print("NILAI KALIBRASI=");
  // Serial.println(kalibrasi);
  atas:
  if(kirim_server.available())
  {
    char data=(char)kirim_server.read();
    Serial.println(data);
    if(data=='1')
    {
      Serial.println("POWER ENDURANCE");
      power_endurance();
      goto atas;
    }
  }
  else if(data=='5')

```

```

    {
        Serial.println("Vjump");
        Vjump();
        goto atas;
    }
else if(data=='3')
{
    while(1)
    {
        if(kirim_server.available())
        {
            char kal=(char)kirim_server.read();
            digitalWrite(buzzer,HIGH);
            kalibrasi_jarak();
            EEPROM.write(ekalibrasi, jarak);
            digitalWrite(buzzer,LOW);
            delay(30);
            if(kal=='4')
            {
                kalibrasi=EEPROM.read(ekalibrasi);
                kirim_server.println(kalibrasi);
                Serial.println(kalibrasi);
                delay(10);
                goto atas;
            }
        }
    }
}
}
}

void Vjump()
{
    while(1)
    {
        Proximity();
        kalibrasi=EEPROM.read(ekalibrasi);
        int syarat=kalibrasi+15;
        digitalWrite(led_hijau, HIGH);
        if(naik)
        {
            sw_millis.start();
            sw_micros.start();
            sw_secs.start();
            digitalWrite(led_merah, HIGH);
            digitalWrite(buzzer, HIGH);
            // for(int a=1;a<=40;a++){long
            distance=ultrasonic_distance();tinggi_Vjump[a]=distance;}
            goto ambil_data;
        }
        else
        {
            digitalWrite(led_merah, LOW);
        }
    }
    ambil_data:
    while(1)
    {
        Proximity();
        long distance=ultrasonic_distance();
        int a;
        a++;
        tinggi_Vjump[a]=distance;
        if(hasil_proximity<100)
        {
            sw_millis.stop();
            sw_micros.stop();
            sw_secs.stop();
            digitalWrite(buzzer, LOW);
            digitalWrite(led_merah, LOW);
            goto berhenti;
        }
    }
}

```

```

}
berhenti:
while(1)
{
    kalibrasi=EEPROM.read(ekalibrasi);
    vertikal_jump();
    timer();
    EEPROM.write(erata2tinggi_Vjump, terbesar_Vjump);
    EEPROM.write(ewaktu_depan_koma, pembagi_waktu);
    EEPROM.write(ewaktu_belakang_koma, hasil_belakang_koma);
    delay(100);
    paket_data_vertikal_jump();
    Serial.println("Paket data sudah dikirim.");
    goto akhir;
}
akhir:
sw_millis.reset();
sw_micros.reset();
sw_secs.reset();
digitalWrite(led_hijau, LOW);
while(1){break;}
terima_data();
}

void power_endurance()
{
    while(1)
    {
        Proximity();
        digitalWrite(led_hijau, HIGH);
        if(naik)
        {
            int jeda=700;
            digitalWrite(led_merah, HIGH);
            counter++;
            Serial.println(counter);
            if(counter==1)
            {
                for(int a=1;a<=10;a++){kalibrasi_jarak();tinggi1[a]=jarak;}
                digitalWrite(buzzer, HIGH);
                delay(100);
                digitalWrite(buzzer, LOW);
                delay(jeda);
            }
            if(counter==2)
            {
                for(int a=1;a<=10;a++){kalibrasi_jarak();tinggi2[a]=jarak;}
                digitalWrite(buzzer, HIGH);
                delay(100);
                digitalWrite(buzzer, LOW);
                delay(jeda);
            }
            if(counter==3)
            {
                for(int a=1;a<=10;a++){kalibrasi_jarak();tinggi3[a]=jarak;}
                digitalWrite(buzzer, HIGH);
                delay(100);
                digitalWrite(buzzer, LOW);
                delay(jeda);
            }
            if(counter==4)
            {
                for(int a=1;a<=10;a++){kalibrasi_jarak();tinggi4[a]=jarak;}
                digitalWrite(buzzer, HIGH);
                delay(100);
                digitalWrite(buzzer, LOW);
                delay(jeda);
            }
            if(counter==5)
            {
                for(int a=1;a<=10;a++){kalibrasi_jarak();tinggi5[a]=jarak;}
                digitalWrite(buzzer, HIGH);
            }
        }
    }
}

```

```

    delay(100);
    digitalWrite(buzzer, LOW);
    delay(jeda);
}
if(counter==6)
{
    for(int a=1;a<=10;a++){kalibrasi_jarak();tinggi6[a]=jarak;}
    digitalWrite(buzzer, HIGH);
    delay(100);
    digitalWrite(buzzer, LOW);
    delay(jeda);
}
if(counter==7)
{
    for(int a=1;a<=10;a++){kalibrasi_jarak();tinggi7[a]=jarak;}
    digitalWrite(buzzer, HIGH);
    delay(100);
    digitalWrite(buzzer, LOW);
    delay(jeda);
}
if(counter==8)
{
    for(int a=1;a<=10;a++){kalibrasi_jarak();tinggi8[a]=jarak;}
    digitalWrite(buzzer, HIGH);
    delay(100);
    digitalWrite(buzzer, LOW);
    delay(jeda);
}
if(counter==9)
{
    for(int a=1;a<=10;a++){kalibrasi_jarak();tinggi9[a]=jarak;}
    digitalWrite(buzzer, HIGH);
    delay(100);
    digitalWrite(buzzer, LOW);
    delay(jeda);
}
if(counter==10)
{
    for(int a=1;a<=10;a++){kalibrasi_jarak();tinggi10[a]=jarak;}
    digitalWrite(buzzer, HIGH);
    delay(100);
    digitalWrite(buzzer, LOW);
    delay(jeda);
    goto ngisor;
}
if(counter>10)
{
    counter=11;
}
}
else{digitalWrite(led_merah, LOW);}
}
ngisor:
digitalWrite(buzzer, HIGH);
delay(2000);
digitalWrite(buzzer, LOW);
while(1)
{
    lompat1();
    EEPROM.write(erata2tinggi1, terbesar[1]);
    lompat2();
    EEPROM.write(erata2tinggi2, terbesar[2]);
    lompat3();
    EEPROM.write(erata2tinggi3, terbesar[3]);
    lompat4();
    EEPROM.write(erata2tinggi4, terbesar[4]);
    lompat5();
    EEPROM.write(erata2tinggi5, terbesar[5]);
    lompat6();
    EEPROM.write(erata2tinggi6, terbesar[6]);
    lompat7();
    EEPROM.write(erata2tinggi7, terbesar[7]);
}

```

```

        lompat8();
        EEPROM.write(erata2tinggi8, terbesar[8]);
        lompat9();
        EEPROM.write(erata2tinggi9, terbesar[9]);
        lompat10();
        EEPROM.write(erata2tinggi10, terbesar[10]);
        Serial.println("KIRIM DATA KE SERVER");
        delay(1000);
        while(1)
        {
            paket_data_power_endurance();
            goto akhir;
        }
    }
    akhir:
    digitalWrite(led_hijau, LOW);
    counter=0;
    while(1){break;}
    terima_data();
}
void paket_data_power_endurance()
{
    rata2tinggi1=EEPROM.read(erata2tinggi1);
    rata2tinggi2=EEPROM.read(erata2tinggi2);
    rata2tinggi3=EEPROM.read(erata2tinggi3);
    rata2tinggi4=EEPROM.read(erata2tinggi4);
    rata2tinggi5=EEPROM.read(erata2tinggi5);
    rata2tinggi6=EEPROM.read(erata2tinggi6);
    rata2tinggi7=EEPROM.read(erata2tinggi7);
    rata2tinggi8=EEPROM.read(erata2tinggi8);
    rata2tinggi9=EEPROM.read(erata2tinggi9);
    rata2tinggi10=EEPROM.read(erata2tinggi10);
    Serial.print(rata2tinggi1);
    Serial.print(',');
    Serial.print(rata2tinggi2);
    Serial.print(',');
    Serial.print(rata2tinggi3);
    Serial.print(',');
    Serial.print(rata2tinggi4);
    Serial.print(',');
    Serial.print(rata2tinggi5);
    Serial.print(',');
    Serial.print(rata2tinggi6);
    Serial.print(',');
    Serial.print(rata2tinggi7);
    Serial.print(',');
    Serial.print(rata2tinggi8);
    Serial.print(',');
    Serial.print(rata2tinggi9);
    Serial.print(',');
    Serial.println(rata2tinggi10);
    int a=rata2tinggi1;
    int b=rata2tinggi2;
    int c=rata2tinggi3;
    int d=rata2tinggi4;
    int e=rata2tinggi5;
    int f=rata2tinggi6;
    int g=rata2tinggi7;
    int h=rata2tinggi8;
    int i=rata2tinggi9;
    int j=rata2tinggi10;
    kirim_server.print(a);
    kirim_server.print(',');
    kirim_server.print(b);
    kirim_server.print(',');
    kirim_server.print(c);
    kirim_server.print(',');
    kirim_server.print(d);
    kirim_server.print(',');
    kirim_server.print(e);
    kirim_server.print(',');
    kirim_server.print(f);
}

```

```

    kirim_server.print(',');
    kirim_server.print(g);
    kirim_server.print(',');
    kirim_server.print(h);
    kirim_server.print(',');
    kirim_server.print(i);
    kirim_server.print(',');
    kirim_server.println(j);
    delay(1000);
}

void paket_data_vertikal_jump()
{
    rata2tinggi_Vjump=EEPROM.read(erata2tinggi_Vjump);
    waktu_depan_koma=EEPROM.read(ewaktu_depan_koma);
    waktu_belakang_koma=EEPROM.read(ewaktu_belakang_koma);
    Serial.print("V JUMP:");
    Serial.println(rata2tinggi_Vjump);
    Serial.print("depan koma:");
    Serial.println(waktu_depan_koma);
    Serial.print("belakang koma:");
    Serial.println(waktu_belakang_koma);
    int a=rata2tinggi_Vjump;
    int b=waktu_depan_koma;
    int c=waktu_belakang_koma;
    kirim_server.print(a);
    kirim_server.print(',');
    kirim_server.print(b);
    kirim_server.print(',');
    kirim_server.print(c);
    kirim_server.print(',');
    kirim_server.print(a);
    kirim_server.print(',');
    kirim_server.println(a);
    delay(1000);
}

void vertikal_jump()
{
    kalibrasi=EEPROM.read(ekalibrasi);
    Serial.print("TINGGI VERTIKAL JUMP");
    Serial.println("");
    for(int i=0;i<100;i++)
    {
        Serial.print(tinggi_Vjump[i]);
    }
    Serial.println(" ");
    terbesar_Vjump=tinggi_Vjump[0];
    for(int i=0;i<100;i++)
    {
        if(tinggi_Vjump[i]>terbesar_Vjump)
        {
            terbesar_Vjump=tinggi_Vjump[i]-kalibrasi;
        }
    }
    Serial.println(terbesar_Vjump);
}

void lompat1()
{
    kalibrasi=EEPROM.read(ekalibrasi);
    Serial.print("TINGGI LOMPATAN PERTAMA");
    Serial.println("");
    for(int i=0;i<10;i++)
    {
        Serial.print(tinggi1[i]);
    }
    Serial.println(" ");
    terbesar[1]=tinggi1[0];
    for(int i=0;i<10;i++)
    {
        if(tinggi1[i]>terbesar[1])
        {

```

```

        terbesar[1]=tinggi1[i]-kalibrasi;
    }
}
Serial.println(terbesar[1]);
}
void lompat2()
{
    kalibrasi=EEPROM.read(ekalibrasi);
    Serial.print("TINGGI LOMPATAN KEDUA");
    Serial.println("");
    for(int i=0;i<10;i++)
    {
        Serial.print(tinggi2[i]);
    }
    Serial.println(" ");
    terbesar[2]=tinggi2[0];
    for(int i=0;i<10;i++)
    {
        if(tinggi2[i]>terbesar[2])
        {
            terbesar[2]=tinggi2[i]-kalibrasi;
        }
    }
    Serial.println(terbesar[2]);
}
void lompat3()
{
    kalibrasi=EEPROM.read(ekalibrasi);
    Serial.print("TINGGI LOMPATAN KETIGA");
    Serial.println("");
    for(int i=0;i<10;i++)
    {
        Serial.print(tinggi3[i]);
    }
    Serial.println(" ");
    terbesar[3]=tinggi3[0];
    for(int i=0;i<10;i++)
    {
        if(tinggi3[i]>terbesar[3])
        {
            terbesar[3]=tinggi3[i]-kalibrasi;
        }
    }
    Serial.println(terbesar[3]);
}
void lompat4()
{
    kalibrasi=EEPROM.read(ekalibrasi);
    Serial.print("TINGGI LOMPATAN KEEMPAT");
    Serial.println("");
    for(int i=0;i<10;i++)
    {
        Serial.print(tinggi4[i]);
    }
    Serial.println(" ");
    terbesar[4]=tinggi4[0];
    for(int i=0;i<10;i++)
    {
        if(tinggi4[i]>terbesar[4])
        {
            terbesar[4]=tinggi4[i]-kalibrasi;
        }
    }
    Serial.println(terbesar[4]);
}
void lompat5()
{
    kalibrasi=EEPROM.read(ekalibrasi);
    Serial.print("TINGGI LOMPATAN KELIMA");
    Serial.println("");
    for(int i=0;i<10;i++)
    {

```

```

    Serial.print(tinggi5[i]);
  }
  Serial.println(" ");
  terbesar[5]=tinggi5[0];
  for(int i=0;i<10;i++)
  {
    if(tinggi5[i]>terbesar[5])
    {
      terbesar[5]=tinggi5[i]-kalibrasi;
    }
  }
  Serial.println(terbesar[5]);
}
void lompat6()
{
  kalibrasi=EEPROM.read(ekalibrasi);
  Serial.print("TINGGI LOMPATAN KEENAM");
  Serial.println("");
  for(int i=0;i<10;i++)
  {
    Serial.print(tinggi6[i]);
  }
  Serial.println(" ");
  terbesar[6]=tinggi6[0];
  for(int i=0;i<10;i++)
  {
    if(tinggi6[i]>terbesar[6])
    {
      terbesar[6]=tinggi6[i]-kalibrasi;
    }
  }
  Serial.println(terbesar[6]);
}
void lompat7()
{
  kalibrasi=EEPROM.read(ekalibrasi);
  Serial.print("TINGGI LOMPATAN KETUJUJUH");
  Serial.println("");
  for(int i=0;i<10;i++)
  {
    Serial.print(tinggi7[i]);
  }
  Serial.println(" ");
  terbesar[7]=tinggi7[0];
  for(int i=0;i<10;i++)
  {
    if(tinggi7[i]>terbesar[7])
    {
      terbesar[7]=tinggi7[i]-kalibrasi;
    }
  }
  Serial.println(terbesar[7]);
}
void lompat8()
{
  kalibrasi=EEPROM.read(ekalibrasi);
  Serial.print("TINGGI LOMPATAN KEDELAPAN");
  Serial.println("");
  for(int i=0;i<10;i++)
  {
    Serial.print(tinggi8[i]);
  }
  Serial.println(" ");
  terbesar[8]=tinggi8[0];
  for(int i=0;i<10;i++)
  {
    if(tinggi8[i]>terbesar[8])
    {
      terbesar[8]=tinggi8[i]-kalibrasi;
    }
  }
  Serial.println(terbesar[8]);
}

```

