



***AUTOMATIC STOPING AND INFUSION MONITORING WITH
TELEMETRY SYSTEM BASED ON ANDROID***

PROYEK AKHIR

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



OLEH:

MOHAMAD SIROJUL AZIIS

NIM. 14507134011

**PROGRAM STUDI TEKNIK ELEKTRONIKA
JURUSAN PENDIDIKAN TEKNIK ELEKTRONIKA DAN INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2018**

LEMBAR PERSETUJUAN

PROYEK AKHIR

***AUTOMATIC STOPING AND INFUSION MONITORING WITH
TELEMETRY SYSTEM BASED ON ANDROID***

Oleh

MOHAMAD SIROJUL AZIIS

14507134011

Telah diperiksa dan disetujui oleh pembimbing:

Untuk diuji

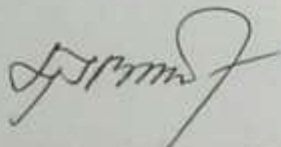
Yogyakarta, 22 Februari 2018

Mengetahui,

Kaprodi Teknik Elektronika

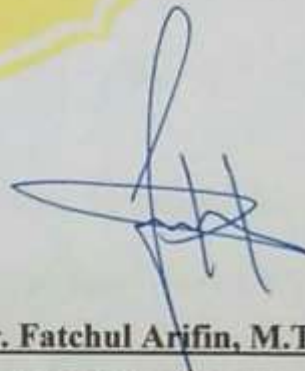
Menyetujui,

Pembimbing Proyek Akhir



Dr. Sri Waluyanti, M.Pd.

NIP. 19581218 198603 2 001



Dr. Fatchul Arifin, M.T.

NIP. 19720508 198802 1 002

LEMBAR PENGESAHAN

PROYEK AKHIR

*AUTOMATIC STOPING AND INFUSION MONITORING WITH
TELEMETRY SYSTEM BASED ON ANDROID*

Dipersiapkan dan Disusun Oleh:

MOHAMAD SIROJUL AZIIS

14507134011

Telah dipertahankan di depan 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

Nama	Jabatan	Tanda Tangan	Tanggal
1. Dr. Fatchul Arifin, M.T.	Ketua Penguji		22/02-2018
2. Bekti Wulandari, S.Pd.T., M.Pd.	Sekretaris Penguji		21 Feb 18
3. Dr. Dra. Umi Rochayati, M.T.	Penguji Utama		21 Feb 2018

Yogyakarta, 22 Februari 2018

Dekan Fakultas Teknik UNY



Dr. Widarto, M.Pd.

NIP. 19631230 198812 1 001

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : Mohamad Sirojul Aziis

NIM : 14507134011

Program Studi : Teknik Elektronika D-III

Judul Proyek Akhir : *Automatic Stopping and Infusion Monitoring with Telemetry System based on Android*

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 Februari 2018

Yang menyatakan,



Mohamad Sirojul Aziis
NIM. 14507134011

MOTTO

Susah mendapatkannya mudah melepaskannya yaitu,

“WAKTU”

*Semua perilaku baik atau buruk pasti akan ada waktu didalamnya
Bergerak, diam, sedih, senang, bernafas, samapi mati juga ada waktu.*

“manfaatkanlah waktu jangan terbuang sia-sia”

*Dan Dia (pula) yang menjadikan malam dan siang silih berganti bagi orang yang
ingin mengambil pelajaran atau orang yang ingin bersyukur.*

(Al Furqan : 62)

LEMBAR PERSEMBAHAN

Kutunjukkan dan kupersembahkan karyaku kepada orang yang telah membimbing, mendoakan, menasehati, memberi kasih sayang, serta mencukupi semua kebutuhanku. Kepada segenap keluargaku sebagai tanda bakti, hormat, dan rasa terima kasih yang tidak terhingga hingga akhir masa. Setumpuk lembaran ini semoga dapat menjadikan sepercik kebahagiaan bagi kedua orang tuaku serta keluargaku. Terima kasih kepada orang tuaku yang selalu memotivasi saya setiap saat.

Kepada dia, teman-temanku di kelas B Teknik Elektronika 2014, seluruh teman-teman di UNY, serta teman-teman terdekatku karena berkat kalian semua telah membimbingku kedalam diri yang lebih dewasa. Mengiringi kebahagiaan setiap saat yang kalian berikan kepadaku. Mengubah sedih menjadi senyum, mengubah tangis menjadi tawa, dan selalu membagi ilmu kepadaku. Semoga Allah selalu melindungi serta memberikan barokahnya kepada kalian semua.

Seluruh Dosen Pengajar Teknik Elektronika beserta staff karyawannya, terima kasih kalian semua telah meluangkan waktu untuk mendidik dan membagikan ilmu kalian. Ilmu dan pengalaman yang kalian berikan sangatlah bermanfaat bagiku. Kepada dosen pembimbing saya ucapkan terima kasih atas waktu, bimbingan, dan bantuannya selama ini. Beribu-ribu kata terima kasih tidak akan mengembalikan ilmu dan waktu yang telah engkau berikan. Semoga Allah selalu mencurahkan rahmat dan hidayahnya pada kalian semua serta melindungi kalian semua dari hal yang tidak diinginkan.

PROYEK AKHIR

AUTOMATIC STOPING AND INFUSION MONITORING WITH TELEMETRY SYSTEM BASED ON ANDROID

Oleh : Mohamad Sirojul Aziiss

NIM : 14507134011

ABSTRAK

Kegiatan perawat atau dokter yang banyak mengakibatkan pemantauan terhadap infus menjadi kurang. Selain itu, kelalaian juga berpengaruh terhadap pemantauan infus. Pemantauan infus sangat bermanfaat pada pasien karena untuk menjaga supaya infus tidak macet, tidak terisi oleh udara, dan habis. Infus yang dipergunakan sampai habis dapat menjadi masalah yang besar yaitu darah pada pasien menjadi naik ke botol infus sehingga dapat menyebabkan pasien menjadi kekurangan cairan. Sehingga untuk mengatasinya diperlukan pemantau infus dan penghenti infus secara tepat dan cepat. Tujuan pembuatan alat ini adalah untuk merancang dan mengetahui unjuk kerja dari rancangan.

Pembuatan alat terdiri dari 8 macam bagian yang memiliki fungsi masing-masing. Bagian-bagian tersebut adalah, tiang infus yang terbuat dari *stainless steel* dan alumunium untuk memudahkan dalam menurunkan atau menaikkan tiang infus pada ketinggian tertentu, *box* sensor tetesan terbuat dari akrilik yang berada dibagian *drip chamber*, penghenti infus dengan motor servo yang didesain sehingga dapat menekuk selang supaya terhenti, Arduino Mega 2560 sebagai kontrol utama sistem, sensor tetesan terdiri atas sensor cahaya photodioda dan laser dioda, *keypad* 4x4 berfungsi memilih infus dan memasukkan nomor telepon atau batas infus, Wemos D1 Mini sebagai pengirim data ke *website* dan aplikasi android, serta LCD *display*, *website*, dan aplikasi android sebagai penampil hasil jumlah tetesan.

Dari hasil pengujian unjuk kerja yang telah dilakukan dengan objek pengujian sejumlah 5 kali pada masing-masing bagian, didapatkan hasil bahwa kualitas alat ini dapat diimplementasikan dan digunakan dengan baik. Sementara secara kuantitatif diketahui bahwa rata-rata kesalahan pada sensor tetesan infus 1 adalah 3,5% dan pada sensor tetesan infus 2 sebesar 2,52%. Penampilan data di LCD, *website*, serta aplikasi android tidak mengalami masalah dengan nilai *error* adalah 0%. Penghenti infus serta sistem pemberitahuan lewat panggilan telepon juga berfungsi dengan tepat dan memiliki nilai *error* sebesar 0%. Kesalahan jumlah tetesan disebabkan oleh faktor sensor yang mudah terpengaruh oleh cahaya disekitar sehingga pembacaan nilai ADC menjadi kurang tepat.

Kata Kunci : *Monitoring, Infusion, Telemetry, Android, website*

KATA PENGANTAR

Assamu'alaikum wr. wb.

Segala Puji dan syukur penulis panjatkan kehadirat Allah SWT dengan rahmat dan hidayah-Nya Laporan Proyek Akhir ini dapat terselesaikan tanpa halangan yang berarti. Sholawat serta salam tercurah pada Nabi Agung kita Rasulullah Muhammad SAW dan 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. Dr. Fatchul Arifin, M.T. selaku Ketua Jurusan Pendidikan Teknik Elektronika dan Informatika Universitas Negeri Yogyakarta dan sekaligus dosen pembimbing Penyusunan Laporan Proyek Akhir.
2. Dr. Sri Waluyanti, M.Pd. selaku Ketua Program Studi Diploma III, Koordinator Proyek Akhir Jurusan Pendidikan Teknik Elektronika dan Informatika Universitas Negeri Yogyakarta.
3. Drs. Totok Sukardiyono, M.T. selaku Sekertaris Jurusan Pendidikan Teknik Elektronika dan Informatika Universitas Negeri Yogyakarta.
4. Seluruh Dosen Pengajar Teknik Elektronika Universitas Negeri Yogyakarta atas bekal ilmu yang diberikan kepada penulis.
5. Dr. Widarto, M.Pd. selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.

6. Semua pihak yang tidak dapat saya sebutkan satu persatu yang telah membantu hingga terselesainya laporan ini.

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, 22 Februari 2018

Penulis,

Mohamad Sirojul Aziis

DAFTAR ISI

Halaman

PROYEK AKHIR	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR PERNYATAAN KEASLIAN	iv
MOTTO	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xvii
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Identifikasi Masalah	4
C. Batasan Masalah.....	5
D. Rumusan Masalah	5
E. Tujuan.....	5
1. Bagi Mahasiswa.....	6
2. Bagi Jurusan Pendidikan Teknik Elektronika.....	6
3. Bagi Masyarakat	6
G. Keaslian Gagasan	6
BAB II PENDEKATAN PEMECAHAN MASALAH	8
A. Pemantauan Kesehatan.....	8
B. Pasien.....	9
C. Terapi Intravena (Infus)	10
1. Tujuan Pemasangan Infus	11
2. Pedoman Pemilihan Daerah Inseri Pemasangan Infus	12
3. Tipe-tipe Cairan Intravena	12

4.	Komposisi Cairan Terapi Intravena	14
5.	Tipe-tipe Pemberian Terapi Intravena.....	14
6.	<i>Monitoring</i> Infus	15
7.	Penentuan Kecepatan Cairan Intravena (Infus)	17
8.	Komplikasi Terapi Intravena.....	18
D.	Komunikasi Serial.....	18
1.	Komunikasi Serial <i>Synchronous</i> dan <i>Asynchronous</i>	19
2.	<i>Data Transfer Rate</i>	22
E.	Komunikasi IIC (<i>Inter Integrated Circuit</i>)	22
F.	Mikrokontroler.....	24
G.	Arduino	26
1.	Arduino Mega 2560	27
2.	Arduino IDE	28
H.	Modul Wi-Fi	32
1.	<i>Chipset</i> ESP8266	33
2.	<i>Chipset</i> CH340	34
I.	<i>ThingSpeak</i>	34
J.	Sensor Tetesan Infus	35
1.	Photodiode.....	36
2.	Laser diode	38
K.	Modul GSM/GPRS SIM800L	40
L.	Motor Servo	46
1.	Konstruksi Motor Servo	46
2.	Jenis-Jenis Motor Servo	48
M.	LCD (<i>Liquid Crystal Display</i>).....	53
N.	<i>Keypad</i>	56
O.	LM2596	57
P.	Catu Daya	58
BAB III KONSEP RANCANGAN		61
A.	Identifikasi Kebutuhan	62
1.	Bagian Catu Daya	62