```
}  
void lompat9()  
{  
  kalibrasi=EEPROM.read(ekalibrasi);  
  Serial.print("TINGGI LOMPATAN SEMBILAN");  
  Serial.println("");  
  for(int i=0;i<10;i++)  
  {  
    Serial.print(tinggi9[i]);  
  }  
  Serial.println(" ");  
  terbesar[9]=tinggi9[0];  
  for(int i=0;i<10;i++)  
  {  
    if(tinggi9[i]>terbesar[9])  
    {  
      terbesar[9]=tinggi9[i]-kalibrasi;  
    }  
  }  
  Serial.println(terbesar[9]);  
}  
void lompat10()  
{  
  kalibrasi=EEPROM.read(ekalibrasi);  
  Serial.print("TINGGI LOMPATAN KEDELAPAN");  
  Serial.println("");  
  for(int i=0;i<10;i++)  
  {  
    Serial.print(tinggi10[i]);  
  }  
  Serial.println(" ");  
  terbesar[10]=tinggi10[0];  
  for(int i=0;i<10;i++)  
  {  
    if(tinggi10[i]>terbesar[10])  
    {  
      terbesar[10]=tinggi10[i]-kalibrasi;  
    }  
  }  
  Serial.println(terbesar[10]);  
}
```

Lampiran 4. Program *Slave 2 Controller* (Arduino NANO 2)

```

#include <StopWatch.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
#include <EEPROM.h>
LiquidCrystal_I2C lcd_ziky(0x27 ,2,1,0,4,5,6,7,3, POSITIVE);
SoftwareSerial kirim_server(10,9);// RX, TX
StopWatch sw_millis; // MILLIS (default)
StopWatch sw_micros(StopWatch::MICROS);
StopWatch sw_secs(StopWatch::SECONDS);
const int m_maju=8;
const int m_mundur=7;
const int m_enable=6;
int encoderSwitchPin = 1; //push button switch
int encoderPin1 = 2;
int encoderPin2 = 3;
int sensorWarna = A1;
int sensorProximity = A2;

volatile int lastEncoded = 0;
volatile long encoderValue = 0;

long lastencoderValue = 0;

int lastMSB = 0;
int lastLSB = 0;
int sensorValue1, hSwarna=0;
int sensorValue2, hSprox=0;

int jumlahPutaran=0;
float jarak=0;
float jarak2=0;
int jarak_fix=0;
int pembagi_jarak=0;
float hasil_bagi_jarak=0;
float hasil_belakang_koma_jarak=0;

float waktu=0;
float teta_waktu=0;
int pembagi_waktu=0;
float hasil_bagi_waktu=0;
float hasil_belakang_koma_timer=0;

//ALAMAT EEPROM
int addr_jarak_depan_koma=1;
int addr_jarak_depan_koma_plus=2;
int addr_jarak_belakang_koma=3;
int addr_waktu_depan_koma=4;
int addr_waktu_belakang_koma=5;
//VARIABEL BACA ISI EEPROM
int read_ejarak_depan_koma=0;
int read_ejarak_depan_koma_plus=0;
int read_ejarak_belakang_koma=0;
int read_ewaktu_depan_koma=0;
int read_ewaktu_belakang_koma=0;

void timer()
{
    waktu=sw_micros.elapsed();
    teta_waktu=waktu/1000000;
    pembagi_waktu=teta_waktu-0.5;
    hasil_bagi_waktu=teta_waktu-pembagi_waktu;
    hasil_belakang_koma_timer=hasil_bagi_waktu*100;
}

void Jarak()
{

```

```

    jarak=(encoderValue*0.80984)+5;//1.07984 adalah nilai dari persamaan jarak
    tempuh sebenarnya
    if(jarak>250)
    {
        jarak2=jarak-250;
        pembagi_jarak=jarak2;
        hasil_bagi_jarak=jarak2-pembagi_jarak;
        hasil_belakang_koma_jarak=hasil_bagi_jarak*100;
        jarak_fix=250+jarak2;
        EEPROM.write(addr_jarak_depan_koma, 250);
        EEPROM.write(addr_jarak_depan_koma_plus, jarak2);
        EEPROM.write(addr_jarak_belakang_koma, hasil_belakang_koma_jarak);
    }
    else
    {
        jarak2=0;
        pembagi_jarak=jarak;
        hasil_bagi_jarak=jarak-pembagi_jarak;
        hasil_belakang_koma_jarak=hasil_bagi_jarak*100;
        jarak_fix=pembagi_jarak+jarak2;
        EEPROM.write(addr_jarak_depan_koma, pembagi_jarak);
        EEPROM.write(addr_jarak_depan_koma_plus, jarak2);
        EEPROM.write(addr_jarak_belakang_koma, hasil_belakang_koma_jarak);
    }
}

void setup()
{
    Serial.begin (57600);
    kirim_server.begin(9600);
    lcd_zlky.begin(16,2);
    pinMode(m_maju, OUTPUT);
    pinMode(m_mundur, OUTPUT);
    pinMode(m_enable, OUTPUT);
    pinMode(encoderPin1, INPUT);
    pinMode(encoderPin2, INPUT);
    pinMode(encoderSwitchPin, INPUT);
    pinMode(sensorWarna, INPUT);
    pinMode(sensorProximity, INPUT);

    digitalWrite(encoderPin1, HIGH); //turn pullup resistor on
    digitalWrite(encoderPin2, HIGH); //turn pullup resistor on
    digitalWrite(encoderSwitchPin, HIGH); //turn pullup resistor on

    //on interrupt 0 (pin 2), or interrupt 1 (pin 3)
    attachInterrupt(0, updateEncoder, CHANGE);
    attachInterrupt(1, updateEncoder, CHANGE);
}

void loop()
{
    motor_stop();
    terima_data();
}

void sensor()
{
    sensorValue1 = analogRead(sensorWarna);
    hSwarna =map(sensorValue1,0,1023,0,255);
    sensorValue2 = analogRead(sensorProximity);
    hSprox =map(sensorValue2,0,1023,0,255);
    Serial.print(hSwarna);
    Serial.print("\t");
    Serial.println(hSprox);
}

void terima_data()
{
    atas:
    if(kirim_server.available())
    {

```

```

char data=(char) kirim_server.read();
if(data=='1')
{
  Serial.println("HORIZONTAL JUMP SIAP");
  lcd_ziky.clear();
  horizontal_jump();
  goto atas;
}
}
else
{
  lcd_ziky.setCursor(0,0);
  lcd_ziky.print("menunggu perintah");
  lcd_ziky.setCursor(0,1);
  lcd_ziky.print("dari SERVER ZIOS");
}
}

void paket_data_horizontal_jump()
{
  read_ejarak_depan_koma=EEPROM.read(addr_jarak_depan_koma);
  read_ejarak_depan_koma_plus=EEPROM.read(addr_jarak_depan_koma_plus);
  read_ejarak_belakang_koma=EEPROM.read(addr_jarak_belakang_koma);
  read_ewaktu_depan_koma=EEPROM.read(addr_waktu_depan_koma);
  read_ewaktu_belakang_koma=EEPROM.read(addr_waktu_belakang_koma);
  jarak_fix=read_ejarak_depan_koma+read_ejarak_depan_koma_plus;
  Serial.print("dkj:");
  Serial.println(read_ejarak_depan_koma);
  Serial.print("dkj_plus:");
  Serial.println(read_ejarak_depan_koma_plus);
  Serial.print("bkj:");
  Serial.println(read_ejarak_belakang_koma);
  Serial.print("dkt:");
  Serial.println(read_ewaktu_depan_koma);
  Serial.print("bkt:");
  Serial.println(read_ewaktu_belakang_koma);
  int a=read_ejarak_depan_koma;
  int b=read_ejarak_depan_koma_plus;
  int c=read_ejarak_belakang_koma;
  int d=read_ewaktu_depan_koma;
  int e=read_ewaktu_belakang_koma;
  kirim_server.print(a);
  kirim_server.print(',');
  kirim_server.print(b);
  kirim_server.print(',');
  kirim_server.print(c);
  kirim_server.print(',');
  kirim_server.print(d);
  kirim_server.print(',');
  kirim_server.println(e);
  delay(1000);
}

void horizontal_jump()
{
  while(1)
  {
    motor_stop();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("ZIOS H.JUMP=ready");
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print("J.Lompat:");
    lcd_ziky.print(jarak);
    lcd_ziky.print(" ");
    if(kirim_server.available())
    {
      char data=(char) kirim_server.read();
      if(data=='7')
      {
        Serial.println("GO!!!");
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("ZIOS H.JUMP=GO!!!");
      }
    }
  }
}

```

```

        delay(250);
        lcd_ziky.clear();
        goto go;
    }
}
go:
while(1)
{
    sw_millis.start();
    sw_micros.start();
    sw_secs.start();
    while(1)
    {
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("ZIOS H.JUMP=run");
        motor_maju();
        delay(500);
        goto next1;
    }
    next1:
    while(1)
    {
        sensor();
        motor_maju();
        jarak=(encoderValue*0.80984)+5;//1.07984 adalah nilai dari persamaan
jarak tempuh sebenarnya
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("ZIOS H.JUMP=run");
        lcd_ziky.setCursor(0,1);
        lcd_ziky.print("J.Lompat:");
        lcd_ziky.print(jarak);
        lcd_ziky.print("  ");
        if(hSprox<100)
        {
            sw_millis.stop();
            sw_micros.stop();
            sw_secs.stop();
            lcd_ziky.setCursor(0,0);
            lcd_ziky.print("ZIOS H.JUMP=stop");
            motor_stop();
            lcd_ziky.clear();
            goto next2;
        }
        if(hSwarna<60)
        {
            hSwarna=0;
            lcd_ziky.clear();
            goto mundur2_gan;
        }
    }
    next2:
    while(1)
    {
        timer();
        Jarak();
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("dkj:");
        lcd_ziky.print(jarak_fix);
        lcd_ziky.setCursor(0,1);
        lcd_ziky.print("bkj:");
        lcd_ziky.print(hasil_belakang_koma_jarak);
        lcd_ziky.setCursor(9,0);
        lcd_ziky.print("dkt:");
        lcd_ziky.print(pembagi_waktu);
        lcd_ziky.setCursor(9,1);
        lcd_ziky.print("bkt:");
        lcd_ziky.print(hasil_belakang_koma_timer);
        EEPROM.write(addr_waktu_depan_koma, pembagi_waktu);
        EEPROM.write(addr_waktu_belakang_koma, hasil_belakang_koma_timer);
        delay(1000);
        lcd_ziky.clear();
    }
}

```

```

        goto next3;
    }
    next3:
    while(1)
    {
        paket_data_horizontal_jump();
        lcd_ziky.setCursor(0,0);
        lcd_ziky.print("dkj:");
        lcd_ziky.print(jarak_fix);
        lcd_ziky.setCursor(0,1);
        lcd_ziky.print("bkj:");
        lcd_ziky.print(read_ejarak_belakang_koma);
        lcd_ziky.setCursor(9,0);
        lcd_ziky.print("dkt:");
        lcd_ziky.print(read_ewaktu_depan_koma);
        lcd_ziky.setCursor(9,1);
        lcd_ziky.print("bkt:");
        lcd_ziky.print(read_ewaktu_depan_koma);
        delay(2000);
        lcd_ziky.clear();
        goto mundur1_gan;
    }
}
mundur1_gan:
while(1)
{
    sensor();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("MUNDUR");
    motor_mundur();
    if(hSwarna<60)
    {
        goto selesai;
    }
}
mundur2_gan:
while(1)
{
    sensor();
    lcd_ziky.setCursor(0,1);
    lcd_ziky.print("MUNDUR");
    motor_mundur();
    delay(4000);
    if(hSwarna<60)
    {
        goto selesai;
    }
}
selesai:
while(1)
{
    sensor();
    if(hSwarna<55)
    {
        goto akhir;
    }
}
akhir:
while(1)
{
    motor_stop();
    lcd_ziky.setCursor(0,0);
    lcd_ziky.print("STOP");
    sw_millis.reset();
    sw_micros.reset();
    sw_secs.reset();
    encoderValue=0;
    jarak=0;
    jarak2=0;
    delay(300);
    lcd_ziky.clear();
    goto berhenti;
}

```

```
}
berhenti:
while(1){break;}
}

void jumlah_putaran_roda()
{
  lcd_ziky.setCursor(0,1);
  lcd_ziky.print("jmlh putaran:");
  if(encoderValue==22)
  {
    jumlahPutaran=1;
  }
  if(encoderValue==24)
  {
    jumlahPutaran=2;
  }
  if(encoderValue==66)
  {
    jumlahPutaran=3;
  }
  if(encoderValue==88)
  {
    jumlahPutaran=4;
  }
  if(encoderValue==110)
  {
    jumlahPutaran=5;
  }
  if(encoderValue==132)
  {
    jumlahPutaran=6;
  }
  if(encoderValue==154)
  {
    jumlahPutaran=7;
  }
  if(encoderValue==176)
  {
    jumlahPutaran=8;
  }
  if(encoderValue==198)
  {
    jumlahPutaran=9;
  }
  if(encoderValue==220)
  {
    jumlahPutaran=10;
  }
  if(encoderValue==242)
  {
    jumlahPutaran=11;
  }
  lcd_ziky.print(jumlahPutaran);
  lcd_ziky.print(" ");
}

void motor_maju()
{
  digitalWrite(m_maju, LOW);
  digitalWrite(m_mundur, LOW);
  digitalWrite(m_enable, LOW);
}

void motor_mundur()
{
  digitalWrite(m_maju, HIGH);
  digitalWrite(m_mundur, HIGH);
  digitalWrite(m_enable, LOW);
}

void motor_stop()
```

```

{
  digitalWrite(m_maju, HIGH);
  digitalWrite(m_mundur, LOW);
  digitalWrite(m_enable, LOW);
}

void bablas()
{
  if(hSwarna<60)
  {
    hSwarna=0;
    goto mundur_gan;
  }
  mundur_gan:
  while(1)
  {
    sensor();
    motor_mundur();
    delay(3000);
    if(hSwarna<60)
    {
      goto selesai;
    }
  }
  selesai:
  while(1)
  {
    sensor();
    if(hSwarna<55)
    {
      goto akhir;
    }
  }
  akhir:
  while(1)
  {
    motor_stop();
    break;
  }
}

void encoder()
{
  sensor();
  digitalWrite(m_enable, HIGH);
  if(digitalRead(encoderSwitchPin))
  {
    //program yang berjalan saat center switch tidak ditekan
  }
  else
  {
    //program yang dijalankan saat center switch ditekan
    encoderValue=0;
  }
  Serial.print(hSwarna);
  Serial.print("\t");
  Serial.print(hSprox);
  Serial.print("\t");
  Serial.println(encoderValue);
  delay(100);
}

void updateEncoder()
{
  int MSB = digitalRead(encoderPin1); //MSB = most significant bit
  int LSB = digitalRead(encoderPin2); //LSB = least significant bit

  int encoded = (MSB << 1) | LSB; //converting the 2 pin value to single number
  int sum = (lastEncoded << 2) | encoded; //adding it to the previous encoded
  value

  if(sum == 0b1101) encoderValue++;
  if(sum == 0b1110) encoderValue--;
}

```

```
lastEncoded = encoded; //store this value for next time  
}
```

Lampiran 5. Program *Slave 3* (Wemos D1)