2.	Bagian Masukan	62
3.	Bagian Proses	63
4.	Bagian Keluaran	63
B.	Analisis Kebutuhan	63
1.	Bagian Catu Daya	63
2.	Bagian Masukan	64
3.	Bagian Proses	64
4.	Bagian Keluaran	65
C.	Blok Diagram	67
D.	Perancangan Sistem.....	68
1.	Blok Rangkaian Catu Daya.....	68
2.	Blok Alur Rangkaian Mikrokontroler	70
E.	Langkah Pembuatan Alat	71
1.	Pembuatan PCB	72
2.	Pemasangan Komponen.....	72
3.	Perancangan Tempat Sensor Tetes Infus.....	72
4.	Pemasangan Rangkaian pada <i>Box Control</i>	73
5.	Perancangan <i>Box</i>	74
6.	Pemasangan Mekanik <i>Box Chamber</i> Infus dan Penghenti Infus	75
F.	Perangkat Lunak	76
1.	Arduino IDE	76
2.	Algoritma Program	77
G.	<i>Flow Chart</i>	80
H.	Spesifikasi Alat	82
I.	Pengujian Alat.....	83
K.	Pengoperasian Alat.....	84
BAB IV PENGUJIAN DAN PEMBAHASAN.....		86
A.	Hasil Pengujian	86
1.	Hasil Pengujian Tegangan Catu Daya.....	86
2.	Hasil Pengujian Modul Sensor Tetesan.....	87
3.	Hasil Pengujian Penghenti Laju Infus	88

4.	Hasil Pengujian Sistem Notifikasi Melalui Panggilan.....	89
5.	Hasil Pengujian Tampilan Tetesan.....	90
6.	Hasil Pengujian Jangkauan Transmisi Data Wemos D1 Mini	91
7.	Hasil Pengujian LCD	91
8.	Hasil Pengujian Unjuk Kerja	92
B.	Pembahasan	96
1.	Analisa Tegangan Catu Daya.....	96
2.	Analisa Pengujian Sensor Tetesan	97
3.	Analisa Pengujian Penghenti Laju Infus	98
4.	Analisa Pengujian Sistem Pemberitahuan Melalui Panggilan	100
5.	Analisa Pengujian Tampilan Tetesan	101
6.	Analisa Pengujian Jangkauan Transmisi Data.....	102
7.	Analisa Pengujian LCD	103
8.	Analisa Pengujian Unjuk Kerja.....	103
BAB V KESIMPULAN DAN SARAN.....		108
A.	Kesimpulan	108
B.	Keterbatasan Alat.....	109
C.	Penelitian Lanjutan.....	110
DAFTAR PUSTAKA.....		112
LAMPIRAN		115

DAFTAR TABEL

Tabel 1. Tipe Data Bahasa C	30
Tabel 2. Operasi Kondisi	31
Tabel 3. Jenis- Jenis Perintah AT- <i>Command</i>	45
Tabel 4. Operasi Dasar LCD.....	55
Tabel 5. <i>Pinout</i> LCD 20x4.....	55
Tabel 6. Karakteristik IC Regulator	60
Tabel 7. Kebutuhan Komponen	66
Tabel 8. Hasil Pengujian Tegangan Catu Daya Tanpa Beban	86
Tabel 9. Hasil Pengujian Tegangan Catu Daya dengan Beban.....	87
Tabel 10. Hasil Pengujian Sensor Tetesan Infus 1	88
Tabel 11. Hasil Pengujian Sensor Tetesan Infus 2	88
Tabel 12. Hasil Pengujian Penghenti Laju Infus 1	89
Tabel 13. Hasil Pengujian Penghenti Laju Infus 2	89
Tabel 14. Hasil Pengujian Sistem Notifikasi Melalui Panggilan Pada Infus 1	89
Tabel 15. Hasil Pengujian Sistem Notifikasi Melalui Panggilan Pada Infus 2	90
Tabel 16. Hasil Pengujian Tampilan LCD dengan website Thingspeak.....	90
Tabel 17. Hasil Pengujian Tampilan LCD dengan aplikasi Thingview	90
Tabel 18. Hasil Pengujian Tampilan ThingSpeak dengan ThingView	91
Tabel 19. Hasil Pengujian Jangkauan Transmisi Data	91
Tabel 20. Hasil Pengujian LCD <i>Display</i>	91
Tabel 21. Hasil Pengujian sensor photodiode dan laser diode Pada Infus 1	93
Tabel 22. Hasil Pengujian sensor photodiode dan laser diode Pada Infus 2	93
Tabel 23. Hasil Pengujian Motor Servo Pada Infus 1.....	94
Tabel 24. Hasil Pengujian Motor Servo Pada Infus 2.....	94
Tabel 25. Hasil Pengujian Modul SIM800L Pada Infus 1	94
Tabel 26. Hasil Pengujian Modul SIM800L Pada Infus 2.....	95
Tabel 27. Hasil Pengujian Unjuk Kerja Tampilan pada website ThingSpeak	95
Tabel 28. Hasil Pengujian Unjuk Kerja Tampilan ThingView	95

DAFTAR GAMBAR

Gambar 1. Pasien Rumah Sakit.....	9
Gambar 2. Pemasangan Infus dengan Suntikkan	10
Gambar 3. Pemasangan Infus dengan Luka Sayat.....	11
Gambar 4. Macam-Macam Jenis Infus.....	13
Gambar 5. Sinyal Transmisi Asinkron.....	20
Gambar 6. Pembesaran Karakter ASCII “A” (41h)	21
Gambar 7. Aliran Data IIC.....	23
Gambar 8. Bentuk Fisik Modul I2C.....	23
Gambar 9. Arduino Mega 2560	28
Gambar 10. Tampilan Arduino IDE.....	29
Gambar 11. Bentuk Fisik Wemos D1 Mini.....	33
Gambar 12. Sensor Cahaya Photodiode.....	37
Gambar 13. Spektrum Panjang Gelombang Sinar Laser	38
Gambar 14. Bentuk Fisik Modul GSM SIM800L	40
Gambar 15. Skematik dari Sistem GSM	41
Gambar 16. Motor Servo DC.....	46
Gambar 17. Pin <i>out</i> Motor Servo	47
Gambar 18. Konstruksi Motor Servo.....	48
Gambar 19. Teknik Pengaturan Sudut Pada Motor Servo.....	49
Gambar 20. Arah Putaran Motor Servo Standar.....	50
Gambar 21. Arah Putaran Motor Servo <i>Continuous</i> 360 ⁰	51
Gambar 22. Pulsa Pengendalian Motor Servo Rotasi Sudut.....	52
Gambar 23. LCD Karakter 20x4.....	54
Gambar 24. Rangkaian <i>Keypad</i> dengan Arduino UNO.....	56
Gambar 25. <i>Switching Power Supply</i> LM2596	57
Gambar 26. Catu Daya Linier	59
Gambar 27. Catu Daya <i>Switching</i>	59
Gambar 28. Diagram Blok Sistem	61
Gambar 29. Blok Diagram Keseluruhan	67

Gambar 30. Rangkaian <i>Switching Power Supply</i> 12 Volt	68
Gambar 31. Rangkaian <i>Step Down LM2596 Adjustable</i>	69
Gambar 32. Alur Rangkaian Mikrokontroler	70
Gambar 33. Bentuk Fisik Tempat Sensor Tetes Infus	73
Gambar 34. Bentuk Fisik <i>Box</i> Penghenti Laju Infus	74
Gambar 35. Bentuk Fisik <i>Box Control System</i>	75
Gambar 36. Desain Alat Penghitung Jumlah Tetesan dan.....	75
Gambar 37. Tampilan dari <i>Software</i> Arduino IDE.....	77
Gambar 38. <i>Flowchart</i> Program bagian Pertama	80
Gambar 39. <i>Flowchart</i> Program bagian Kedua.....	81
Gambar 40. <i>Flow Chart</i> Program bagian Ketiga.....	82
Gambar 41. Hasil Pengujian Tetesan Sensor 1.....	97
Gambar 42. Hasil Pengujian Tetesan Sensor 2.....	97
Gambar 43. Hasil Pengujian Penghenti Laju Infus 1.....	98
Gambar 44. Hasil Pengujian Penghenti Laju Infus 2.....	99
Gambar 45. Hasil Pengujian Sistem Notifikasi Melalui Panggilan 1	100
Gambar 46. Hasil Pengujian Sistem Notifikasi Melalui Panggilan Infus 2	100
Gambar 47. Hasil Pengujian LCD dengan ThingSpeak	101
Gambar 48. Hasil Pengujian Tampilan LCD dengan Thingview	101
Gambar 49. Hasil Pengujian Tampilan ThingSpeak dengan ThingView	102
Gambar 50. Hasil Pengujian Tetesan Pada Sensor 1	104
Gambar 51. Hasil Pengujian Tetesan Pada Sensor 2	104
Gambar 52. Hasil Pengujian Penghentian Laju Infus 1	105
Gambar 53. Hasil Pengujian Penghentian Laju Infus 2.....	105
Gambar 54. Hasil Pengujian Sistem Notifikasi Melalui Panggilan Infus 1	106
Gambar 55. Hasil Pengujian Sistem Notifikasi Melalui Panggilan Infus 2	106

DAFTAR LAMPIRAN

Lampiran 1. Foto Alat.....	116
Lampiran 2. Program Alat	121
Lampiran 3. Skema Rangkaian Alat	163
Lampiran 4. Skema Rangkaian Sensor Tetesan	164
Lampiran 5. <i>Layout Shield Arduino Mega</i>	165
Lampiran 6. <i>Datasheet</i> Arduino Mega 2560.....	163
Lampiran 7. <i>Datasheet</i> Motor Servo MG995.....	170
Lampiran 8. <i>Datasheet</i> SIM800L V2.....	171
Lampiran 9. <i>Datasheet</i> Wemos D1 Mini	173
Lampiran 10. <i>Datasheet</i> Keypad 4x4.....	175
Lampiran 11. <i>Datasheet</i> LM2596	177
Lampiran 12. <i>Datasheet</i> LCD 20x4.....	180
Lampiran 13. <i>Datasheet</i> Laser dioda	182
Lampiran 14. <i>Datasheet</i> Photodioda.....	184

BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi di dunia kedokteran saat ini banyak mengalami kemajuan yang sangat pesat. Alat-alat kesehatan pada zaman sekarang sudah dikombinasi dengan berbagai *Personal Computer* (PC) dan *Smartphone*. Salah satu contohnya yaitu alat *ultrasonografi* yang pada dahulu masih dua dimensi (2D) sekarang sudah menjadi tiga dimensi (3D) bahkan sudah ada yang empat dimensi (4D). Kemajuan teknologi seperti ini sangat dibutuhkan dalam kedokteran untuk menunjang kemampuan manusia yang terbatas. Namun, ada salah satu alat kedokteran yang belum mengalami kemajuan dalam bidang teknologi dari zaman dahulu sampai sekarang yaitu infus.

Infus adalah suatu cairan kimia yang disuntikkan ketubuh manusia melalui rute *intravena* dalam beberapa waktu tertentu. Pemberian cairan infus umumnya diberikan bagi seseorang yang mengalami kekurangan cairan dalam tubuh atau membutuhkan pengobatan. Infus merupakan pengobatan sangat tepat dalam kedokteran karena mudah terserap oleh sel-sel tubuh manusia dikarenakan bentuknya yang cair. Terkadang infus juga digunakan sebagai pengganti makanan atau minuman untuk pasien. Selain itu pemasukan cairannya yang melalui pembuluh darah *vena* mudah terserap oleh tubuh daripada melalui mulut yang harus mengalami proses pencernaan terlebih dahulu.

Penggunaan infus juga sangat sederhana dan mudah. Caranya dengan menggantung botol infus dan memanfaatkan grafitasi bumi maka, tetesan infus akan mengalir melalui selangnya. Kemudian cairan infus akan masuk ke pembuluh darah *vena* jika adanya beda tekanan antara botol dengan pembuluh darah. Walau sangat sederhana dan penting dalam kedokteran infus masih memiliki beberapa kelemahan. Pertama adalah dalam sistem pemantauan cairan infus yang masih kurang dan yang kedua adalah jika infus habis maka klemnya harus segera ditutup supaya tidak adanya pembuluh darah yang naik keselang infus. Bahkan jika kejadian infus habis dibiarkan lama-kelamaan maka darah akan mengisi kantong infus tentunya hal ini sangat berbahaya bagi pasien. Dimana pasien yang sedang berobat seharusnya memperoleh kesehatan dari infus, sehingga mendapat penyakit bertambah.

Pemasangan infus ini biasanya dilakukan oleh perawat. Saat perawat memasang infus kepada pasien mereka mencatat tetesan setiap menit yang terdapat pada bagian *drip chambers* dengan melihat secara langsung, untuk kemudian disesuaikan dengan kebutuhan pasien. Setelah mengetahui semuanya maka perawat akan mengitungnya dan mendapatkan pada jam berapa infus akan habis. Selain itu biasanya perawat sering memantau cairan infus dengan melihat secara langsung infus tersebut. Jadi dibutuhkan kesigapan setiap saat dari para perawat dan dokter untuk segera mengganti infus yang habis atau bisa juga jika dengan pengecekan berkala saat infus digunakan. Tetapi masih ada juga pasien yang diketahui sampai kehabisan infus sehingga darah sampai masuk ke kantong infus. Kejadian kehabisan

infus biasanya disebabkan oleh perawat atau dokter yang ceroboh atau salah perhitungan dalam tetesan.

Penggunaan infus yang sudah otomatis masih jarang digunakan di rumah sakit menjadi permasalahan dalam pemantauannya. Harga infus otomatis yang dijual secara umum masih sangat mahal. Infus otomatis juga masih dipantau secara manual melalui LCD, belum terintegrasi dengan sistem telemetri. Selain itu sistem pemberitahuan serta pemberhenti saat infus habis atau macet jarang yang memiliki dan penggunaan infus otomatis yang sulit menjadi permasalahan bagi tenaga medis.

Mendapati permasalahan tersebut dibutuhkanlah alat yang dapat memberikan pertolongan pertama jika terjadi kehabisan infus atau jika terjadi *human error* oleh perawat atau dokter. Alat yang diberi nama “*automatic stoping and infusion monitoring with telemetry system based on android*” memiliki dua fungsi yaitu untuk pemantauan tetesan infus melalui website atau aplikasi dan memberikan indikasi serta menghentikan laju infus saat infus habis. Alat ini menggunakan sensor cahaya yaitu photodioda dan laser dioda. Selain menggunakan sensor tersebut, alat ini juga menggunakan mikrokontroler Arduino Mega ATmega 2560, modul GSM/GPRS, modul wifi Wemos D1 Mini, dan motor servo. Sensor photodioda dan laser dioda berfungsi mengetahui jumlah tetesan setiap saatnya. Selain untuk menghitung jumlah tetesan infus juga berfungsi mendeteksi saat infus macet dan habis.

Sensor photodioda dan laser dioda ini terhubung ke Arduino Mega ATmega 2560 sebagai pengolah datanya tetesan infus. Selanjutnya Arduino Mega ATmega 2560 dikoneksikan dengan motor servo dan modul GSM/GPRS SIM 800L. Motor

servo berfungsi sebagai penghenti laju cairan infus, sedangkan modul GSM/GPRS SIM 800L sebagai sistem yang akan memberitahukan melalui *miscall* ke perawat atau dokter. *Keypad* 4x4 digunakan untuk kemudahan dalam memasukkan nomor telepon yang menjadi tanda saat infus habis atau mengalami kemacetan. *Lyquid Crystal Display* (LCD) berfungsi dalam penampilan saat mengatur alat. Data serial dari Arduino Mega 2560 dikirim ke Wemos D1 Mini sebagai sistem telemetri yang kemudian, data dari sensor tersebut ditampilkan website dan aplikasi android sehingga memudahkan dalam pemantauanya.

B. Identifikasi Masalah

Berdasarkan uraian latar belakang di atas, maka dapat diidentifikasi beberapa permasalahan yang ada sebagai berikut:

1. Banyaknya kasus darah naik menuju kantong infus saat infus mengalami kehabisan cairan.
2. Kurangnya efisiensi dan akurasi dalam pemantauan cairan infus pada pasien secara manual.
3. Masih minimnya sistem penghentian infus yang dapat berfungsi secara otomatis.
4. Kurangnya alat pemantau jumlah cairan infus yang terintegrasi dengan sistem telemetri.
5. Perlunya sistem pemberitahuan saat infus mengalami kemacetan atau habis..
6. Perhitungan jumlah tetesan yang kurang tepat oleh perawat atau dokter.
7. Masih sulitnya dalam mendapatkan infus otomatis dan harganya yang mahal.
8. Sulitnya penggunaan infus otomatis yang sudah ada.

C. Batasan Masalah

Pembatasan masalah pada proyek akhir ini bertujuan supaya alat yang dikembangkan lebih terfokus pada pokok masalah yang ada. Masalah yang ditangani adalah pada poin 3, 4, dan 5. Alat ini terdiri atas sensor tetesan yang keseluruhan kerjanya tersebut dikendalikan oleh mikrokontroler. Hasil pengolahan tersebut ditampilkan pada media penampil, *website*, dan aplikasi. Selain itu, pada alat ini juga memiliki sistem penghentian selang infus dengan cara menekuk. Alat ini juga memiliki sistem pemberitahuan ke tenaga medis untuk dilakukan penggantian infus pada pasien disebabkan infus macet atau cairan telah habis.

D. Rumusan Masalah

Berdasarkan latar belakang masalah, maka dapat dirumuskan permasalahan sebagai berikut :

1. Bagaimana rancang bangun *automatic stoping and infusion monitoring with telemetry system based on android* ?
2. Bagaimana unjuk kerja *automatic stoping and infusion monitoring with telemetry system based on android* ?

E. Tujuan

Adapun tujuan dari penulis karya ini adalah :

1. Merealisasikan rancang bangun *automatic stoping and infusion monitoring with telemetry system based on android*.

2. Mengetahui unjuk kerja dari *automatic stoping and infusion monitoring with telemetry system based on android*.

F. Manfaat

Pembuatan proyek akhir ini diharapkan dapat bermanfaat, Adapun manfaat yang diharapkan dari pembuatan tugas akhir ini antara lain :

1. Bagi Mahasiswa

Dapat menambah wawasan, pengetahuan, dan kepedulian mahasiswa dalam bidang yang diteliti serta sebagai pengaplikasian ilmu yang telah didapatkan selama menempuh pendidikan di bangku perkuliahan.

2. Bagi Jurusan Pendidikan Teknik Elektronika

Dapat bermanfaat menambah alat yang inovatif dalam dunia kedokteran serta sebagai wujud dari perkembangan Ilmu Pengetahuan dan Teknologi (IPTEK) dalam dunia pendidikan.

3. Bagi Masyarakat

Dapat diterapkan di pusat-pusat pelayanan kesehatan masyarakat sehingga dapat meningkatkan kualitas kesehatan serta menurunkan jumlah angka kematian di pusat kesehatan masyarakat akibat indikasi kehabisan infus pasien dapat teratasi secara dini.

G. Keaslian Gagasan

Berikut ini beberapa penelitian relevan yang dapat dijadikan acuan untuk proyek akhir ini diantaranya sebagai berikut:

1. Rancang Bangun Alat Pengaturan Jumlah Tetesan Infus pada Pasien dan *Monitoring* Jarak Jauh dengan PC, Bagus Kokoh S.A, Politeknik Elektronika Negeri Surabaya, 2009. Pada alat ini mikrokontroler ATmega 16 digunakan sebagai pembangkit PWM untuk kontrol motor servo serta photodiode dan *infrared* sebagai sensor pendeteksi adanya tetesan cairan infus. Persamaan dengan karya ini adalah penggunaan motor servo sebagai penghenti infus. Sedangkan perbedaan karya ini dengan penelitian yang dilakukan adalah terletak pada jenis sensor dan mikrokontroler yang digunakan. Alat yang dibuat ini juga memiliki sistem pemberitahuan saat habis atau macet dengan panggilan. Pemantauannya dapat melalui website serta aplikasi android.
2. *Design and Implementation Of A Microcontroller Based Infusion With Bluetooth Wireless Remote Monitoring* karya Rana Riyadh Saeed dari *Near East University*, 2013. Alat ini menggunakan sensor photodiode dan *infrared* untuk membaca sinyal analog pada tetesan infus. Sinyal tersebut kemudian dikirimkan melalui *bluetooth* ke dalam *chip* mikrokontroler, lalu dihubungkan dengan PC untuk memantaunya. Selain itu, pada alat ini juga dilengkapi dengan alarm dan sistem penghentian laju cairan infus saat mendekati batas minimum. Sistem penghentian ini dibuat dengan menggunakan motor servo yang didesain berbentuk menyerupai sebuah pintu. Persamaan pada karya ini dengan penelitian yang dilakukan adalah sistem penghentian laju infus menggunakan motor servo. Sedangkan perbedaannya yaitu terletak pada jenis sensor tetesan, sistem pemantaunya, desain sistem penghenti infus, dan jenis *chip* mikrokontroler yang digunakan.

BAB II

PENDEKATAN PEMECAHAN MASALAH

A. Pemantauan Kesehatan

Kesehatan adalah keadaan sejahtera dari badan, jiwa, dan sosial yang memungkinkan setiap orang hidup produktif secara sosial dan ekonomis. Menurut Undang-Undang Nomor 36 Tahun 2009 Tentang Kesehatan, kesehatan merupakan suatu keadaan sehat baik secara fisik, mental, spiritual, maupun sosial yang memungkinkan setiap orang untuk hidup produktif secara sosial dan ekonomis. Sedangkan pemantauan menurut berdasarkan Peraturan Pemerintah Nomor 39 Tahun 2006 merupakan suatu kegiatan mengamati secara seksama suatu keadaan atau kondisi, termasuk juga perilaku atau kegiatan tertentu dengan tujuan agar semua data masukan atau informasi yang diperoleh dari hasil pengamatan tersebut dapat dijadikan sebagai landasan dalam pengambilan keputusan.