```

#include <ESP8266WiFi.h>
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
SoftwareSerial wemos(13,12);

const char* ssid = "@wifi.id_ToKoZik.com";
const char* password = "56741234";

const char* host = "smartech515.xyz";           //ex : "ganyang-
angkara.blogspot.com"
const char* php = "/change.php";             //ex : "/change.php"
const char* get_id = "?get_id=1";           //**** = harus di ubah ke bentuk
angka, sesuai dengan id
const char* get_tb = "&get_tb=";
const char* get_bb = "&get_bb=";
const char* get_tlvj = "&get_tlvj=";
const char* get_wvj = "&get_wvj=";
const char* get_kvj = "&get_kvj=";
const char* get_pvj = "&get_pvj=";
const char* get_plhj = "&get_plhj=";
const char* get_whj = "&get_whj=";
const char* get_khj = "&get_khj=";
const char* get_phj = "&get_phj=";
const char* get_wbr = "&get_wbr=";
const char* get_pe1 = "&get_pe1=";
const char* get_pe2 = "&get_pe2=";
const char* get_pe3 = "&get_pe3=";
const char* get_pe4 = "&get_pe4=";
const char* get_pe5 = "&get_pe5=";
const char* get_pe6 = "&get_pe6=";
const char* get_pe7 = "&get_pe7=";
const char* get_pe8 = "&get_pe8=";
const char* get_pe9 = "&get_pe9=";
const char* get_pe10 = "&get_pe10=";
const char* get_rpe = "&get_rpe=";
const char* get_maxpe = "&get_maxpe=";
const char* get_minpe = "&get_minpe=";
const int httpPort = 80;

WiFiClient client;

String dataIn1;
String dt1[30];
boolean parsing=false;
int i;
char inChar;

String url;
int reconnect=0;

float d1,d2,d3,d4,d5,d6,d7,d8,d9,d10;
float d11,d12,d13,d14,d15,d16,d17,d18,d19,d20;
float d21,d22,d23,d24;

void setup()
{
  wemos.begin(9600);
  Serial.begin(115200);
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  connect_wifi();
  Serial.println("Server started");
  dataIn1="";
}

```

```

void loop()
{
  while(wemos.available())
  {
    inChar=(char)wemos.read();
    dataIn1 += inChar;
    if (inChar == '\n')
    {
      parsing = true;
    }
  }
  if(parsing)
  {
    parsing_data_dari_serverMega();
    while(1)
    {
      if (!client.connect(host, httpPort))
      {
        Serial.println("connection failed");
        reconnect++;
        if(reconnect>=5)
        {
          reconnect=0;
          connect_wifi();
        }
      }
      else
      {
        data();
        client.print("GET ");
        client.print(url);
        client.println(" HTTP/1.1");
        client.print("Host: ");
        client.println(host);
        client.println();
        delay(10);
        while(client.available())
        {
          String line = client.readStringUntil('\r');
          Serial.print(line);
          delay(100);
        }
        delay(100);
        Serial.println();
        Serial.println("Joss Manisee...");
        break;
      }
    }
    parsing=false;
  }
}

void data_random()
{
  /* url = php;
  url += get_id;
  url += get_tb + String(12.5);
  url += get_bb + String(random(30,60));
  url += get_tlvj + String(random(60,90));
  url += get_wvj + String(random(90,120));
  url += get_kvj + String(random(120,30));
  url += get_pvj + String(random(100,200));
  url += get_plhj + String(random(150,180));
  url += get_whj + String(random(190,130));
  url += get_khj + String(random(120,300));
  url += get_phj + String(random(110,230));
  url += get_wbr + String(random(100,310));
  url += get_pe1 + String(random(70,120));
  url += get_pe2 + String(random(30,50));
  url += get_pe3 + String(random(25,50));
  url += get_pe4 + String(random(10,40));
  url += get_pe5 + String(random(10,30));
  */
}

```

```

url += get_pe6 + String(random(35, 65));
url += get_pe7 + String(random(45, 10));
url += get_pe8 + String(random(10, 60));
url += get_pe9 + String(random(15, 50));
url += get_pe10 + String(random(10, 90));
url += get_rpe + String(random(12, 30));
url += get_maxpe + String(random(10, 30));
url += get_minpe + String(random(50, 80));    */
}
void data()
{
    url = php;
    url += get_id;
    url += get_tb + String(dt1[0].toInt());
    url += get_bb + String(dt1[1].toInt());
    url += get_tlvj + String(dt1[2].toInt());
    url += get_wvj + String(dt1[3].toInt());
    url += get_kvj + String(dt1[4].toInt());
    url += get_pvj + String(dt1[5].toInt());
    url += get_plhj + String(dt1[6].toInt());
    url += get_whj + String(dt1[7].toInt());
    url += get_khj + String(dt1[8].toInt());
    url += get_phj + String(dt1[9].toInt());
    url += get_wbr + String(dt1[10].toInt());
    url += get_pe1 + String(dt1[11].toInt());
    url += get_pe2 + String(dt1[12].toInt());
    url += get_pe3 + String(dt1[13].toInt());
    url += get_pe4 + String(dt1[14].toInt());
    url += get_pe5 + String(dt1[15].toInt());
    url += get_pe6 + String(dt1[16].toInt());
    url += get_pe7 + String(dt1[17].toInt());
    url += get_pe8 + String(dt1[18].toInt());
    url += get_pe9 + String(dt1[19].toInt());
    url += get_pe10 + String(dt1[20].toInt());
    url += get_rpe + String(dt1[21].toInt());
    url += get_maxpe + String(dt1[22].toInt());
    url += get_minpe + String(dt1[23].toInt());
}

void connect_wifi()
{
    Serial.println();
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi connected");

    Serial.print("IP Address :");
    Serial.println(WiFi.localIP());
}

void parsing_data_dari_serverMega()
{
    int j=0;
    Serial.print("data masuk : ");
    Serial.print(dataIn1);
    Serial.print("\n");