Oleh sebab itu, pemantauan kesehatan merupakan suatu kegiatan mengamati kondisi seseorang atau pasien dengan tujuan untuk mengetahui kondisi seseorang, sehingga dapat diberikan tindak lanjut dari hasil pemantauan kesehatan tersebut. Secara umum, kegiatan pemantauan bertujuan untuk mengumpulkan data dan informasi yang diperlukan dan mendapatkan gambaran ketercapaian tujuan setelah adanya kegiatan (Daman, 2012). Salah satu kegiatan pemantauan kesehatan yaitu pada pemantauan cairan infus di Rumah Sakit.

B. Pasien

Menurut Johnson dalam Christensen dan Kenney (2009), pasien adalah seseorang yang sedang mengalami gangguan atau ketidakseimbangan kesehatannya. Sedangkan King (2009) mengatakan bahwa pasien adalah individu (sistem personal) yang tidak mampu mengatasi peristiwa atau masalah kesehatan ketika berinteraksi dengan lingkungan. Sementara menurut istilah pasien yang berasal dari bahasa latin yaitu orang yang menerima perawatan. Sehingga disimpulkan bahwa pasien merupakan seseorang yang sedang menerima perawatan akibat terjadinya gangguan kesehatan yang menyerangnya.

Pasien dianggap sebagai manusia yang memiliki kebutuhan tertentu, yang umumnya memiliki sifat mendasar untuk semua orang dan membutuhkan pemenuhan yang sesuai. Kebutuhan pasien yang mendasar saat dirawat di rumah sakit biasanya adalah infus. Berikut ini salah satu contoh pasien yang sedang dirawat di rumah sakit yang dapat dilihat pada Gambar 1 sebagai berikut.



Gambar 1. Pasien Rumah Sakit

(Simanjuntak, 2017)

C. Terapi Intravena (Infus)

Pemberian cairan *intravena* (infus) adalah memasukan cairan atau obat langsung ke dalam pembuluh darah *vena* dalam jumlah dan waktu tertentu dengan menggunakan infus set (Perry, 2005). Teknik penusukan *vena* melalui *transkutan* dengan *stilet* yang kaku, seperti *angiokateter* atau dengan jarum yang disambungkan. Terapi *intravena* atau yang biasa disebut dengan terapi infus merupakan metode yang efektif untuk mensuplai cairan, elektrolit, nutrisi, dan obat melalui pembuluh darah atau *intravaskular* (Mubarak, 2008). Kateterisasi *vena* adalah pembuatan jalur vena untuk pemberian cairan, darah atau obat, dan suntikan berulang (Mansjoer, 2000).

Pemberian cairan *intravena* adalah pemberian cairan atau darah langsung ke dalam *vena* yang dapat dikerjakan dengan dua cara yaitu tanpa membuat luka sayat dan dengan membuat sayatan. Teknik tanpa sayatan dilakukan dengan cara menusukkan jarum infus (ujung tajam) langsung ke dalam *vena*. Sedangkan dengan teknik sayatan dilakukan dengan cara menyayat sedikit kulit untuk mencari jalur *vena* dan melubanginya serta setelah itu memasukkan jarum infus. Adapun teknik pemberian cairan *intravena* tanpa sayatan dapat dilihat pada Gambar 2, sedangkan dengan teknik sayatan dapat dilihat pada Gambar 3 di bawah ini.



Gambar 2. Pemasangan Infus dengan Suntikkan

(Rahman, 2015)



Gambar 3. Pemasangan Infus dengan Luka Sayat

(Catatan Perawat, 2017)

Terapi *intravena* adalah kemampuan untuk mendapat akses ke sistem *vena* guna memberikan cairan dan obat merupakan keterampilan perawat. Tanggung jawab ini termasuk memilih *vena*, jenis *kanula* yang sesuai, dan mahir dalam teknik penusukan *vena*. Faktor-faktor lain yang mempengaruhi pemasangan infus termasuk jenis larutan yang akan diberikan, lamanya terapi *intravena* yang diharapkan, keadaan umum pasien, dan *vena* yang digunakan.

1. Tujuan Pemasangan Infus

Menurut Setyorini (2006), tujuan pemberian terapi intravena yaitu memberikan atau menggantikan cairan tubuh yang mengandung air, elektrolit, vitamin, protein, lemak, dan kalori yang tidak dapat dipertahankan secara adekuat melalui oral, memperbaiki keseimbangan asam-basa, memperbaiki volume komponen-komponen darah, memberikan jalan masuk untuk pemberian obat-obatan ke dalam tubuh, memantau tekanan *vena* sentral, dan memberikan nutrisi pada saat sistem pencernaan diistirahatkan.

2. Pedoman Pemilihan Daerah Inseri Pemasangan Infus

Menurut Perry & Potter (2006) banyak tempat yang bisa digunakan untuk terapi intravena, tetapi untuk kemudahan akses dan potensi bahaya berbeda di antara tempat-tempat ini. Selain itu biasanya perawat melakukan pertimbangan dalam memilih *vena* yaitu: 1) usia klien, 2) lamanya pemasangan infus, 3) tipe larutan yang akan diberikan, 4) kondisi *vena* klien, 5) aktivitas pasien (misal bergerak, tidak bergerak, perubahan tingkat kesadaran, gelisah), 6) terapi intravena sebelumnya. Tempat inseri *vena* yang umum digunakan adalah tangan dan lengan. Tetapi juga dapat dilakukan di kaki apabila klien dalam kondisi tidak memungkinkan untuk dipasang di daerah tangan.

3. Tipe-tipe Cairan Intravena

Cairan hipotonik adalah osmolaritasnya lebih rendah dibandingkan serum (konsentrasi ion Na^+ lebih rendah dibandingkan serum), sehingga larut dalam serum, dan menurunkan osmolaritas serum. Maka cairan “ditarik” dari dalam pembuluh darah keluar ke jaringan sekitarnya (prinsip cairan berpindah dari osmolaritas rendah ke osmolaritas tinggi), sampai akhirnya mengisi sel-sel yang dituju. Digunakan pada keadaan sel “mengalami” dehidrasi, misalnya pada pasien cuci darah (*dialysis*) dalam terapi diuretik, juga pada pasien hiperglikemia (kadar gula darah tinggi) dengan ketoasidosis diabetik. Komplikasi yang membahayakan adalah perpindahan tiba-tiba cairan dari dalam pembuluh darah ke sel, menyebabkan *kolaps kardiovaskular* dan peningkatan tekanan *intrakranial* (dalam otak) pada beberapa orang. Contohnya adalah NaCl 45% dan *Dekstrosa* 2,5% produk darah (darah), dan *albumin*. (Perry & Potter, 2006).



Gambar 4. Macam-Macam Jenis Infus

(www.widatra.com)

Cairan Isotonik: osmolaritas (tingkat kepekatan) cairannya mendekati serum (bagian cair dari komponen darah), sehingga terus berada di osmolaritas (tingkat kepekatan) cairannya mendekati serum (bagian cair dari komponen darah), sehingga terus berada di dalam pembuluh darah. Bermanfaat pada pasien yang mengalami *hipovolemi* (kekurangan cairan tubuh, sehingga tekanan darah terus menurun). Memiliki risiko terjadinya *overload* (kelebihan cairan), khususnya pada penyakit gagal jantung kongestif dan hipertensi. Contohnya adalah cairan Ringer-Laktat (RL), dan normalsaline/larutan garam fisiologis (NaCl 0,9%). Cairan hipertonik: osmolaritasnya lebih tinggi dibandingkan serum, sehingga “menarik” cairan dan elektrolit dari jaringan dan sel ke dalam pembuluh darah. Mampu menstabilkan tekanan darah, meningkatkan produksi urin, dan mengurangi edema (bengkak). Penggunaannya kontradiktif dengan cairan Hipotonik. Misalnya Dextrose 5%, NaCl 45% hipertonik, Dextrose 5%+Ringer-Lactate, Dextrose 5%+NaCl 0,9%,

Pembagian cairan lain adalah berdasarkan kelompoknya:

- a. Cairan Kristaloid : bersifat isotonik, maka efektif dalam mengisi sejumlah volume cairan (*volume expanders*) ke dalam pembuluh darah dalam waktu yang singkat, dan berguna pada pasien yang memerlukan cairan segera. Misalnya Ringer-Laktat dan garam fisiologis.
- b. Cairan Koloid : ukuran molekulnya (biasanya protein) cukup besar sehingga tidak akan keluar dari *membrane capiler*, dan tetap berada dalam pembuluh darah, maka sifatnya hipertonik, dan dapat menarik cairan dari luar pembuluh darah. Contohnya adalah albumin dan steroid (Perry & Potter, 2006).

4. Komposisi Cairan Terapi Intravena

Larutan NaCl (berisi air dan elektrolit (Na⁺, Cl⁻), Larutan *dextrose* (berisi air atau garam dan kalori), Ringer laktat, berisi air (Na⁺, K⁺, cl⁻, ca⁺⁺, laktat), *Balans isotonic* berisi (air, elektrolit, kalori (Na⁺, K⁺, Mg²⁺, Cl⁻, HCO, Glukonat), *whole blood* (darah lengkap) dan komponen darah, *Plasma expanders* (berisi albumin, dextran, fraksi protein plasma 5%, *hespan* yang dapat meningkatkan tekanan *osmotic*, menarik cairan dari *intertisiall*, kedalam sirkulasi dan meningkatkan volume darah sementara), *Hiperelimentasi parenteral* (berisi cairan, elektrolit, asam amino, dan kalori) (Smeltzer & Bare, 2002).

5. Tipe-tipe Pemberian Terapi Intravena

Intravena (IV) *push* (IV *bolus*), adalah memberikan obat dari jarum suntik secara langsung ke dalam saluran/jalan infus. Indikasi pada keadaan *emergency resusitasi* jantung paru, memungkinkan pemberian obat langsung kedalam intravena, untuk mendapat respon yang cepat terhadap pemberian obat (*furosemid*

dan *digoksin*), untuk memasukkan dosis obat dalam jumlah besar secara terus menerus melalui infus (*lidocain, xilocain*), untuk menurunkan ketidaknyamanan pasien dengan mengurangi kebutuhan akan injeksi, untuk mencegah masalah yang mungkin timbul apabila beberapa obat yang dicampur.

Continous Infusion (infus berlanjut) dapat diberikan secara tradisional melalui cairan yang digantung, dengan atau tanpa pengatur kecepatan aliran. Infus melalui intravena, *intra arteri*, dan *intra thecal (spinal)* dapat dilengkapi dengan menggunakan pompa khusus yang ditanam maupun eksternal. Hal yang perlu dipertimbangkan yaitu mampu untuk mengimpus cairan dalam jumlah besar dan kecil dengan akurat, adanya alarm menandakan adanya masalah seperti adanya udara di selang infus atau adanya penyumbatan, mengurangi waktu perawatan untuk memastikan kecepatan aliran infus. Kerugian: memerlukan selang yang khusus dan biaya lebih mahal. *Intermittent Infusion* (Infus Sementara) dapat diberikan melalui *heparin lock*, “*piggy bag*” untuk infus yang kontiniu, atau untuk terapi jangka panjang melalui perangkat infus (Perry & Potter, 2006).

6. *Monitoring* Infus

Monitoring infus adalah pemantauan perawat untuk mencatat hasil data dari pasien sebelum maupun setelah melakukan tindakan perawatan infus. Pemantauan berkadar tingkat tinggi dilakukan agar dapat membuat pengukuran melalui waktu yang menunjukkan pergerakan ke arah tujuan atau menjauh dari itu. *Monitoring* akan memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa terhadap proses

berikut objek atau untuk mengevaluasi kondisi atau kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis antara lain tindakan untuk mempertahankan manajemen yang sedang berjalan.

Monitoring merupakan tanggung jawab perawat dan meliputi laju arus infus sambil memastikan kebetahan dan keselamatan pasien atau klien. Laju arus infus ditetapkan menurut perintah dokter, dokter mungkin telah menentukan jumlah infus dalam 8 atau 24 jam. Laju infus dihitung berdasarkan jumlah tetes larutan per menit. Dibawah ini disertakan rumus yang dapat digunakan untuk menentukan laju arus infus :

Jumlah tetes per menit = Tetesan infus diatur sesuai pogram pengobatan, tidak boleh teralu cepat atau terlalu lambat. Ada dua metode yang digunakan untuk menghitung jumlah tetesan, yakni :

- a. Jumlah mililiter/jam.jumlah tetesan dihitung dengan mebandingkan voleme cairan yang harus diberikan (ml) dengan lamanya pemberian (jam).

Contoh : 3000 ml cairan RL. Harus diberikan dalam 24 jam. Dengan demikian

$$\text{Jumlah tetesan} = \frac{\text{Jumlah kebutuhan cairan (ml)}}{\text{waktu (jam)}}$$

$$\text{Jumlah tetesan} = \frac{3000 \text{ ml}}{24 \text{ jam}}$$

$$\text{Jumlah tetesan} = 125\text{ml/jam}$$

- b. Tetesan/menit. Jumlah tetesan dihitung dengan mengalikan jumlah cairan yang dibutuhkan (ml) dengan faktor tetes, kemudian membaginya dengan lama pemberian (menit). Faktor tetes detentukan berdasarkan alat yang digunakan.

Contoh pemberian cairan :

Seorang klien datang dengan keluhan mual dan muntah yang terus menerus. dari pengkajian itu di temukan tanda-tanda dehidrasi sedang. Berdasarkan pemeriksaan, klien harus mendapatkan terapi cairan intervena. Dokter menginstruksikan pemberian 3 kolf RL dalam 24 jam. Dengan demikian jumlah tetesan infus/menit untuk klien tersebut adalah :

$$1 \text{ cc} = 1 \text{ ml}$$

$$1 \text{ kolf} = 500 \text{ cc}$$

$$3 \text{ kolf} = 3 \times 500 \text{ cc} = 1500 \text{ cc atau } 1500 \text{ ml}$$

$$\text{Jumlah tetesan} = \frac{\text{Jumlah kebutuhan cairan (ml)} \times \text{Faktor Tetes}}{\text{waktu (menit)}}$$

$$\text{Jumlah tetesan} = \frac{1500 \text{ ml} \times 20}{1440 \text{ menit}}$$

$$\text{Jumlah tetesan} = 20,8 \text{ tetes/menit}$$

$$\text{Jumlah tetesan} = 21 \text{ tetes/menit}$$

Faktor tetes, atau jumlah tetes per milimeter, ditentukan oleh ukuran bukan pada peralatan infus. Faktor tetes yang lebih banyak di pergunakan adalah **15 tetes/ml, 20 tetes/ml, dan 60 tetes/ml.**

7. Penentuan Kecepatan Cairan Intravena (Infus)

Pertama atur kecepatan tetesan pada tabung intravena. Tabung makrodrip dapat meneteskan 10 atau 15 tetes per 1 ml. Tabung mikrodrip meneteskan 60 tetes per 1 ml. Jumlah tetesan yang diperlukan untuk 1 ml disebut faktor tetes. Mengatur jumlah mililiter cairan yang akan diberikan dengan jumlah total cairan yang akan diberikan dengan jumlah jam infus yang berlangsung. Kemudian kalikan hasil

tersebut dengan faktor tetes. Untuk menentukan berapa banyak tetesan yang akan diberikan permenit, bagi dengan 60. Hitung jumlah tetesan permenit yang akan diinfuskan. Jika kecepatan alirannya tidak tepat, sesuaikan dengan kecepatan tetesan (Smeltzer & Bare, 2002).

8. Komplikasi Terapi Intravena

Menurut Darmadi (2010), terdapat beberapa komplikasi yang dapat terjadi dalam pemasangan infus, yaitu: 1) hematoma, yakni darah mengumpul dalam jaringan tubuh akibat pecahnya pembuluh darah arteri vena, atau kapiler, terjadi akibat penekanan yang kurang tepat saat memasukkan jarum, atau “tusukan” berulang pada pembuluh darah, 2) Infiltrasi, yakni masuknya cairan infus ke dalam jaringan sekitar (bukan pembuluh darah), terjadi akibat ujung jarum infus melewati pembuluh darah, 3) Plebitis, atau bengkak (*inflamasi*) pada pembuluh vena akibat infus yang dipasang tidak dipantau secara ketat dan benar, 4) Emboli udara yaitu masuknya udara ke dalam sirkulasi darah, terjadi akibat masuknya udara yang ada dalam cairan infus ke dalam pembuluh darah, rasa perih/sakit dan reaksi alergi.

D. Komunikasi Serial

Komunikasi serial merupakan komunikasi dimana transmisi data dilakukan per bit. Proses pengiriman data ini dilakukan secara serial (data dikirim satu persatu secara berurutan), sehingga komunikasi serial lebih lambat daripada komunikasi paralel. Keunggulan dari komunikasi serial yaitu biaya yang lebih murah, dapat digunakan untuk menghubungkan dua peralatan yang sangat jauh. Misalnya menumpang pada kabel telepon. Komunikasi serial dapat bekerja dengan baik

dengan cara *data byte* harus diubah ke dalam bit-bit serial menggunakan peralatan yang disebut *shift register parallel-in serial-out*, kemudian data dikirimkan hanya dengan satu jalur data saja.

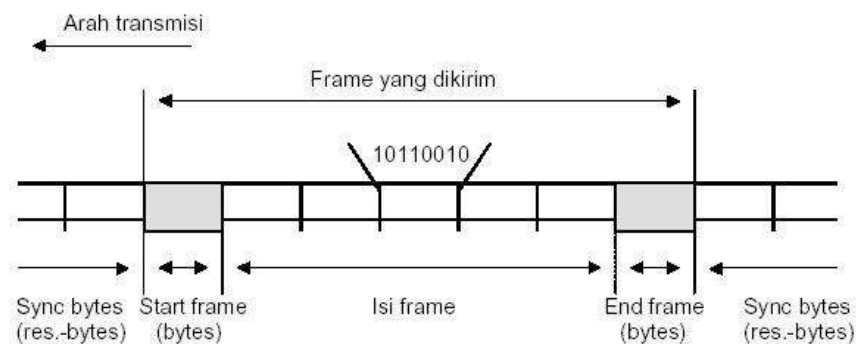
Hal yang serupa dikerjakan pada penerima, dimana penerima harus mengubah bit-bit serial yang diterimanya menjadi data byte yang persis seperti data semula pada pengirim, dengan menggunakan *shift register serial-in parallel-out*. Data serial tersebut dikirim menumpang jalur telpon, maka dibutuhkan peralatan pengubah status digital 0s atau 1s menjadi sinyal suara audio. Peralatan seperti ini kemudian disebut *modem (modulator/demodulator)*. *Modulator* sebagai pengubah sinyal digital menjadi sinyal audio. Sedangkan *demodulator* adalah sebagai mengubah kembali sinyal audio menjadi sinyal digital. Pada jarak yang sangat dekat, kita dapat menggunakan komunikasi serial sederhana dan tidak perlu modulasi. Seperti yang dapat kita lihat pada hubungan komputer kita dengan *keyboard* atau *mouse*.

1. Komunikasi Serial *Synchronous* dan *Asynchronous*

Komunikasi data serial mengenal dua buah metode, yaitu *synchronous* dan *asynchronous*. Metode *synchronous* mengirimkan datanya beberapa *byte* atau karakter (atau disebut blok data atau *frame*) sebelum meminta konfirmasi apakah data sudah diterima dengan baik atau tidak. Sementara metode *asynchronous* data dikirim satu byte setiap pengiriman. Biasanya tidak dibutuhkan konfirmasi penerimaan data. Dari kedua jenis metode tersebut dapat dipilih dan dilakukan lewat program. Tentu saja dibutuhkan program yang baik dan teliti untuk melakukannya. Proses pengiriman data serial tersebut sudah dilakukan oleh sebuah

chip tersendiri (*hardware*). Salah satu *chip* disebut UART (*Universal Asynchronous Receiver Transmitter*) dan satunya lagi disebut USART (*Universal Synchronous Asynchronous Receiver Transmitter*). Dalam protokol berbeda, *synchronous* memerlukan sinyal tambahan yang digunakan untuk mensinkronisasi setiap denyut dari proses transfer.

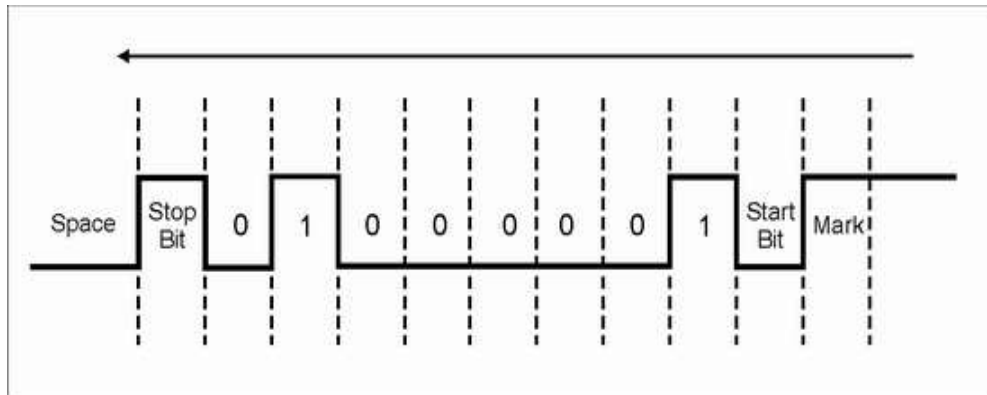
Berikut ini sinyal transmisi sinkron yang dapat dilihat pada Gambar 5 sebagai berikut:



Gambar 5. Sinyal Transmisi Asinkron
(Suyadi, 2012)

Komunikasi data serial *Asynchronous* sekarang sudah digunakan demikian luas untuk transmisi yang berorientasi karakter, sementara metode *Synchronous* digunakan untuk transmisi yang berorientasi blok. Pada mode *Asynchronous*, setiap karakter ditempatkan berada diantara *bit start* dan *bit stop*. *Bit start* selalu satu bit, tapi *stop bit* bisa satu bit atau dua bit. *Start bit* selalu 0 (*low*) dan *stop bit* selalu 1 (*high*). Contohnya, pada gambar 5 dimana karakter A (01000001 biner) dibingkai (dikurung) oleh start bit dan satu *stop bit*.