    dt1[j]="";
    for(i=0;i<=dataIn1.length();i++)
    {
        if ((dataIn1[i] == '#') || (dataIn1[i] == ','))
        {
            j++;

```

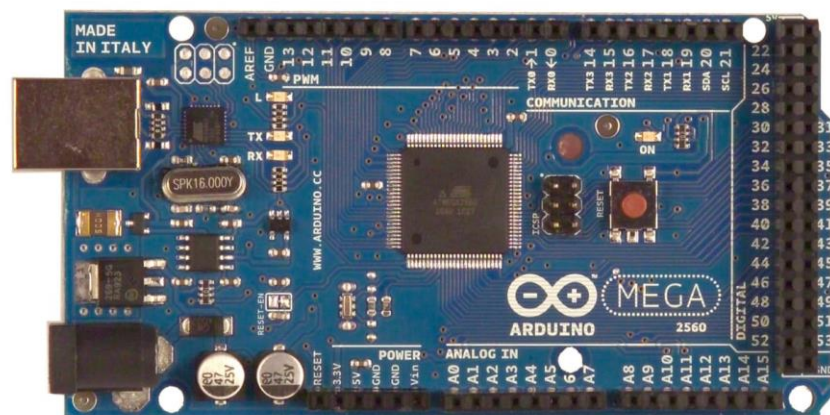
```
        dt1[j]="";          //inisialisasi variabel array dt[j]
    }
    else
    {
        dt1[j] = dt1[j] + dataIn1[i];
    }
}
Serial.print("data 1 : ");
Serial.print(dt1[0]);
Serial.print("\n");
Serial.print("data 2 : ");
Serial.print(dt1[1]);
Serial.print("\n");
Serial.print("data 3 : ");
Serial.print(dt1[2]);
Serial.print("\n");
Serial.print("data 4 : ");
Serial.print(dt1[3]);
Serial.print("\n");
Serial.print("data 5 : ");
Serial.print(dt1[4]);
Serial.print("\n");
Serial.print("data 6 : ");
Serial.print(dt1[5]);
Serial.print("\n");
Serial.print("data 7 : ");
Serial.print(dt1[6]);
Serial.print("\n");
Serial.print("data 8 : ");
Serial.print(dt1[7]);
Serial.print("\n");
Serial.print("data 9 : ");
Serial.print(dt1[8]);
Serial.print("\n");
Serial.print("data 10 : ");
Serial.print(dt1[9]);
Serial.print("\n");
Serial.print("data 11 : ");
Serial.print(dt1[10]);
Serial.print("\n");
Serial.print("data 12 : ");
Serial.print(dt1[11]);
Serial.print("\n");
Serial.print("data 13 : ");
Serial.print(dt1[12]);
Serial.print("\n");
Serial.print("data 14 : ");
Serial.print(dt1[13]);
Serial.print("\n");
Serial.print("data 15 : ");
Serial.print(dt1[14]);
Serial.print("\n");
Serial.print("data 16 : ");
Serial.print(dt1[15]);
Serial.print("\n");
Serial.print("data 17 : ");
Serial.print(dt1[16]);
Serial.print("\n");
Serial.print("data 18 : ");
Serial.print(dt1[17]);
Serial.print("\n");
Serial.print("data 19 : ");
Serial.print(dt1[18]);
Serial.print("\n");
Serial.print("data 20 : ");
Serial.print(dt1[19]);
Serial.print("\n");
Serial.print("data 21 : ");
Serial.print(dt1[20]);
Serial.print("\n");
Serial.print("data 22 : ");
Serial.print(dt1[21]);
```

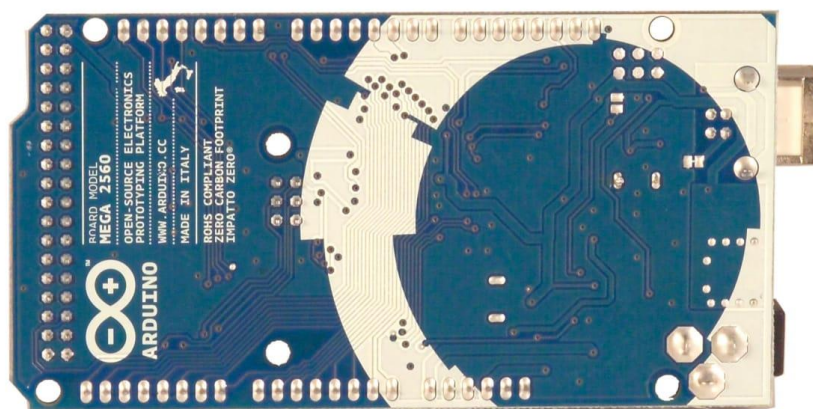
```
Serial.print("\n");  
Serial.print("data 23 : ");  
Serial.print(dtl[22]);  
Serial.print("\n");  
Serial.print("data 24 : ");  
Serial.print(dtl[23]);  
Serial.print("\n\n");  
delay(100);  
}
```

Lampiran 6. Datasheet Arduino Mega 2560



Arduino Mega 2560 Datasheet





Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Schematic & Reference Design

EAGLE files: [arduino-mega2560-reference-design.zip](#)



Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.



La robotique à votre service! - Robotics at your service!

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH



value, the LED is on, when the pin is LOW, it's off.

- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and `analogReference()` function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I²C (TWI) and SPI communication. The Arduino software includes a `Wire` library to simplify use of the I²C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It



communicates using the original STK500 protocol ([reference](#), [C header files](#)). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

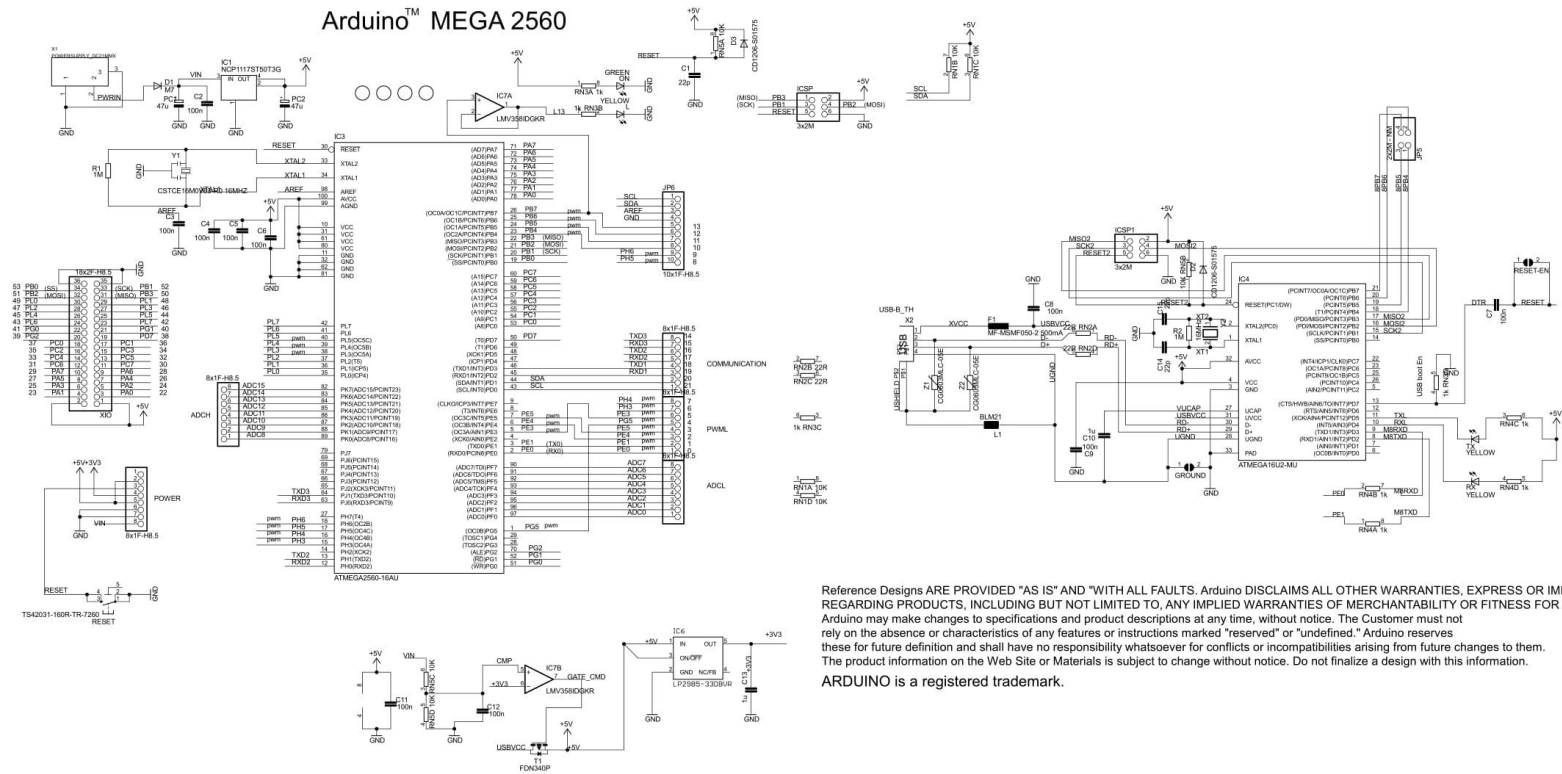
The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility



The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Diecimila. *Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).*

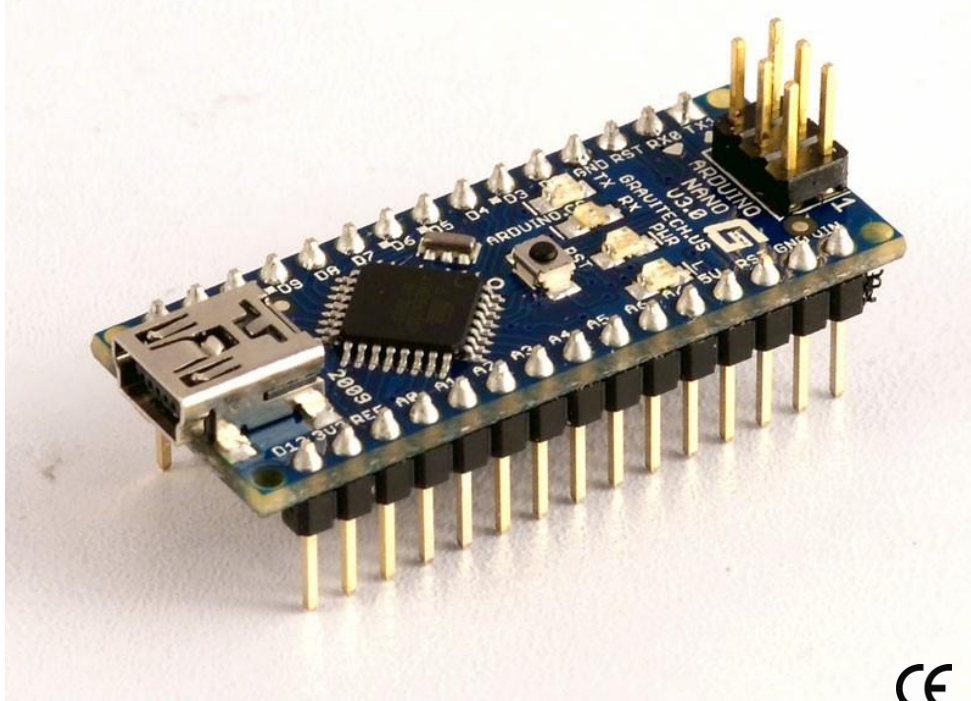


Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information. ARDUINO is a registered trademark.

Lampiran 7. Datasheet Arduino Nano 328p



Arduino Nano



Product Overview

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.0) or ATmega168 (Arduino Nano 2.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one. The Nano was designed and is being produced by Gravitech.

Index

Technical Specifications

Page 2

How to use Arduino
Programming Environment, Basic Tutorials

Page 6

Terms & Conditions

Page 7



Technical Specification



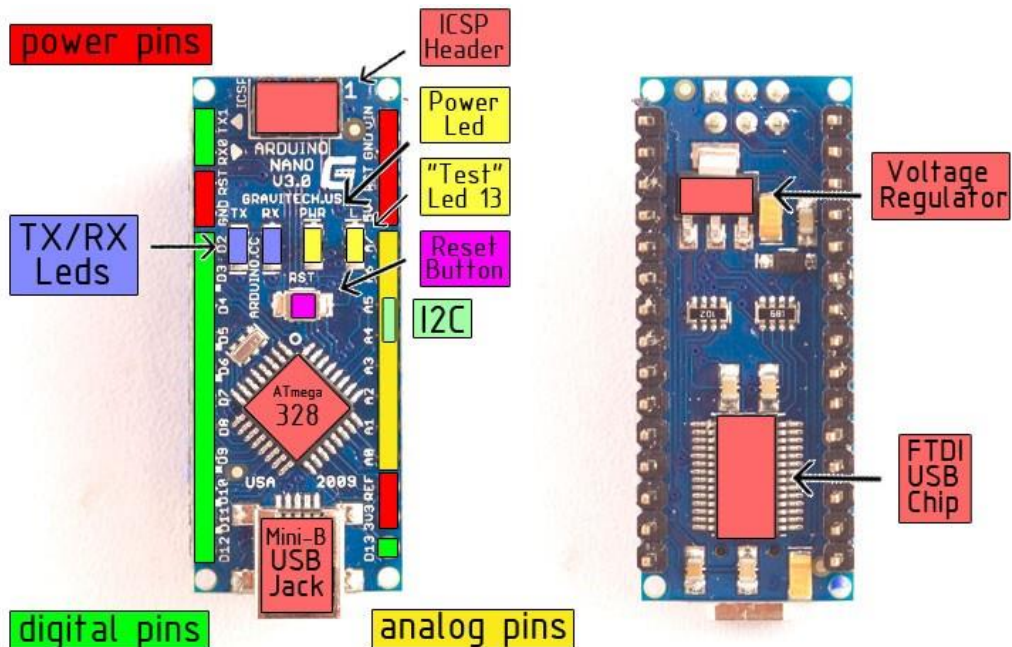
Arduino Nano 3.0 (ATmega328): [schematic](#), [Eagle files](#).

Arduino Nano 2.3 (ATmega168): [manual](#) (pdf), [Eagle files](#). Note: since the free version of Eagle does not handle more than 2 layers, and this version of the Nano is 4 layers, it is published here unrouted, so users can open and use it in the free version of Eagle.

Summary

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"

the board



Power

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

The FTDI FT232RL chip on the Nano is only powered if the board is being powered over USB. As a result, when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328 has 32 KB, (also with 2 KB used for the bootloader). The ATmega168 has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the [EEPROM library](#)); the ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

Each of the 14 digital pins on the Nano can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega168 ports](#).

Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 and ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the [FTDI drivers](#) (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

The ATmega168 and ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega168 or ATmega328 datasheet.

Programming

The Arduino Nano can be programmed with the Arduino software ([download](#)). Select "Arduino Diecimila, Duemilanove, or Nano w/ ATmega168" or "Arduino Duemilanove or Nano w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega168 or ATmega328 on the Arduino Nano comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Nano is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega168 or ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Nano is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Nano. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you'll need to install the FTDI Drivers to let your PC talk to the board. First **Plug the Arduino to your PC via USB cable**.

Blink led

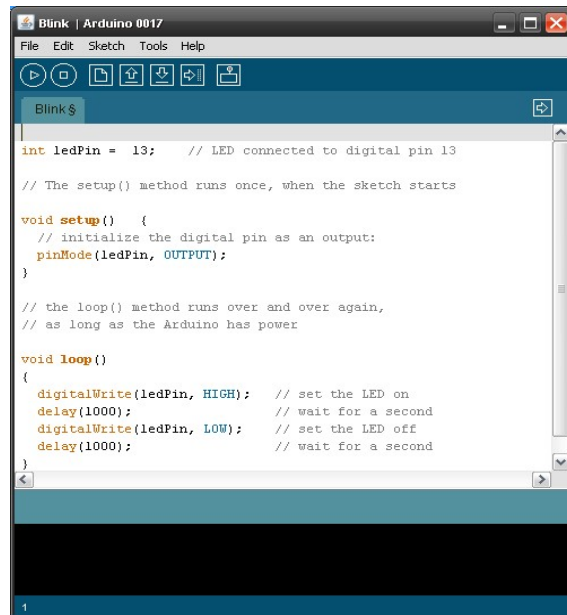
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select Arduino NANO and with the AtMEGA you're using (probably 328)

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.