Berikut ini contoh pembingkai karakter ASCII “A” yang dapat dilihat pada Gambar 6 sebagai berikut:



Gambar 6. Pembingkai Karakter ASCII “A” (41h)

(Suyadi, 2012)

Pada komunikasi serial *Asynchronous*, peralatan dan modem dapat diprogram untuk menggunakan lebar data 7 atau 8-bit. Tentu saja ditambah dengan *Stop bit*. Pada zaman dahulu, sistem karakter ASCII masih terbatas pada data 7-bit, namun sekarang ASCII *extended* sudah umum menggunakan lebar data 8-bit. Pada peralatan kuno, dengan komponen-komponen yang lambat pula, dibutuhkan *stop bit* yang sedikit panjang, hal ini dimaksudkan untuk memberi kesempatan peralatan untuk menangani data yang telah diterimanya, dan untuk mempersiapkan diri untuk menerima *byte* berikutnya.

Namun saat ini modem PC biasanya hanya menggunakan satu *bit stop*. Jika kita hitung, dengan menggunakan satu bit *stop* total kita memiliki 10-bit untuk setiap karakter 8-bit. Artinya setiap karakter 8-bit dikirim bersama-sama *start* dan *stop* bit sehingga total menjadi 10-bit, yang artinya ada proses transfer lebih panjang 20% setiap karakternya. Pada sistem yang sangat mementingkan integritas data yang disimpan, maka ditambahkan lah bit paritas ke bingkai data tersebut.

Maksudnya untuk setiap karakter 8-bit kita masih menambahkan bit paritas di samping bit *start* dan bit *stop*. Sehingga total adalah 11-bit. Adapun bit paritas adalah bit yang menunjukkan bahwa data yang dimaksud adalah memiliki jumlah bit 1s (*high*) ganjil atau genap. Bit paritas adalah bit di luar data yang bersangkutan atau merupakan tambahan. *Chip* UART khusus biasanya sudah dilengkapi dengan keperluan paritas tersebut secara *hardware*. Bahkan ada beberapa pilihan untuk penanganan paritas ini, misalnya *odd-*, *even-* dan *no-parity*.

2. *Data Transfer Rate*

Kecepatan transfer data pada komunikasi data serial diukur dalam satuan BPS (*bits persecond*). Sebutan terkenal lainnya adalah *baud rate*. Namun *Baud* dan *bps* tidaklah sama. Hal ini mengacu kepada fakta bahwa *baud rate* adalah terminologi modem dan diartikan sebagai perubahan sinyal dalam satuan bit sinyal setiap detik. Sedang *data tranfer rate* penamaannya mengacu pada jumlah *bit* dari *byte* data yang ditransfer setiap detik. Sementara itu untuk kecepatan transfer data (*data transfer rate*) pada komputer tergantung pada jenis komunikasi yang diberlakukan atasnya.

E. **Komunikasi IIC (*Inter Integrated Circuit*)**

IIC singkatan dari *Inter Integrated Circuit* merupakan sebuah protokol untuk komunikasi serial antar IC, dan sering disebut juga *Two Wire Interface* (TWI). Bus yang digunakan untuk komunikasi antara mikrokontroler dan *device* periferifal seperti memori, sensor temperatur dan *I/O expander*. Komunikasi IIC bersifat *synchronous* dengan menggunakan protokol dan hanya menggunakan dua buah kabel untuk berkomunikasi, yaitu *synchronous clock* (SCL) dan *synchronous data*

(SDA). Secara berurutan alur komunikasi IIC yaitu apabila terdapat data yang dikirimkan dari *master* ke *slave* kemudian data tersebut dikirimkan kembali dari *slave* ke *master*. Berikut ini aliran data IIC yang dapat dilihat pada Gambar 7 sebagai berikut:



Gambar 7. Aliran Data IIC

(Darma, 2016)

Perangkat IIC menggunakan dua buah pin *open-drain* dua arah dengan memberikan *pull-up* resistor untuk setiap garis bus sehingga berlaku seperti AND menggunakan kabel. AVR dapat menggunakan 120 jenis perangkat untuk berbagi pada *bus* IIC yang masing-masing disebut sebagai *node*. Setiap *node* beroperasi sebagai *master* atau *slave*. *Master* merupakan perangkat yang menghasilkan *clock* untuk sistem, menginisiasi, dan juga memutuskan sebuah transmisi. *Slave* merupakan *node* yang menerima *clock* dan dialamatkan oleh *master*. Baik *master* dan *slave* dapat menerima dan mentransmisikan data. Adapun bentuk fisik dari modul IIC dapat dilihat pada Gambar 8 sebagai berikut:



Gambar 8. Bentuk Fisik Modul I2C

(<http://www.14core.com>)

IIC merupakan *protocol* komunikasi serial dimana setiap bit data ditransfer pada jalur SDA yang disinkronisasikan dengan pulsa *clock* pada jalur SCL. Jalur data tidak dapat berubah ketika jalur *clock* berada dalam kondisi *high*. Dalam IIC, setiap alamat atau data yang ditransmisikan harus dibentuk dalam sebuah paket dengan panjang 9 bit dimana 8 bit pertama disimpan dalam jalur SDA oleh *transmitter*, dan bit ke-9 merupakan *acknowledge* (atau *not acknowledge*) oleh *receiver*. IIC juga diistilahkan sebagai *Two-wire Serial Interface* (TWI), bergantung dari istilah yang digunakan oleh pabrik yang membuat perangkat.

Salah satu perangkat yang digunakan dengan komunikasi IIC adalah *Liquid Crystal Display* (LCD). Perangkat ini menyediakan *display* atau tampilan yang tersusun atas beberapa LED sehingga dapat membuat karakter sesuai dengan program.

F. Mikrokontroler

Mikrokontroler adalah sebuah *chip* yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program di dalamnya. Pada dasarnya mikrokontroler terdiri dari CPU, memori, I/O tertentu dan unit pendukung seperti *Analog-to-Digital Converter* (ADC) yang sudah terintegasi di dalamnya. Pada *chip* mikrokontroler terdapat IC yang dapat diprogram berulang kali, baik ditulis ataupun dihapus. Sehingga dalam penggunaannya dapat menghemat biaya. Selain itu, kelebihan lainnya mikrokontroler yaitu tersedianya RAM dan peralatan I/O pendukung sehingga ukuran *board* mikrokontroler menjadi sangat ringkas (Widodo, 2004).

Saat ini mikrokontroler banyak digunakan untuk kepentingan sistem kendali. Meskipun memiliki bentuk yang jauh lebih kecil dari suatu komputer pribadi, mikrokontroler dibangun dari elemen-elemen dasar yang sama. Secara sederhana, komputer akan menghasilkan keluaran spesifik berdasarkan masukan yang diterima dan program yang dikerjakan. Seperti umumnya komputer, mikrokontroler adalah alat yang mengerjakan instruksi-instruksi yang diberikan kepadanya. Artinya, bagian terpenting dan utama dari suatu sistem terkomputerisasi adalah program itu sendiri yang dibuat oleh seorang *programmer*.

Menurut (Dinawan, 2012), terdapat beberapa fitur yang umum ada di dalam mikrokontroler adalah sebagai berikut:

1. RAM (*Random Access Memory*)

RAM digunakan oleh mikrokontroler untuk penyimpanan variable. Memori ini bersifat *volatile* yang berarti akan kehilangan semua datanya jika tidak mendapatkan catu daya.

2. ROM (*Read Only Memory*)

ROM sering disebut sebagai kode memori karena berfungsi untuk penyimpanan program yang akan diberikan oleh *user*.

3. *Register*

Register merupakan tempat penyimpanan nilai-nilai yang akan digunakan dalam proses yang telah disediakan oleh mikrokontroler.

4. *Special Function Register*

Special Function Register merupakan *register* khusus yang berfungsi untuk mengatur jalannya mikrokontroler. *Register* ini terletak pada RAM.

5. *Input dan Output Pin*

Pin *input* adalah bagian yang berfungsi sebagai penerima sinyal dari luar. Pin ini dapat dihubungkan ke berbagai media masukan seperti *keypad*, sensor, dan sebagainya. Sedangkan pin *output* adalah bagian yang berfungsi untuk mengeluarkan sinyal dari hasil proses algoritma mikrokontroler.

6. *Interrupt*

Interrupt adalah bagian mikrokontroler yang berfungsi sebagai bagian yang dapat melakukan interupsi, sehingga ketika program utama sedang berjalan, program utama tersebut dapat diinterupsi dan menjalankan program interupsi terlebih dahulu. Adapun beberapa *interrupt* pada umumnya adalah sebagai berikut:

a. *Interrupt External*

Interrupt akan terjadi apabila ada masukan dari pin *interrupt*.

b. *Interrupt Timer*

Interrupt ini akan terjadi apabila waktu tertentu telah tercapai.

c. *Interrupt Serial*

Interrupt ini akan terjadi ketika ada penerimaan data dari komunikasi serial.

G. Arduino

Arduino adalah sebuah *platform* dari *physical computing* yang bersifat *open source* (Aslamia, 2015). Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi merupakan kombinasi dari *hardware*, bahasa pemrograman dan *Integrated Development Environment (IDE)* yang canggih. IDE adalah sebuah *software* yang sangat berperan yang digunakan untuk menulis program, meng-*compile* menjadi

kode biner dan mengunggahnya ke dalam memori mikrokontroler. Menurut (Syahwil, 2013), Arduino adalah papan elektronik *open source* yang di dalamnya terdapat komponen utama, yaitu sebuah *chip* mikrokontroler AVR dari perusahaan Atmel. Berikut ini karakteristik dan struktur arduino sebagai berikut:

1. Arduino IDE merupakan multi *platform* yang dapat dijalankan pada berbagai sistem operasi seperti *Windows* ataupun *Linux*. IDE adalah program komputer yang berfungsi untuk menyediakan semua fasilitas yang diperlukan dalam pembangunan perangkat lunak baik *editor*, *compiler*, *linker*, maupun *debugger*.
2. Pemrograman pada *hardware* Arduino menggunakan kabel yang terhubung dengan *port Universal Serial Bus (USB)*. Hal ini disebabkan karena banyak komputer sekarang yang tidak dilengkapi dengan *port serial*.
3. Arduino adalah *hardware* dan *software* yang bersifat *open source* yaitu sistem pengembangan yang tidak dikoordinasi oleh individu atau lembaga pusat, namun oleh para pelaku yang bekerja sama dengan memanfaatkan kode sumber (*source code*).
4. Biaya pembelian *hardware* tergolong cukup murah sehingga tidak akan menghabiskan banyak uang untuk memilikinya.