```

Blink | Arduino 0017
File Edit Sketch Tools Help
Blink $
int ledPin = 13; // LED connected to digital pin 13
// The setup() method runs once, when the sketch starts
void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}
// the loop() method runs over and over again,
// as long as the Arduino has power
void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}

```



Done compiling.

Press Compile button
(to check for errors)



Upload

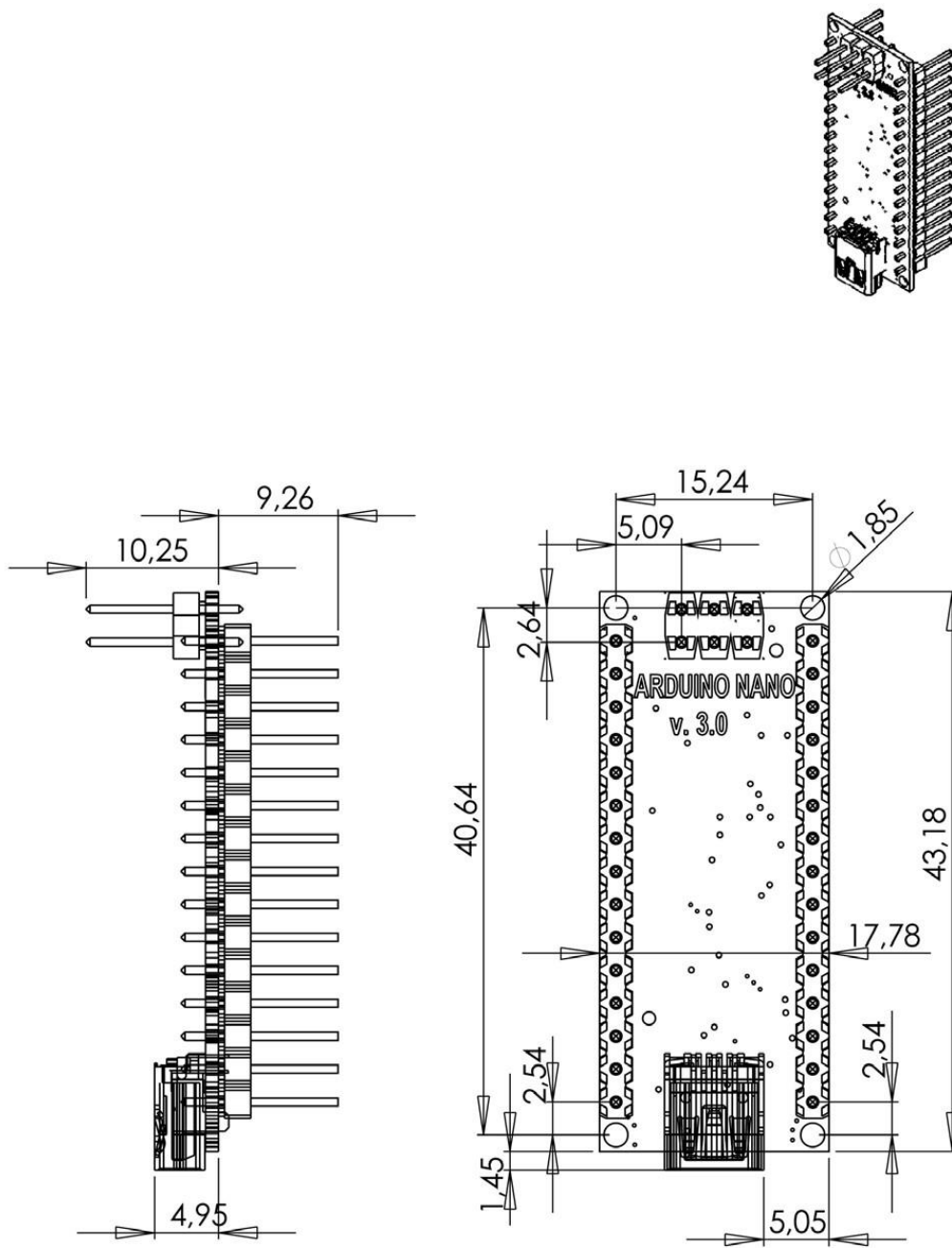


TX RX Flashing



Blinking Led!

Dimensioned Drawing



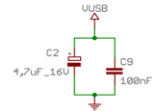
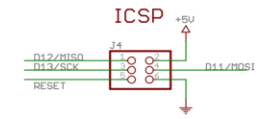
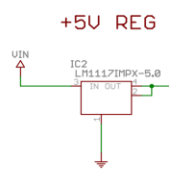
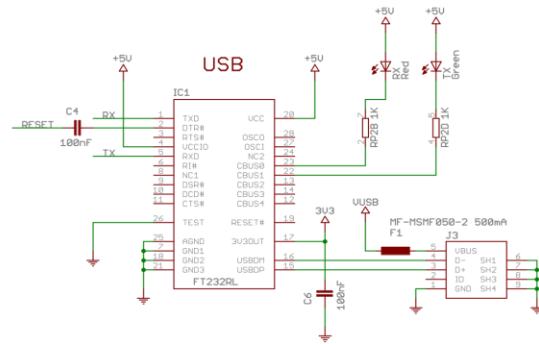
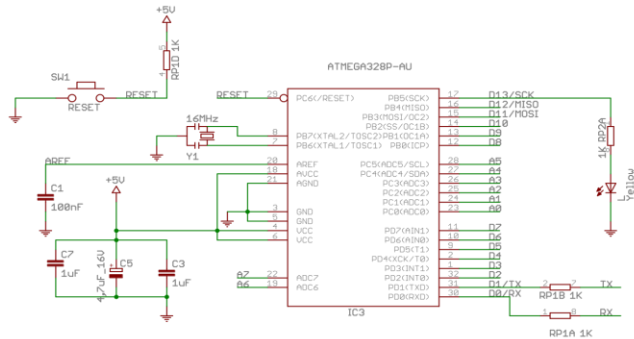
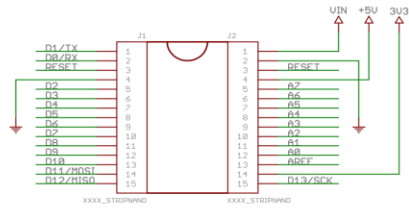
Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

"Arduino" name and logo are trademarks registered by Arduino S.r.l. in Italy, in the European Union and in other countries of the world.

Based on design by Gravitech (gravitech.us)

Arduino Nano

Released under Creative Commons Attribution Share Alike 3.0 Licence
<http://creativecommons.org/licenses/by-sa/3.0/>



Author: E.Uita	
TITLE: Arduino Nano-Rev3.2	
Document Number:	REU: 3.2
Date: 30.12.2014 15:30:26	Sheet: 1/1

Lampiran 8. Datasheet Ultrasonik HCSRFB04

SRFB04 Technical Documentation

10/22/12 5:35 AM

SRFB04 - Ultra-Sonic Ranger Technical Specification



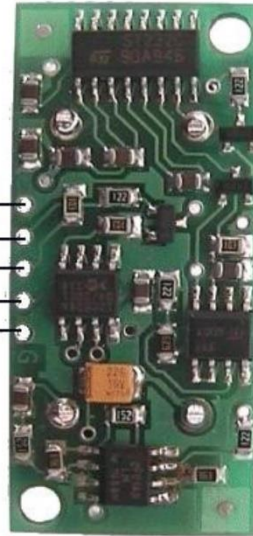
This project started after I looked at the Polaroid Ultrasonic Ranging module. It has a number of disadvantages for use in small robots etc.

1. The maximum range of 10.7 metre is far more than is normally required, and as a result
2. The current consumption, at 2.5 Amps during the sonic burst is truly horrendous.
3. The 150mA quiescent current is also far too high.
4. The minimum range of 26cm is useless. 1-2cm is more like it.
5. The module is quite large to fit into small systems, and
6. It's EXPENSIVE.

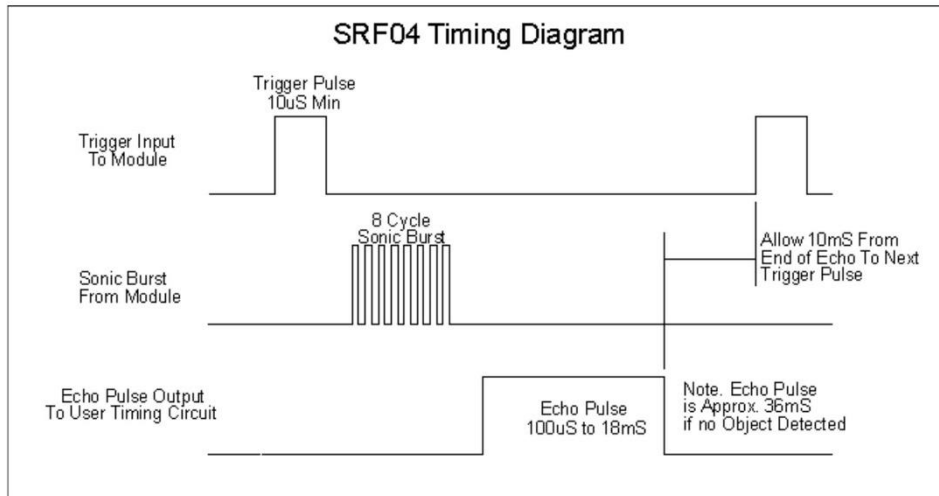
The SRF04 was designed to be just as easy to use as the Polaroid sonar, requiring a short trigger pulse and providing an echo pulse. Your controller only has to time the length of this pulse to find the range. The connections to the SRF04 are shown below:

SRF04 Connections

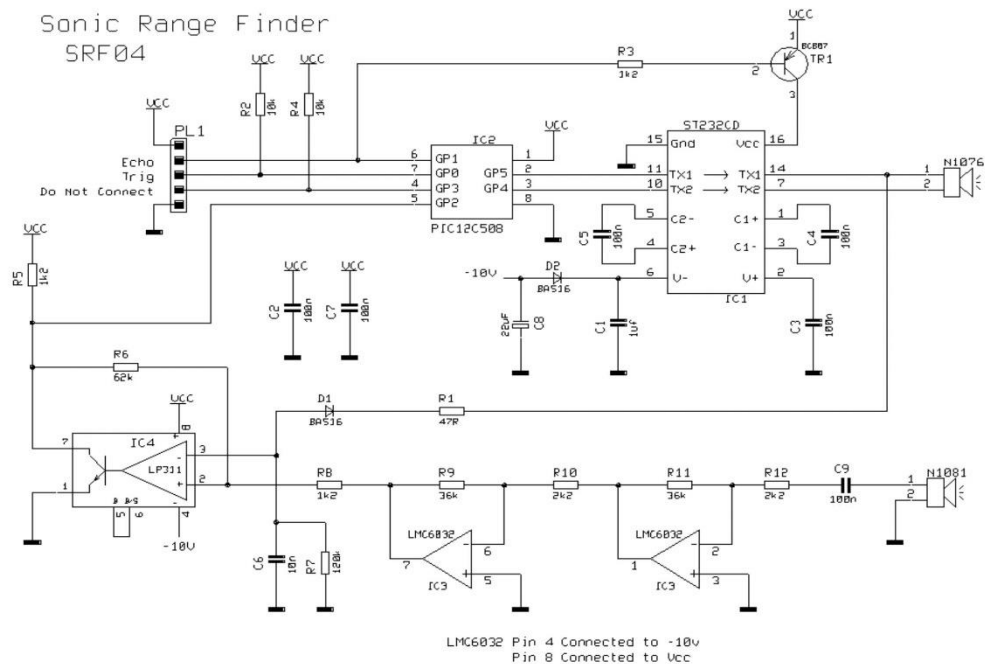
5v Supply —
Echo Pulse Output —
Trigger Pulse Input —
Do Not Connect —
0v Ground —



The SRF04 Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging. The SRF04 will send out an 8 cycle burst of ultrasound at 40khz and raise its echo line high. It then listens for an echo, and as soon as it detects one it lowers the echo line again. The echo line is therefore a pulse whose width is proportional to the distance to the object. By timing the pulse it is possible to calculate the range in inches/centimeters or anything else. If nothing is detected then the SRF04 will lower its echo line anyway after about 36mS.



Here is the schematic, You can download a better quality pdf (161k) version [srf1.pdf](#)



The circuit is designed to be low cost. It uses a PIC12C508 to perform the control functions and standard 40kHz piezo transducers. The drive to the transmitting transducer could be simplest driven directly from the PIC. The 5v drive can give a useful range for large objects, but can be problematic detecting smaller objects. The transducer can handle 20v of drive, so I decided to get up close to this level. A MAX232 IC, usually used for RS232 communication makes an ideal driver, providing about 16v of drive.

The receiver is a classic two stage op-amp circuit. The input capacitor C8 blocks some residual DC which always seems to be present. Each gain stage is set to 24 for a total gain of 576-ish. This is close to the 25 maximum gain available using the LM1458. The gain bandwidth product for the LM1458 is 1Mhz. The maximum gain at 40kHz is $1000000/40000 = 25$. The output of the amplifier is fed into an LM311 comparator. A small amount of positive feedback provides some hysteresis to give a clean stable output.

The problem of getting operation down to 1-2cm is that the receiver will pick up direct coupling from the transmitter, which is right next to it. To make matters worse the piezo transducer is a mechanical object that keeps resonating some time after the drive has been removed. Up to 1mS depending on when you decide it has stopped. It is much harder to tell the difference between this direct coupled ringing and a returning echo, which is why many designs, including the Polaroid module, simply blank out this period. Looking at the returning echo on an oscilloscope shows that it is much larger in magnitude at close quarters than the cross-coupled signal. I therefore adjust the detection threshold during this time so that only the echo is detectable. The 100nF capacitor C10 is charged to about -6v during the burst. This discharges quite quickly through the

10k resistor R6 to restore sensitivity for more distant echo's.

A convenient negative voltage for the op-amp and comparator is generated by the MAX232. Unfortunately, this also generates quite a bit of high frequency noise. I therefore shut it down whilst listening for the echo. The 10uF capacitor C9 holds the negative rail just long enough to do this.

In operation, the processor waits for an active low trigger pulse to come in. It then generates just eight cycles of 40khz. The echo line is then raised to signal the host processor to start timing. The raising of the echo line also shuts off the MAX232. After a while – no more than 10-12mS normally, the returning echo will be detected and the PIC will lower the echo line. The width of this pulse represents the flight time of the sonic burst. If no echo is detected then it will automatically time out after about 30mS (Its two times the WDT period of the PIC). Because the MAX232 is shut down during echo detection, you must wait at least 10mS between measurement cycles for the +/- 10v to recharge.

Performance of this design is, I think, quite good. It will reliably measure down to 3cm and will continue detecting down to 1cm or less but after 2-3cm the pulse width doesn't get any smaller.

Maximum range is a little over 3m. As an example of the sensitivity of this design, it will detect a 1inch thick plastic broom handle at 2.4m.

Average current consumption is reasonable at less than 50mA and typically about 30mA.

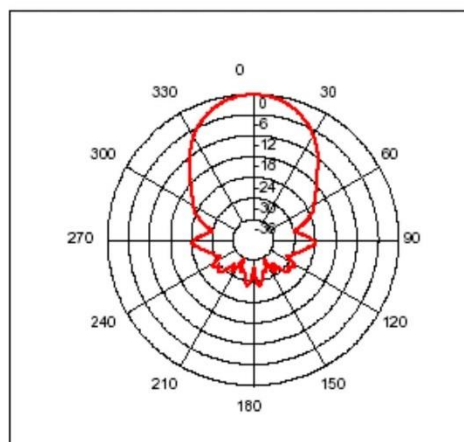
Download the [source code](#) and a ready assembled [hex file](#).

Calculating the Distance

The SRF04 provides an echo pulse proportional to distance. If the width of the pulse is measured in uS, then dividing by 58 will give you the distance in cm, or dividing by 148 will give the distance in inches.
 $uS/58=cm$ or $uS/148=inches$.

Changing beam pattern and beam width

You can't! This is a question which crops up regularly, however there is no easy way to reduce or change the beam width that I'm aware of. The beam pattern of the SRF04 is conical with the width of the beam being a function of the surface area of the transducers and is fixed. The beam pattern of the transducers used on the SRF04, taken from the manufacturers data sheet, is shown below.



There is more information in the [sonar faq](#).

Update - May 2003

Since the original design of the SRF04 was published, there have been incremental improvements to improve performance and manufacturing reliability. The op-amp is now an LMC6032 and the comparator is an LP311. The 10uF capacitor is now 22uF and a few resistor values have been tweaked. These changes have happened over a period of time.