1. Arduino Mega 2560

Arduino Mega 2560 adalah papan mikrokontroler Atmega 2560 yang memiliki 54 pin I/O, yang terdiri dari 15 pin tersebut digunakan sebagai *output PWM* atau *pulse Width Modulation*), 16 pin analog *input*, 4 UART (*Universal Asynchronous Receiver/Transmitter*), osilator kristal 16 MHz, koneksi USB, *jack listrik*, *header*

ICSP (*In-Circuit Serial Programming*), dan tombol *reset*. Berikut ini bentuk fisik dari Arduino Mega 2560 dapat dilihat pada Gambar 9 sebagai berikut:

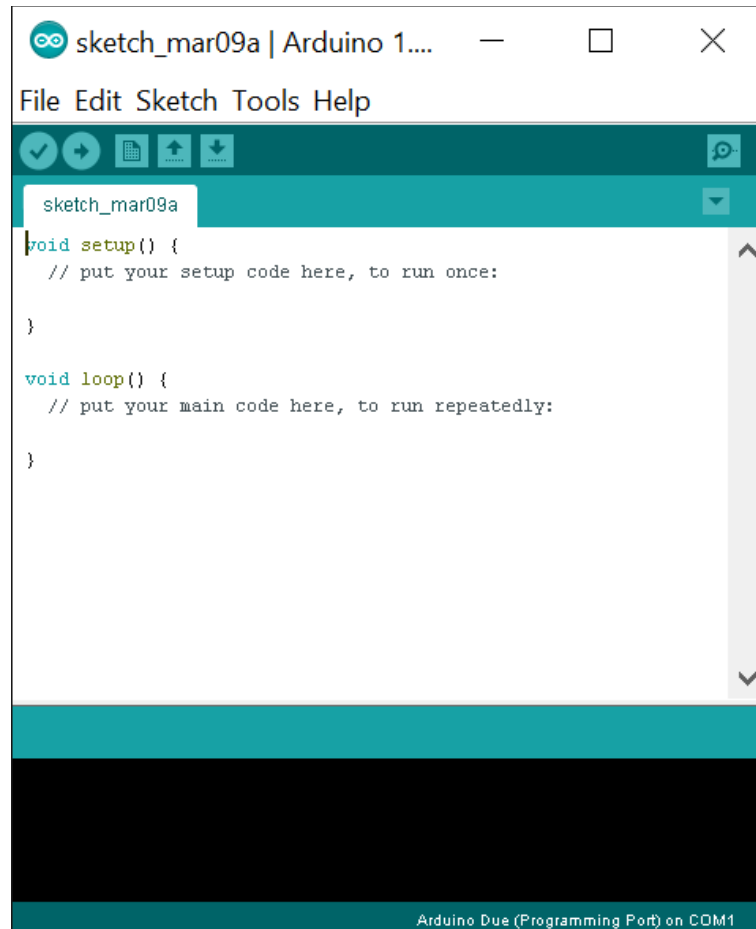


Gambar 9. Arduino Mega 2560

(www.arduino.cc)

2. Arduino IDE

Arduino IDE adalah *software* yang dirancang menggunakan bahasa pemrograman Java yang terdiri dari 3 bagian yaitu *Editor* program, *Compiler*, dan *Uploader*. Pada bagian *editor*, *user* dapat melakukan pemrograman baik itu menulis ataupun mengedit program dalam bahasa *processing*. Sementara pada bagian *Compiler* berisi pengubah kode program menjadi kode biner agar kode tersebut dapat terbaca oleh mikrokontroler. Sedangkan bagian *Uploader* digunakan untuk menuliskan dan menyalin kode biner dari komputer ke dalam memori *board* Arduino. Bahasa pemrograman yang digunakan pada Arduino IDE yaitu jenis bahasa pemrograman C yang mudah digunakan.



Gambar 10. Tampilan Arduino IDE

(<https://blog.arduino.cc/>)

Pada tampilan Arduino IDE terdapat *toolbar* yang didesain untuk mempermudah dalam melakukan pemrograman. Berikut ini fungsi-fungsi pada *toolbar IDE* sebagai berikut:

- a. *Verify*, digunakan untuk melakukan kompilasi program yang saat di *editor*.
- b. *New*, digunakan untuk membuat program baru dengan mengosongkan isi dari jendela *editor* saat ini.
- c. *Open*, digunakan untuk membuka program yang ada dari sistem file.
- d. *Save*, digunakan untuk menyimpan program saat ini.

- e. *Upload*, digunakan untuk menyalin data hasil pemrograman dari komputer ke dalam *memory board* arduino. Ketika melakukan *upload*, maka harus melakukan pengaturan jenis arduino dan *port* COM yang digunakan.
- f. *Serial Monitor*, digunakan untuk melihat hasil pemrograman yang telah tersimpan dalam *memory* arduino.

Kode program yang dituliskan pada *chip* mikrokontroler Arduino umumnya menggunakan beberapa fungsi seperti tipe data, operator, dan program kontrol. Pada setiap bagian dari data yang disimpan dalam program Arduino umumnya memiliki tipe datanya masing-masing. Berikut ini beberapa tipe data yang terdapat pada Arduino yang ditunjukkan pada Tabel 1.

Tabel 1. Tipe Data Bahasa C

JENIS	RANGE
<i>int / sign int</i>	-32768 - +32767 ($2^{15} - 1$)
<i>unsign int</i>	0 – 65535
<i>short int / signed short int</i>	-28 – 127
<i>unsigned short int</i>	0 – 255
<i>long int / signed long int</i>	-2147483648 – 2147483648
<i>unsigned long int</i>	0 – 4294967296
<i>char</i>	karakter ASCII
<i>unsigned char</i>	0 – 255
<i>signed char</i>	-128 - 127
<i>float</i>	maksimum nilai 6 digit
<i>double</i>	maksimum nilai 12 digit
<i>long double</i>	maksimum nilai 24 digit

(Widodo, 2016)

Sementara operator merupakan simbol atau karakter yang biasa dilibatkan dalam program untuk melakukan sesuatu operasi atau manipulasi, seperti menjumlahkan dua buah nilai, memberikan nilai ke suatu variabel, membandingkan kesamaan dua buah nilai. Sebagian operator C tergolong sebagai operator binari, yaitu operator yang dikenakan terhadap dua buah nilai (*operand*). Berikut ini

beberapa operasi kondisi yang sering digunakan dalam bahasa C yang dapat dilihat pada Tabel 2 sebagai berikut:

Tabel 2. Operasi Kondisi

Operator	Keterangan
<	Lebih kecil
<=	Lebih kecil atau sama dengan
>	Lebih besar
>=	Lebih besar atau sama dengan
==	Sama dengan
!=	Tidak sama dengan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Sisa Bagi (modulus)
!	<i>Boolean NOT</i>
&&	<i>Boolean AND</i>
	<i>Boolean OR</i>
~	<i>Komplemen Bitwise</i>
&	<i>Bitwise AND</i>
	<i>Bitwise OR</i>
^	<i>Bitwise Exclusive OR</i>
>>	<i>Right Shift</i>
<<	<i>Left Shift</i>
=	Untuk memasukan nilai
+=	Untuk menambah nilai dari keadaan semula
-=	Untuk mengurangi nilai dari keadaan semula
*=	Untuk mengalikan nilai dari keadaan semula
/=	Untuk melakukan pembagian terhadap bilangan semula
%=	Untuk memasukan nilai sisa bagi dari pembagian bilangan semula
<<=	Untuk memasukan <i>shift left</i>
>>=	Untuk memasukan <i>shift right</i>
&=	Untuk memasukan <i>bitwise AND</i>
^=	Untuk memasukan <i>bitwise XOR</i>
\=	Untuk memasukan <i>bitwise OR</i>

(Widodo, 2016)

Selain itu, dalam melakukan pemrograman menggunakan Arduino IDE juga dibutuhkan program kontrol agar program yang akan dibuat sesuai dengan tujuan. Berikut ini program kontrol dalam bahasa C sebagai berikut:

1. Percabangan *if* dan *if ... else ...*

Perintah *if* dan *if ... else ...* digunakan untuk melakukan operasi percabangan bersyarat.

2. Percabangan *switch*

Pernyataan *switch* adalah sebuah variabel secara berurutan diuji oleh beberapa konstanta bilangan bulat atau konstanta karakter sintaks perintah *switch*.

3. *Looping*

Looping adalah pengulangan satu atau beberapa perintah sampai mencapai keadaan tertentu. Ada tiga perintah *looping*, yaitu:

a). *for ...*

b) *while ...*

c) *do...while....*

H. Modul Wi-Fi

WiFi atau *Wireless Fidelity* merupakan seperangkat standar yang digunakan untuk komunikasi jaringan lokal tanpa kabel (WLAN) yang didasari pada spesifikasi IEEE 802.11 (Yuhefizar, 2008). WiFi menggunakan sinyal radio yang bekerja pada frekuensi tertentu sehingga dengan adanya WiFi ini semua perangkat yang terhubung bisa terkoneksi dengan internet. *Hotspot* merupakan sarana terkoneksi jaringan internet tanpa kabel dengan menggunakan standar wireless

LAN, namun demikian dalam menjalankan *hotspot* diperlukan sarana lain, seperti Notebook/Laptop/PDA yang memiliki fasilitas *wireless* LAN (Ida, 2010).

Mikrokontroler Wemos adalah sebuah mikrokontroler pengembangan berbasis modul mikrokontroler ESP8266. Mikrokontroler Wemos dibuat sebagai solusi dari mahalnya sebuah sistem *wireless* berbasis mikrokontroler lainnya. Mikrokontroler wemos ini memiliki kemampuan untuk menyediakan fasilitas konektivitas WiFi dengan mudah serta *memory* yang digunakan sangat besar yaitu 4 MB. Berikut ini bentuk fisik dari Wemos yang ditunjukkan pada Gambar 11 sebagai berikut.



Gambar 11. Bentuk Fisik Wemos D1 Mini

(<https://www.projetsdiy.fr/>)

Pada mikrokontroler wemos memiliki dua buah *chipset* yang digunakan sebagai otak kerja *platform* tersebut. Beberapa *chipset* pada mikrokontroler ini sebagai berikut:

1. *Chipset* ESP8266

ESP8266 adalah sebuah *chip* mikrokontroler yang memiliki fitur WiFi yang mendukung *stack* TCP/IP. Modul ini dapat memungkinkan mikrokontroler untuk terhubung dengan jaringan WiFi dan membuat koneksi TCP/IP hanya dengan

menggunakan *command* yang sederhana seperti gaya Hayes. Dengan *clock* 80 MHz *chip* ini dibekali dengan 4 MB eksternal RAM, mendukung format IEEE 802.11 b/g/n sehingga tidak menyebabkan *interference* bagi yang lain. Mendukung enkripsi WEP dan WPA sehingga menjadikan *chipset* ini sangat aman digunakan. *Chipset* ini memiliki 16 GPIO pin yang bekerja pada 3.3 volt, 1 pin ADC dengan resolusi 10 bit.

2. *Chipset* CH340

CH340 adalah sebuah *chipset* yang mengubah USB menjadi *serial interface*. Pada 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.

I. *ThingSpeak*

Thingspeak adalah *open source* “*internet of things*” atau biasa disingkat IoT aplikasi dan API untuk menyimpan dan mengambil data menggunakan HTTP melalui internet. Berikut ini berapa fitur dari *thingspeak* diantaranya, yaitu *Open API*, mengoleksi data *real-time*, menampilkan geolokasi data, dapat memproses data melalui Matlab, Visualisasi data (Asyistawa, 2017). Selain itu, *thingspeak* juga memungkinkan dapat digunakan untuk pembuatan aplikasi sensor *logging*, aplikasi lokasi pelacakan, dan jaringan sosial hal dengan update status. *Internet of Things* (IOT) menyediakan akses ke berbagai perangkat *embedded* dan layanan *web*. *Thingspeak* adalah platform IOT yang memungkinkan kita untuk mengumpulkan,

menyimpan, menganalisis, memvisualisasikan, dan bertindak atas data dari sensor atau aktuator, seperti Arduino, Raspberry Pi, BeagleBone Hitam, dan perangkat keras lainnya.