All SRF04's manufactured after May 2003 have new software implementing an optional timing control input using the "do not connect" pin. This connection is the PIC's V_{pp} line used to program the chip after assembly. After programming its just an unused input with a pull-up resistor. When left unconnected the SRF04 behaves exactly as it always has and is described above. When the "do not connect" pin is connected to ground (0v), the timing is changed slightly to allow the SRF04 to work with the slower controllers such as the Picaxe. The SRF04's "do not connect" pin now acts as a timing control. **This pin is pulled high by default and when left unconnected, the timing remains exactly as before.** With the timing pin pulled low (grounded) a 300uS delay is added between the end of the trigger pulse and transmitting the sonic burst. Since the echo output is not raised until the burst is completed, there is no change to the range timing, but the 300uS delay gives the Picaxe time to sort out which pin to look at and start doing so. The new code has shipped in all SRF04's since the end of April 2003. The new code is also useful when connecting the SRF04 to the slower Stamps such as the BS2. Although the SRF04 works with the BS2, the echo line needs to be connected to the lower numbered input pins. This is because the Stamps take progressively longer to look at the higher numbered pins and can miss the rising edge of the echo signal. In this case you can connect the "do not connect" pin to ground and give it an extra 300uS to get there.

Lampiran 9. Datasheet modul Bluetooth HC05

DATASHEET BLUETOOTH TO SERIAL PORT MODULE HC05



Overview

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

www.electronica60norte.com
electronica60norte@hotmail.com

Specifications

Hardware features

- Typical -80dBm sensitivity.
- Up to +4dBm RF transmit power.
- Low Power 1.8V Operation, 3.3 to 5 V I/O.
- PIO control.
- UART interface with programmable baud rate.
- With integrated antenna.
- With edge connector.

Software features

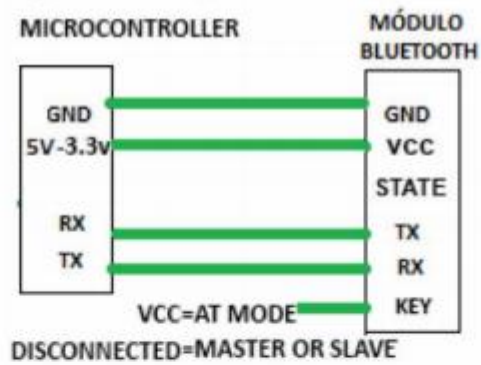
- Slave default Baud rate: 9600, Data bits:8, Stop bit:1,Parity:No parity.
- PIO9 and PIO8 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing **PINCODE:"1234"** as default.
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

www.electronica60norte.com
electronica60norte@hotmail.com

Pin out configuration



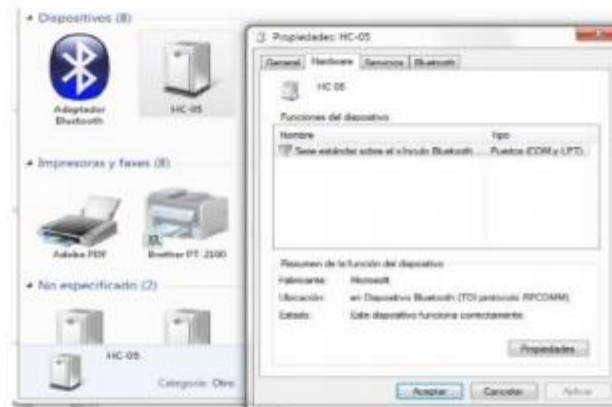
Typical Application Circuit



www.electronica60norte.com
electronica60norte@hotmail.com

After connect the Bluetooth module, scan for new devices from the PC and you will find the module with the device name "HC-05", after that, click to connect, if some message appears asking about "Pairing code" just put "1234" as default code.

BLUE LED = ACTIVE (Blinking 500ms period inactive connection, change 1seg with active connection)



Open a serial terminal and select the serial COM x port number that assigned Windows to Bluetooth Module.

Configure the serial terminal with these parameters:

- Baud rate: 9600.
- Data bits:8.
- Stop bit:1.
- Parity: No parity.

www.electronica60norte.com
electronica60norte@hotmail.com

Open connection and you will be ready to send and receive data from module Bluetooth like Serial Port COM



AT COMMANDS

How to get to AT COMMAND mode

- 1: Connect KEY pin to VCC.
- 2: Supply power to module. Then the module will enter into AT MODE. In this mode you have to use baud rate at 38400. In this way, user should change the baud rate for SLAVE AND MASTER mode.

How to set this module as "Master - Host" role

- 1: Input high level to KEY.
- 2: Supply power to the module. And the module will enter to AT COMMAND.
- 3: Set the parameters of the hyper terminal or the other serial tools (baud rate: 38400, data bit:8, stop bit:1, no parity bit, no Flow Control).
- 4: Sent the characters "AT+ROLE=1\r\n" through serial, then receive the characters "OK\r\n". Here, "\r\n" is the CRLF.
- 5: Sent the characters "AT+CMODE=1\r\n" through serial, then receive the characters "OK\r\n". Here, "\r\n" is the CRLF.
- 6: Default factory password passkey is: 1243, this must be the same in the Bluetooth slave module if you want to pair it.
To read passkey use this command: "AT+PSWD?".
To Reset the password command sent the characters "AT+PSWD=XXXX".
The password must be 4-bits.

7: Leave free KEY, and supply power to the module again. Then this module will become master role and search the other module (slave role) automatically to build the connection (baud rate:9600, data bit:8, stop bit:1, no parity bit, no Flow Control).

How to set this module be the "Slave - Device" role

- 1: Input high level to KEY.
- 2: Supply power to the module. And the module will enter to AT COMMAND.
- 3: Set the parameters of the super terminal or the other serial tools (baud rate: 38400, data bit:8, stop bit:1, no parity bit, no Flow Control).
- 4: Sent the characters "AT+ROLE=0\r\n" through serial, then receive the characters "OK\r\n". Here, "\r\n" is the CRLF.
- 5: Sent the characters "AT+CMODE=0\r\n" through serial, then receive the characters "OK\r\n". Here, "\r\n" is the CRLF.
- 6: Default factory password passkey is: 1243, this must be the same in the Bluetooth master module if you want to pair it.
To read passkey sent the characters "AT+PSWD?".
To Reset the password command sent the characters "AT+PSWD=XXXX".
The password must be 4-bits.
- 7: Leave free KEY, and supply power to the module again. Then this module will become slave role and wait to be discover it by the other module (master role) automatically to build the connection (baud rate:38400, data bit:8, stop bit:1, no parity bit, no Flow Control).

How to get to the standard communication mode

- 1: Leave free KEY, don't connect it to VDD neither GND.
- 2: Supply power to the module. Then the module will enter to communication mode. It can be used for pairing.

Notes

- (1) HC-05's command should end up with "\r\n". It means when you finish programming, you should add terminator ("ENTER" or "0x0d 0x0a") to the program.
 - (2) The most common commands for HC-05 are: AT+ROLE (set master-slave), AT+CMODE(set address pairing) , AT+PSWD (set password).
- If you want the master module has the function of remembering slave module, the most simply way is: First, set AT+CMODE=1. Make the master module pair with the slave module. Second, set AT+CMODE=0. Then the master module just can make pair with that specified slave module.

www.electronica60norte.com

electronica60norte@hotmail.com

Lampiran 10. Datasheet ESP8266

nodemcu-devkit

This is history.

New board is NodeMCU-DEVKIT-V1.0, see also <https://github.com/nodemcu/nodemcu-devkit-v1.0>

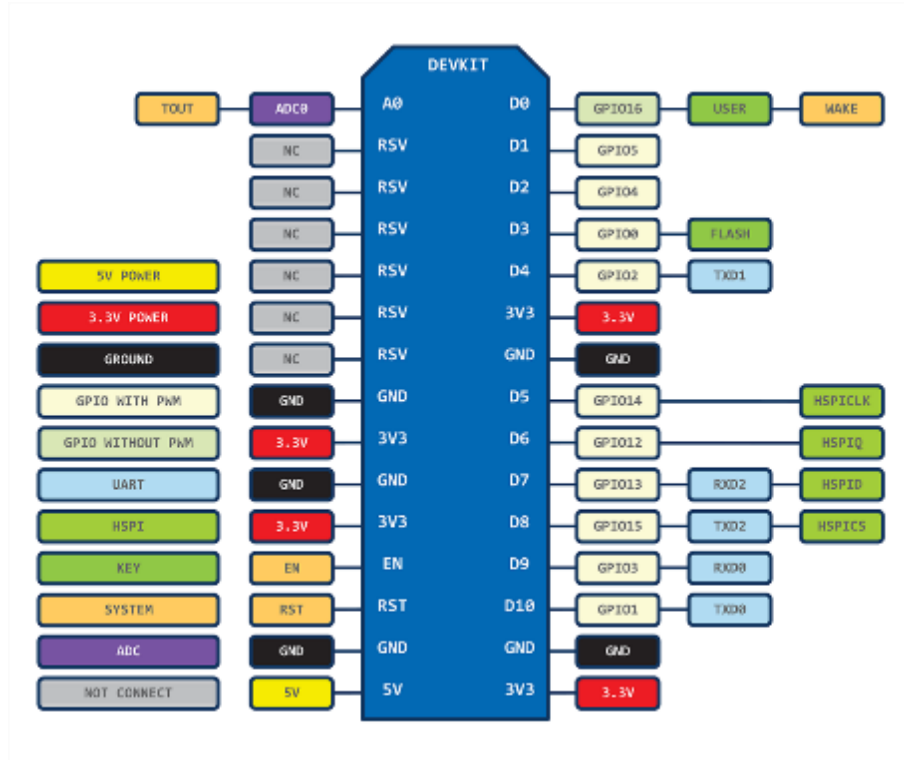
A development kit for NodeMCU firmware.

It will make NodeMCU more easy. With a micro USB cable, you can connect NodeMCU devkit to your laptop and flash it without any trouble, just like Arduino.



It is an open hardware, with ESP-12 core (4MBytes flash version).

Pin map with ESP-12.



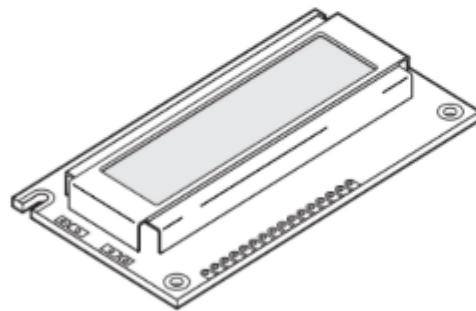
It is designed by Altium Designer, and fully open-source. Now everyone can make their own NODEMCU.

Lampiran 11. Datasheet LCD 16x2

ALPHANUMERIC LCD DISPLAY (16 x 2)

Order Code

LED008 16 x 2 Alphanumeric Display
FRM010 Serial LCD Firmware (optional)



Contents

1 x 16x2 Alphanumeric Display
1 x data booklet

Introduction

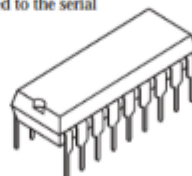
Alphanumeric displays are used in a wide range of applications, including palmtop computers, word processors, photocopiers, point of sale terminals, medical instruments, cellular phones, etc. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. A full list of the characters and symbols is printed on pages 7/8 (note these symbols can vary between brand of LCD used). This booklet provides all the technical specifications for connecting the unit, which requires a single power supply (+5V).

Further Information

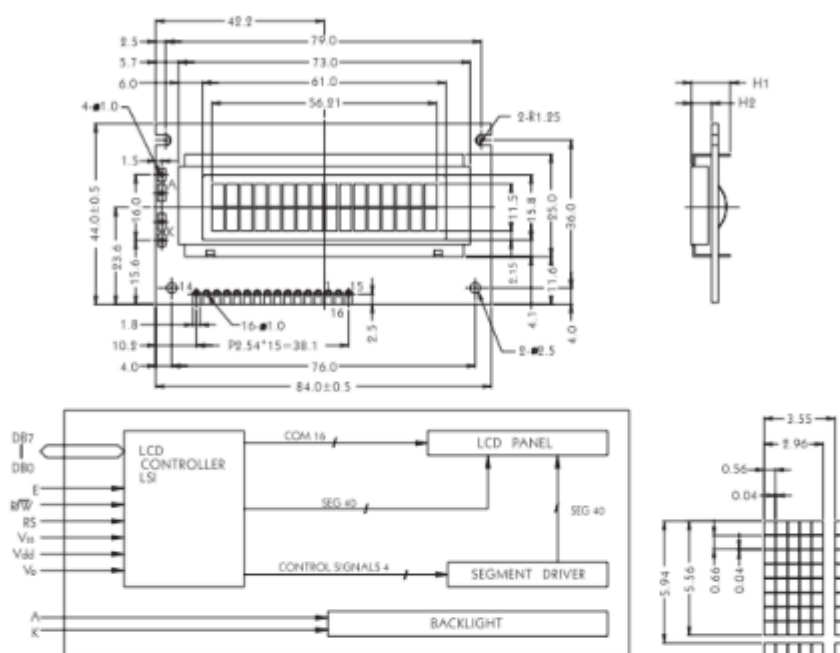
Available as an optional extra is the Serial LCD Firmware, which allows serial control of the display. This option provides much easier connection and use of the LCD module. The firmware enables microcontrollers (and microcontroller based systems such as the PICAXE) to visually output user instructions or readings onto an LCD module. All LCD commands are transmitted serially via a single microcontroller pin. The firmware can also be connected to the serial port of a computer.

An example PICAXE instruction to print the text 'Hello' using the `serout` command is as follows:

```
serout 7,T2400,("Hello")
```



Outline Dimension and Block Diagram



MECHANICAL SPECIFICATION

Overall Size	84.0 * 44.0	Module	H2 / H1
View Area	61.0 * 15.8	W/O B/L	5.1 / 9.7
Dot Size	0.56 * 0.66	EL B/L	5.1 / 9.7
Dot Pitch	0.60 * 0.70	LED B/L	9.4 / 14.0

PIN ASSIGNMENT

Pin no	Symbol	Function
1	Vss	Power supply (GND)
2	Vdd	Power supply (+5V)
3	Vc	Contrast Adjust
4	RS	Register select signal
5	R/W	Data read / write
6	E	Enable signal
7	DB0	Data bus line
8	DB1	Data bus line
9	DB2	Data bus line
10	DB3	Data bus line
11	DB4	Data bus line
12	DB5	Data bus line
13	DB6	Data bus line
14	DB7	Data bus line
15	A	Power supply for LED B/L (+)
16	K	Power supply for LED B/L (-)

ABSOLUTE MAXIMUM RATING

Item	Symbol	Conditions	Min.	Max.	Unit
Power Supply Voltage	Vdd=Vss	—	0	7	V
LCD Driving Supply Voltage	Vdd-Vee	—	0	13	V
Input Voltage	Vin	—	-0.3	Vdd+0.3	V
Operating Temperature	Toper	Nor.	0	50	°C
Storage Temperature	Tstg	Nor.	-20	+70	°C

ELECTRICAL CHARACTERISTICS (Vdd = +5V, Ta = 25°C)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Logic Supply Voltage	Vdd	—	4.5	5	5.5	V
"H" Input Voltage	Vih	—	2.2	—	—	V
"L" Input Voltage	Vil	—	—	—	0.6	V
"H" Output Voltage	Voh	—	2.4	—	—	V
"L" Output Voltage	Vol	—	—	—	0.4	V
Supply Current	Idd	—	2	—	—	mA
LCD Driving Voltage	Vco	Vdd-Vc	4.3	—	4.8	V

Lampiran 12. Datasheet Rotary Encoder Autonics

Incremental Shaft



Features:

- Miniature ø30mm shaft type rotary encoder
- Easy installation at narrow space
- Small inertia moment
- Wide range of power supply : 5VDC, 12-24VDC ±5%
- Various output types

Description:

Diameter ø30mm shaft type incremental rotary encoder E30S Series is easy to be installed at narrow space by minimizing space limit. Small moment of inertia is another feature of the series to optimize user convenience.

Specifications:

Item	Diameter ø30mm shaft type of incrementa rotary encoder		
Resolution(P/R)	100, 200, 360, 500, 1000, 1024, 3000 (Not indicated resolution is customizable.)		
Electrical specification	Output phase	A, B, Z phase(Line driver : A, A, B, B, Z, Z phase)	
	Phase difference of output	Phase difference between A and B : $T/4 \pm T/8$ (T = 1 cycle of A phase)	
	Control output	Totem pole output	Low Load current : Max. 30mA, Residual voltage : Max. 0.4V DC High Load current : Max. 10mA, Output voltage(Power supply 5V DC) : Min. (Power supply - 2)V DC, Output voltage(Power supply 12-24V DC) : Min. (Power supply - 3)V DC
		NPN open collector output	Load current : Max. 30mA, Residual voltage : Max. 0.4V DC
		Voltage output	Load current : Max. 10mA, Residual voltage : Max. 0.4V DC
		Line driver output	Low Load current : Max. 20mA, Residual voltage : Max. 0.5V DC High Load current : Max. -20mA, Output voltage : Min. 2.5V DC
	Response time (Rise/Fall)	Totem pole output	Max. 1µs
		NPN open collector output	Max. 1µs
		Voltage output	Max. 1µs (5V DC : Output resistance 820Ω), Max. 2µs (12 to 24V DC : Output resistance 4.7kΩ)
		Line driver output	Max. 0.5µs
	Max. Response frequency	300kHz	
	Power supply	5V DC ± 5%(Ripple P-P : Max. 5%) 12 to 24V DC ± 5%(Ripple P-P : Max. 5%)	
	Current consumption	Max. 80mA (disconnection of the load), Line driver output : Max. 50mA(disconnection of the load)	
	Insulation resistance	Min. 100MΩ(at 500V DC megger between all terminals and case)	
Dielectric strength	750V AC 50/60Hz for 1 minute(Between all terminals and case)		
Connection	Cable outgoing type, 250mm cable outgoing connector type		

www.element14.com
www.farnell.com
www.newark.com

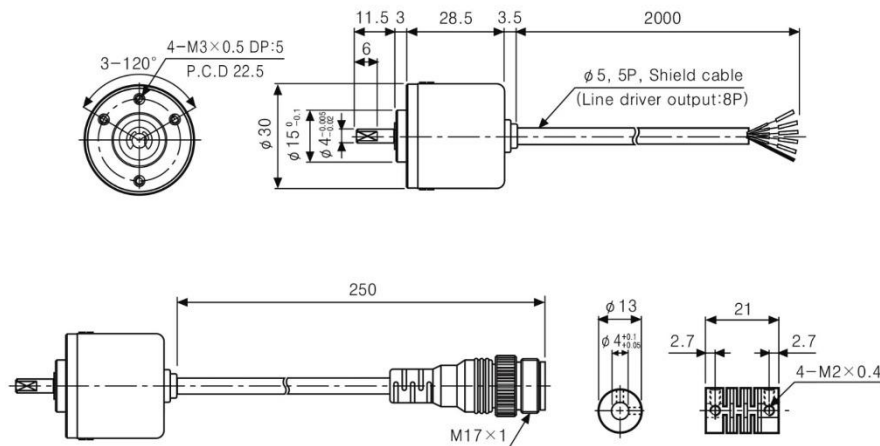


Incremental Shaft



Mechanical Specification	Starting torque	Max. 20gf · cm(0.002N · m)
	Moment of inertia	Max. 20g · cm ² (2 × 10 ⁻⁶ kg · m ²)
	Shaft loading	Radial : Max. 2kgf, Thrust : Max. 1kgf
	Max. allowable revolution	(Note 1) 5000rpm
Vibration	1.5mm amplitude at frequency of 10 to 55Hz in each of X, Y, Z directions for 2 hours	
Shock	Max. 50G	
Ambient temperature	-10°C to +70°C (at non-freezing status), Storage : -25°C to +85°C	
Ambient humidity	35 to 85%RH, Storage : 35 to 90%RH	
Protection	IP50(IEC standard)	
Cable	ø5mm, 5P, Length : 2m, Shield cable(Line driver : ø5mm, 8P)	
Accessory	ø4mm coupling	
Unit weight	Approx. 80g	

Note 1 : Pulse is only for A, B phase (Line Driver output is for A, A, B, B phase). It can be produced under 1,000 P/R for connector integrated type.



Dimensions : Millimetres

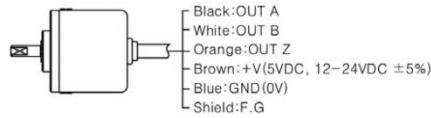
www.element14.com
www.farnell.com
www.newark.com



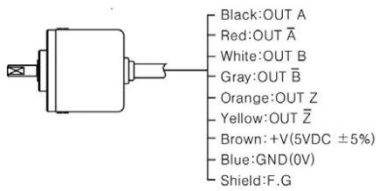
Incremental Shaft



Totem pole output / NPN open collector output / Voltage output



Line driver output



- * Unused wires must be insulated.
- * The metal case and shield wire of encoder should be grounded(F.G).

Part Number Table

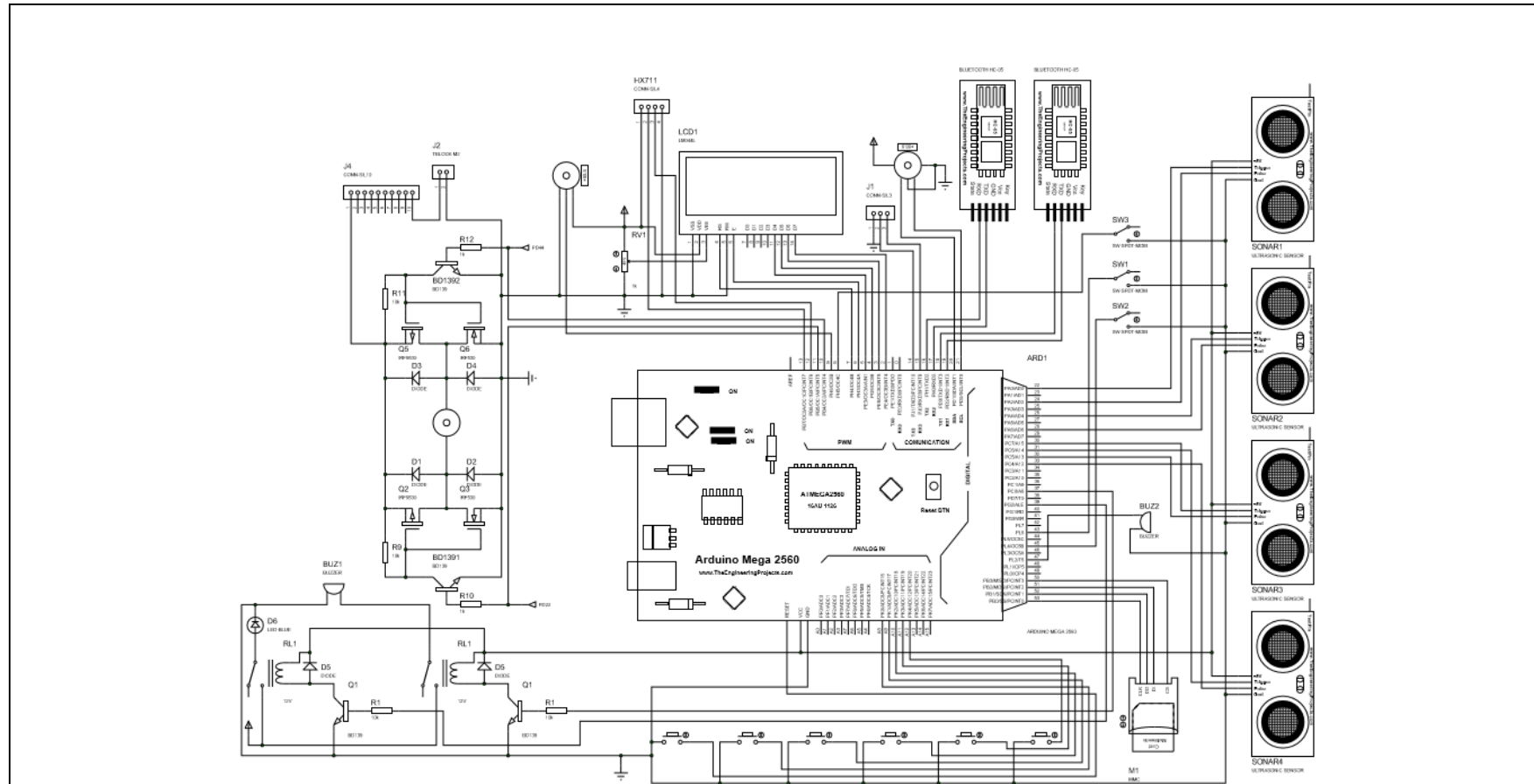
Description	Part Number
Incremental Shaft	E30S4-100-3 V-5

Important Notice: This data sheet and its contents (the "Information") belong to the members of the Premier Farnell group of companies (the "Group") or are licensed to it. No licence is granted for the use of it other than for information purposes in connection with the products to which it relates. No licence of any intellectual property rights is granted. The Information is subject to change without notice and replaces all data sheets previously supplied. The Information supplied is believed to be accurate but the Group assumes no responsibility for its accuracy or completeness, any error in or omission from it or for any use made of it. Users of this data sheet should check for themselves the Information and the suitability of the products for their purpose and not make any assumptions based on information included or omitted. Liability for loss or damage resulting from any reliance on the Information or use of it (including liability resulting from negligence or where the Group was aware of the possibility of such loss or damage arising) is excluded. This will not operate to limit or restrict the Group's liability for death or personal injury resulting from its negligence. Multicomp is the registered trademark of the Group. © Premier Farnell plc 2012.

www.element14.com
 www.farnell.com
 www.newark.com

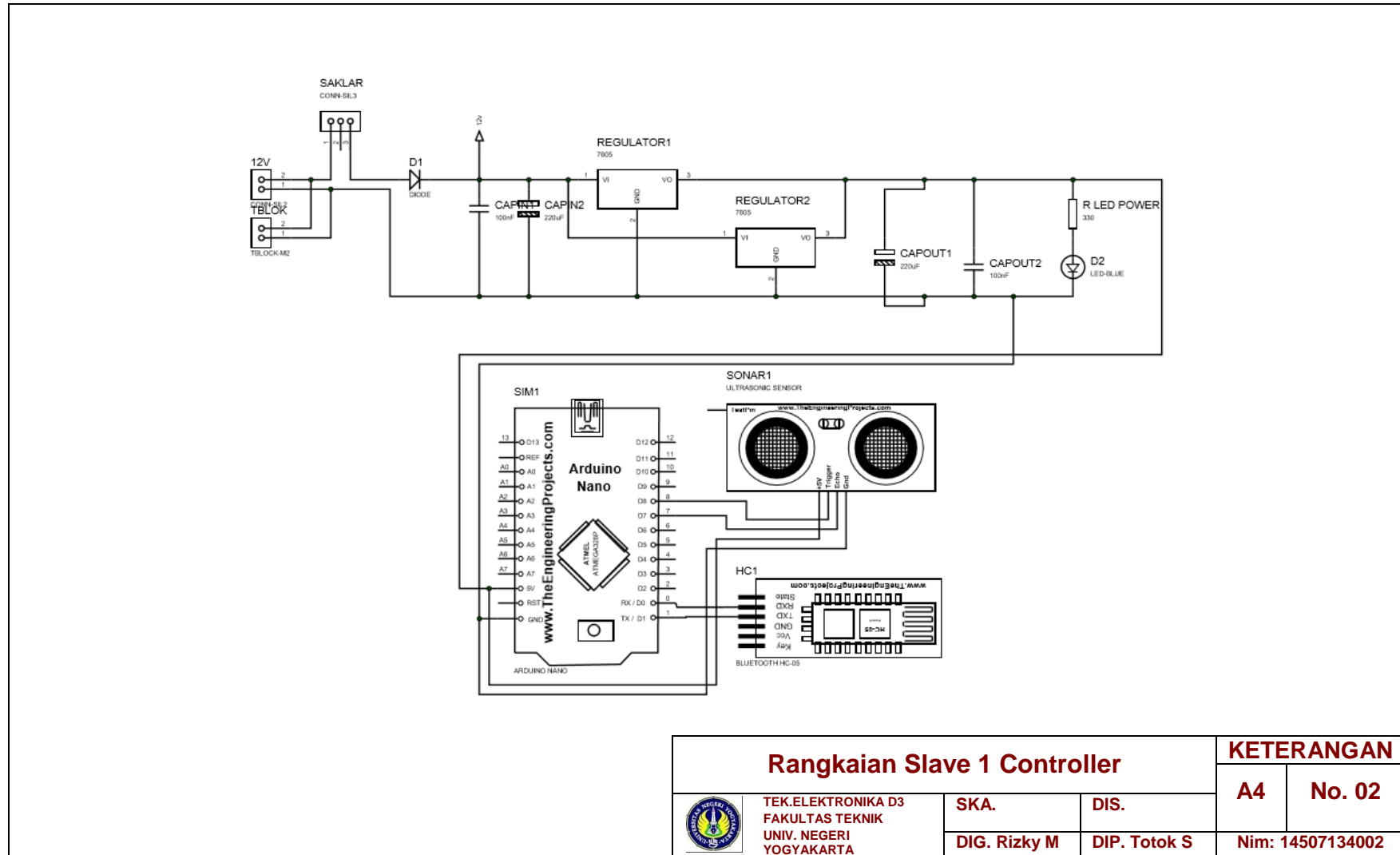


Lampiran 13. Skema Rangkaian Master Controller



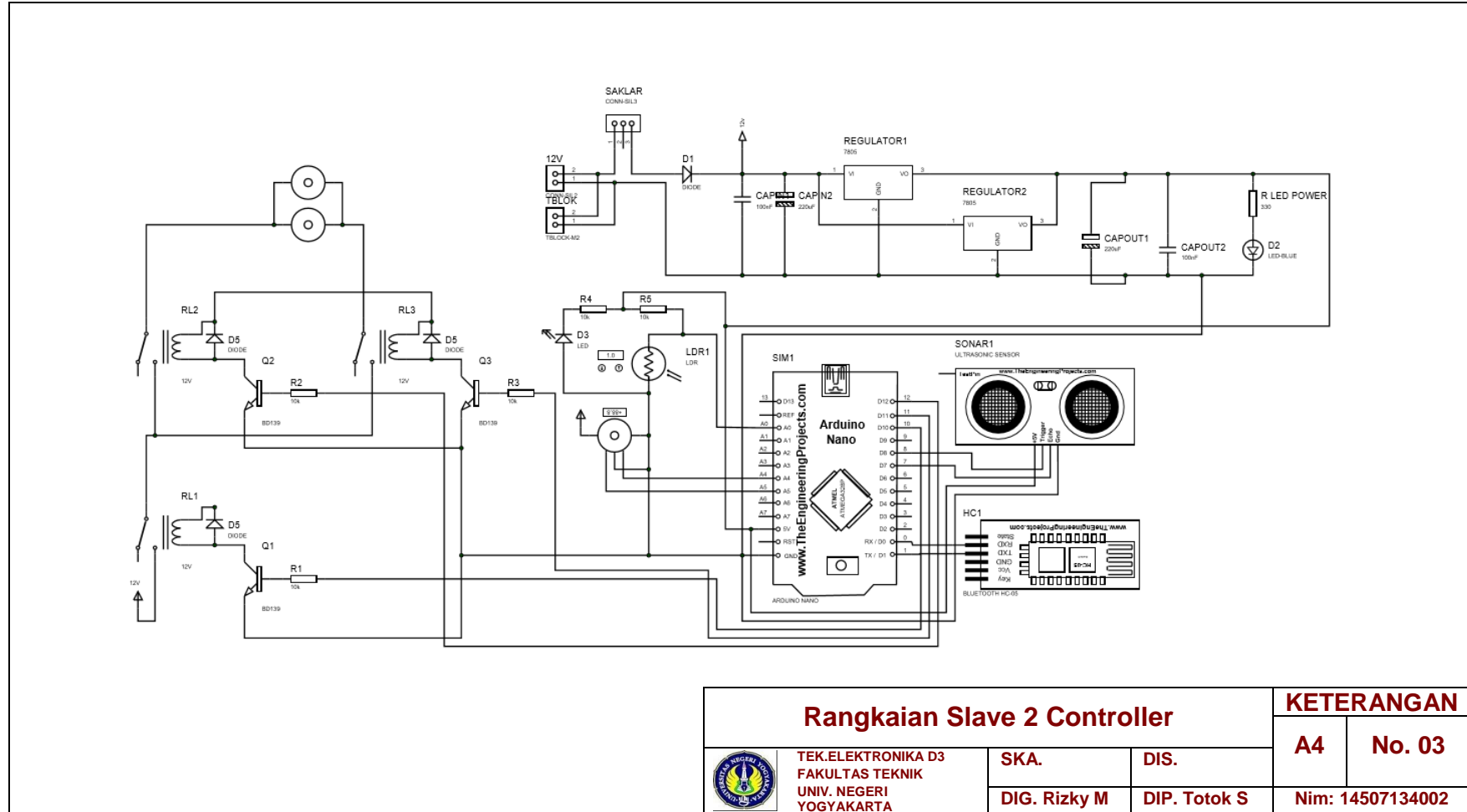
Rangkaian Master Controller			KETERANGAN	
	TEK.ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. DIG. Rizky M	DIS. DIP. Totok S	A4 No. 01
				Nim: 14507134002

Lampiran 14. Skema Rangkaian Slave 1 Controller



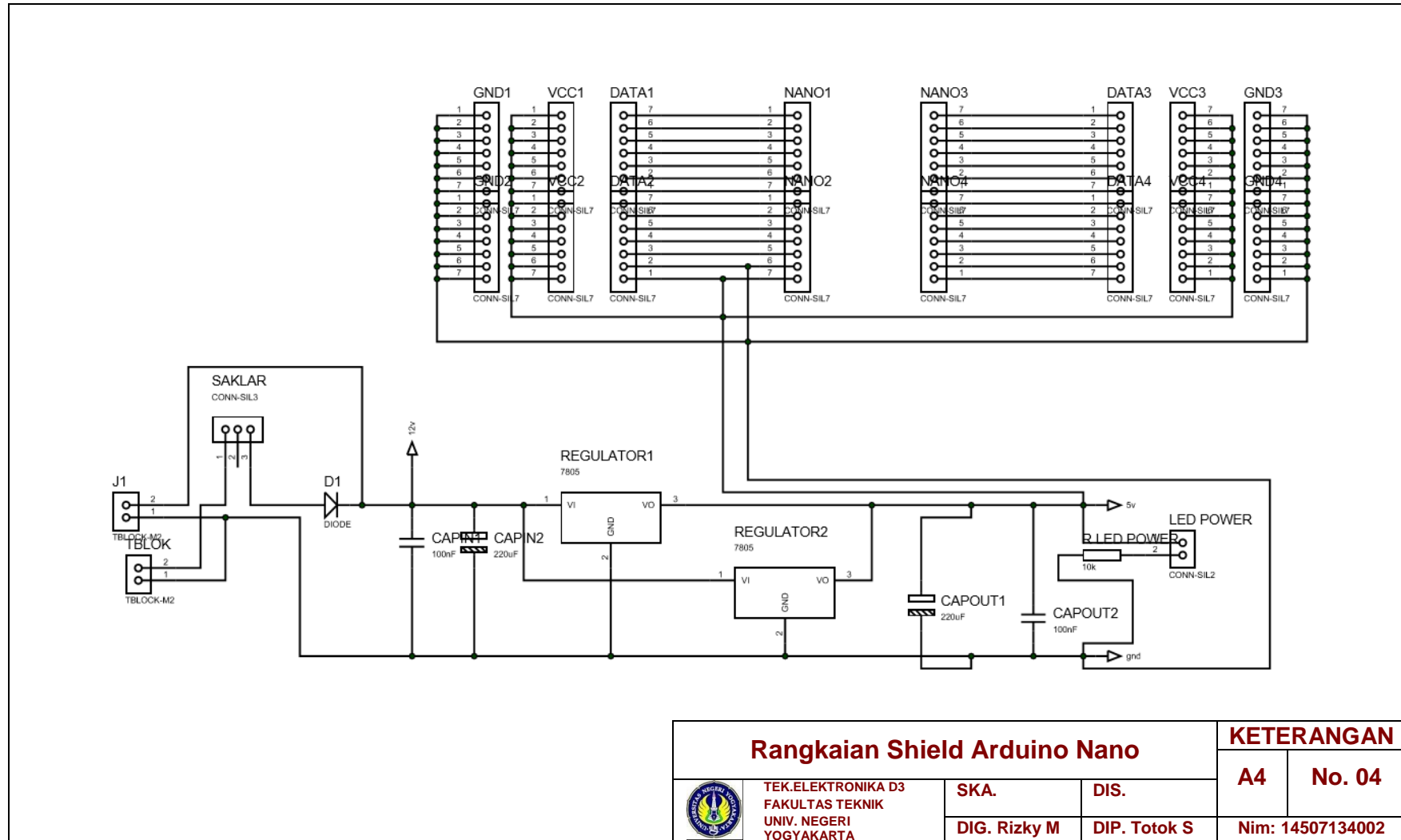
Rangkaian Slave 1 Controller			KETERANGAN	
			A4	No. 02
 TEK.ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA.	DIS.	Nim: 14507134002	
	DIG. Rizky M	DIP. Totok S		

Lampiran 15. Skema Rangkaian Slave 2 Controller

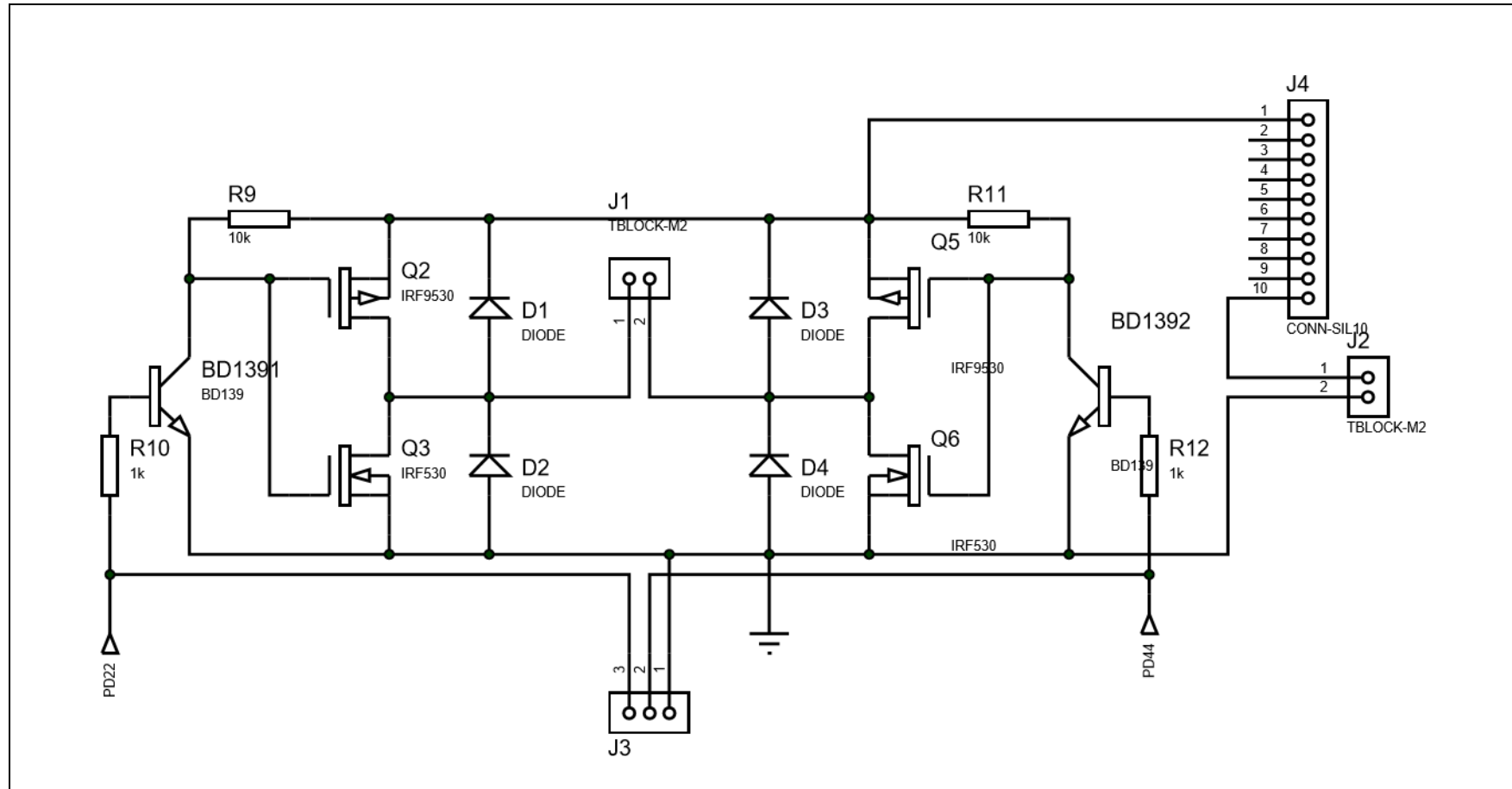



Rangkaian Slave 2 Controller			KETERANGAN	
 TEK.ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA.	DIS.	A4	No. 03
	DIG. Rizky M	DIP. Totok S	Nim: 14507134002	

Lampiran 16. Skema Rangkaian *Shield* Arduino Nano

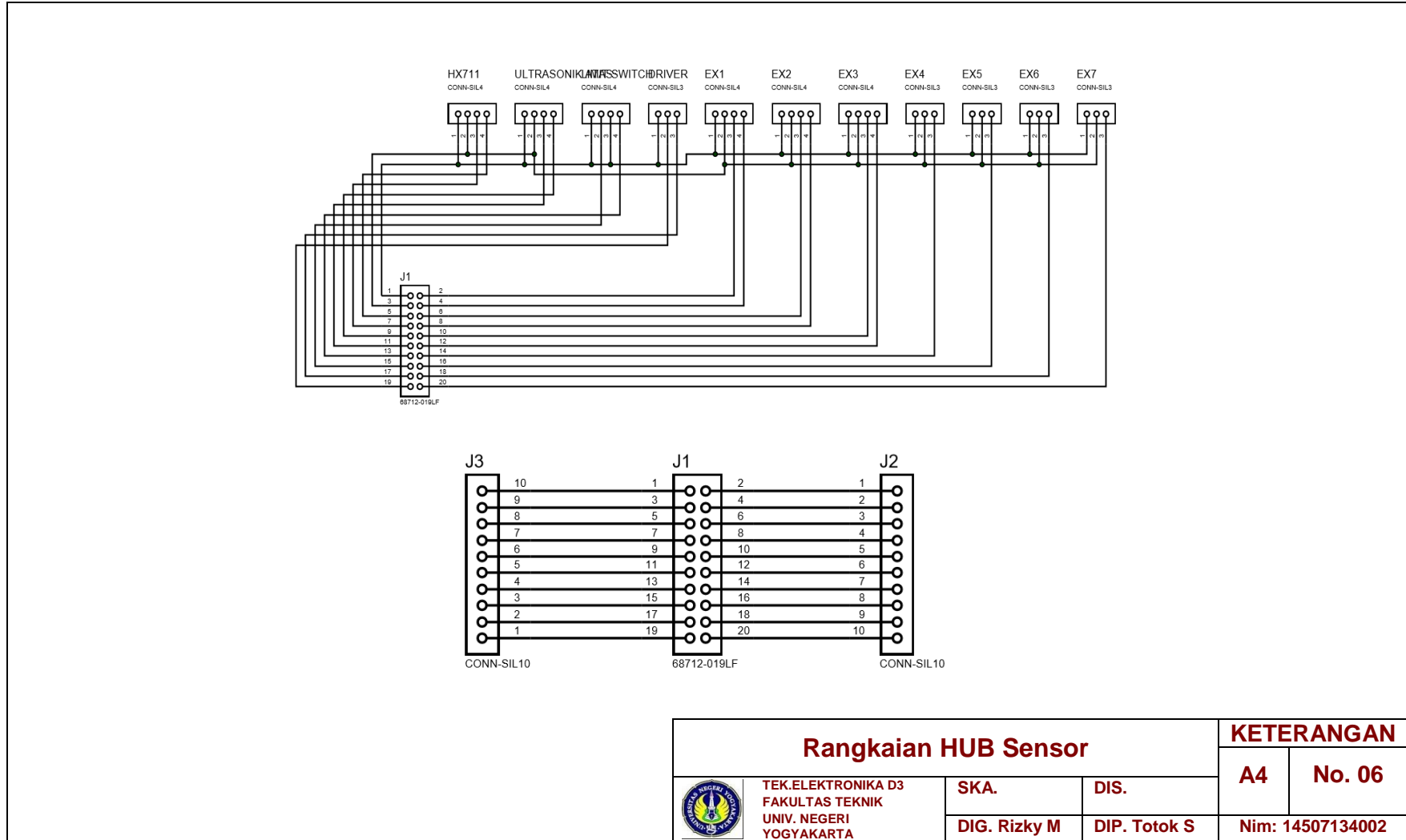


Lampiran 17. Skema Rangkaian Driver Motor H-Bridge Mosfet



 TEK.ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA			Rangkaian Driver Motor H-Bridge Mosfet		KETERANGAN	
			SKA.	DIS.	A4	No. 05
			DIG. Rizky M	DIP. Totok S	Nim: 14507134002	

Lampiran 18. Skema Rangkaian HUB Sensor



Lampiran 19. Daftar List Komponen dan Biaya

No	Nama Komponen	Spesifikasi	Jumlah	Harga Satuan	Harga Total
1.	Sensor Berat	<i>Loadcell</i>	4 buah	Rp 100.000,00	Rp 400.000,00
2.	Sensor Jarak	Ultrasonik SRF04	5 buah	Rp 350.000,00	Rp 1.750.000,00
		SHARP GP2Y0A21YK0F	1 buah	Rp 200.000,00	Rp 200.000,00
3.	Sensor <i>Infrared</i>	<i>Proximity</i>	2 buah	Rp 80.000,00	Rp 160.000,00
		<i>Opto Coupler</i>	1 buah	Rp 100.000,00	Rp 100.000,00
4.	Sensor <i>Encoder</i>	<i>Autonics Rotary Encoder E40H8-3600-3-N-24</i>	1 buah	Rp 800.000,00	Rp 800.000,00
5.	Sensor Warna	Led dan LDR	1 pasang	Rp 30.000,00	Rp 30.000,00
6.	Switch	<i>Limit switch</i> Dekko	7 buah	Rp 10.000,00	Rp 70.000,00
7.	Modul Micro SD	Modul <i>Micro SD Adapter</i>	1 buah	Rp 35.000,00	Rp 35.000,00
8.	Mikrokontroler	Arduino Nano 328P	2 buah	Rp 51.000,00	Rp 102.000,00
		Arduino Mega 2560	1 buah	Rp 300.000,00	Rp 300.000,00
		Wemos D1	1 buah	Rp 120.000,00	Rp 120.000,00
9.	Motor DC	Motor DC <i>Gearbox</i> 12 V 2 A 400 rpm	2 buah	Rp 120.000,00	Rp 240.000,00
		<i>Power window</i> 12 V 5 A	1 buah	Rp 125.000,00	Rp 125.000,00
10.	LCD <i>Display</i>	LCD 16x2	1 buah	Rp 50.000,00	Rp 50.000,00
		LCD 20x4	1 buah	Rp 120.000,00	Rp 120.000,00
11.	<i>Driver</i> Motor	H- <i>Bridge</i> Mosfet	1 buah	Rp 150.000,00	Rp 150.000,00
		H- <i>Bridge</i> Relay	2 buah	Rp 35.000,00	Rp 70.000,00
12.	Motor Servo	Tower-Pro 180 derajat	1 buah	Rp 120.000,00	Rp 120.000,00
13.	LED <i>Flash</i>		1 buah	Rp 135.000,00	Rp 135.000,00
14.	<i>Micro SD</i>	Sandisk 2 Gb	1 buah	Rp 45.000,00	Rp 45.000,00
15.	<i>Bluetooth</i>	HC-05	4 buah	Rp 55.000,00	Rp 220.000,00
16.	<i>Converter</i>	Modul LCD to I2C	1 buah	Rp 35.000,00	Rp 35.000,00
17.	Baterai	Lipo 3 cell 2200 mAh	1 buah	Rp 350.000,00	Rp 350.000,00
		Lipo 2 cell 850 mAh	1 buah	Rp 175.000,00	Rp 175.000,00
18.	<i>Power supply</i>	12V 5A <i>Switching</i>	1 buah	Rp 65.000,00	Rp 65.000,00
		12V 2A <i>Switching</i>	1 buah	Rp 55.000,00	Rp 55.000,00
19.	<i>Step Down</i>	XL4015 DC to DC <i>Step Down</i> Module 5A 75W	3 buah	Rp 35.000,00	Rp 105.000,00
20.	Router	Sierra 4G <i>Wireless</i> Modem Wifi	1 buah	Rp 600.000,00	Rp 600.000,00
	Jumlah				Rp 6.832.000,00