Cara Kerja dari *thingspeak* yaitu dengan mengambil data dari pengirim *thingspeak* menggunakan *API keys* yang telah didapatkan dari website *thingspeak*. *Api keys* yaitu kode-kode khusus yang digunakan untuk membedakan darimana sumber data dan kemana data akan dikirimkan. Setiap *channel* dalam *thingspeak* memiliki kode *API* yang berbeda sehingga tidak terjadi kesalahan dalam menyalurkan data. Kode *API* selanjutnya akan dimasukkan pada program pengirim data menggunakan *http library*. Kemudian data-data tersebut akan disalurkan oleh *thingspeak* ke *channel* yang dituju. Terdapat 2 *API keys* pada website *thingspeak*, yaitu *write* dan *read*. *API write* digunakan untuk memasukan data dari webserver ke dalam *channel*. Sedangkan *API read* digunakan untuk mengambil data dari *thingspeak* dan mengirimnya pada *webserver*.

J. Sensor Tetesan Infus

Pada proyek akhir ini untuk mendeteksi tetesan infus menggunakan kombinasi sensor yaitu photodioda dengan laser dioda. Laser dioda digunakan karena cahayanya yang tampak merah, sehingga memberikan kemudahan untuk pegecekan sensor setiap saat. Komponen yang dapat menerima laser dioda ini adalah photodioda atau phototransistor. Komponen ini akan merubah energi cahaya, dalam hal ini energi cahaya infra merah, menjadi pulsa-pulsa sinyal listrik. Dibawah ini adalah penjelasan secara detail komponen sensor tetesan yang akan digunakan.

1. Photodioda

Photodioda merupakan piranti semikonduktor dengan sambungan p-n yang dirancang untuk beroperasi bila dibiaskan dalam keadaan terbalik, untuk mendeteksi cahaya (Pandiangan, Johannes. 2007).

Ketika energi cahaya dengan panjang gelombang jatuh pada sambungan photodioda, arus mengalir dalam sirkuit eksternal. Komponen ini kemudian akan bekerja sebagai generator arus, yang arusnya sebanding dengan intensitas cahaya itu. Cahaya diserap di daerah penyambungan atau daerah intrinsik menimbulkan pasangan elektron-hole yang mengalami perubahan karakteristik listrik ketika energi cahaya melepaskan pembawa muatan dalam bahan itu, sehingga menyebabkan photodioda dapat menghasilkan tegangan atau arus listrik jika terkena cahaya.

Photodioda digunakan dalam aplikasi – aplikasi yang meliputi kartu bacaan, kontrol cahaya *ambient* dan layar proyektor. Pada photodioda kita mengenal istilah responsivitas yaitu kemampuan dari sebuah photodioda untuk penambahan pada cahaya. Responsivitas dari photodioda merupakan perbandingan dalam mA/mW pada panjang gelombang tertentu photodioda honeywell SE3452 mempunyai perbandingan 0,5mA/mW. Jika cahaya yang teradiasi pada *cell* 2mV, dioda akan menghasilkan arus yang mengalir sebesar 1mA ($0,5\text{mA/mW} \times 2\text{mV}$). Respon tertinggi dari SE3452 sekitar 820nm.



Gambar 12. Sensor Cahaya Photodioda

(Sumber : Tito Apriano. 2012)

c. Mode operasi photodioda

Photodioda dapat dioperasikan dengan 2 mode yang berbeda :

1. Mode potovoltaik : seperti solar sel, penyerapan pada photodioda menghasilkan tegangan yang dapat diukur. Bagaimanapun, tegangan yang dihasilkan dari energi cahaya ini sedikit tidak linier, dan range perubahannya sedikit kecil.
2. Mode potokonduktivitas : disini photodioda diaplikasikan sebagai tegangan *reserve* (tegang balik) dari sebuah dioda yaitu (tegangan pada arah tersebut pada dioda tidak akan dihantarkan tanpa terkena cahaya) dan pengukuran menghasilkan arus poto (hal ini juga bgus untuk mengaplikasikan tegangan mendekati nol). Ketergantungan arus poto pada kekuatan caha dapat linier.

d. Karakteristik bahan photodioda

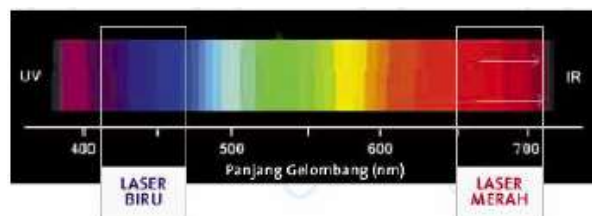
1. Silikon (Si) : arus lemah saat gelap, kecepatan tinggi, sensitivitas yang bagus antara 400 nm sampai 1000nm (terbaik antara 800 nm sampai 900 nm).
2. Germanium (Ge) : arus tinggi saat gelap, kecepatan lampat, sensitivitas antara 600 nm sampai 1800 nm (terbaik 1400 nm sampai 1500 nm).

3. Indium Gallium Arsenida (InGaAs) : mahal, arus kecil saat gelap, kecepatan tinggi, sensitivitas baik pada jarak 800 nm sampai 1700 nm (terbaik antara 1300 nm sampai 1600 nm).

2. Laser dioda

Laser merupakan singkatan dari *Light Amplification by Stimulated Emission of Radiation*, yang artinya penguatan cahaya dengan rangsangan pancaran radiasi. Sifat yang terjadi akibat kesamaan frekuensi adalah monokromatis dan sifat yang terjadi akibat kesamaan fase adalah koherensi. Jadi, syarat terbentuknya laser adalah sumber cahaya yang monokromatis dan koherensi.

Laser mempunyai sifat-sifat yang tidak dimiliki oleh sumber cahaya lain. Sifat – sifat khas laser antara lain kesearahan, intensitas, monokromatis, dan koherensi. Warna laser berkaitan dengan panjang gelombang dari laser tersebut, dimana panjang gelombang ini menentukan karakteristik dan aplikasi dari laser. Panjang gelombang laser biru lebih pendek dibandingkan laser merah atau infra-merah yang kini banyak digunakan pada perangkat-perangkat elektronik. Bila infra-merah berada dalam kisaran 780-850 nm dan laser merah pada 650-670 nm, laser biru memancarkan cahaya dengan panjang gelombang 410-460 nm, lihat Gambar 13.



Gambar 13. Spektrum Panjang Gelombang Sinar Laser
(Gita Lestari, 2013)

Klasifikasi sinar laser secara internasional, yang diuraikan dalam standar konsensus seperti EC 825, tergantung pada panjang gelombang dan apakah digunakan kontinu atau tidak (Tempo.co. 2012). Adapun klasifikasinya adalah sebagai berikut :

a. Kelas I

Aman, tidak ada kemungkinan kerusakan mata. Karena daya output laser rendah biasanya digunakan dalam CD player atau printer.

b. Kelas II

Jika terpapar laser ini mata akan *refleks* berkedip. Output daya bisa mencapai dengan 1 mW. Beberapa laser pointer yang biasa digunakan dalam presentasi masuk kategori ini.

c. Kelas II a

Laser berkekuatan rendah. Perlu lebih dari 1000 detik melihat terus-menerus untuk bisa membakar ke retina. Laser *scanner* masuk dalam kategori ini.

d. Kelas III a

Laser ini kebanyakan berbahaya. Output daya tidak mencapai 5 mW. Banyak digunakan untuk senjata api.

e. Kelas III b

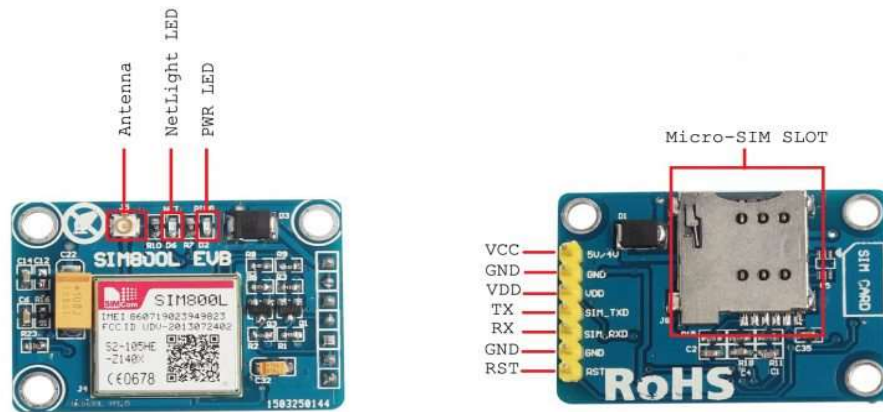
Laser di kelas ini dapat menyebabkan kerusakan mata permanen jika terpapar langsung. Outputnya mencapai 500mW. Laser ini juga bisa membakar kulit taraf ringan. Laser pointer dengan output 300mW warna hijau dalam kategori ini.

f. Kelas IV

Laser di kelas ini memiliki kekuasaan keluaran lebih dari 500 mW dan dapat menyebabkan kerusakan permanen pada mata atau kulit. Bias adigunakan untuk hiburan, industri, militer, dan laser medis.

K. Modul GSM/GPRS SIM800L

Modul GSM adalah peralatan yang didesain supaya dapat digunakan untuk aplikasi komunikasi dari mesin ke mesin atau dari manusia ke mesin. Modul GSM merupakan peralatan yang digunakan sebagai mesin dalam suatu aplikasi. Dalam aplikasi yang dibuat harus terdapat mikrokontroler yang akan mengirimkan perintah kepada modul GSM berupa *AT-Command* melalui RS232 sebagai komponen penghubung (*communication links*). Berikut ini bentuk fisik dari modul GSM SIM800L yang ditunjukkan pada Gambar 14 sebagai berikut:

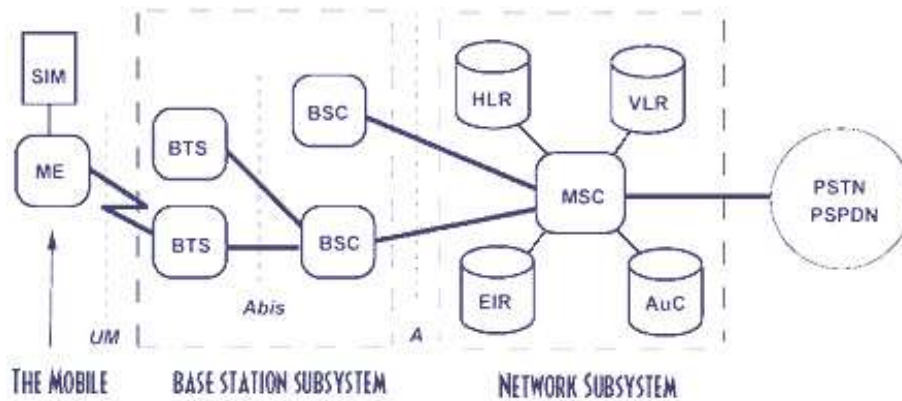


Gambar 14. Bentuk Fisik Modul GSM SIM800L

(<http://www.faranux.com/>)

Modul GSM merupakan bagian dari pusat kendali yang berfungsi sebagai *transceiver*. Modul GSM mempunyai fungsi yang sama dengan sebuah telepon seluler yaitu mampu melakukan fungsi pengiriman dan penerimaan SMS, selain itu

modul GSM juga dapat melakukan panggilan. Adanya modul GSM maka aplikasi yang dirancang dapat dikendalikan dari jarak jauh dengan menggunakan jaringan GSM sebagai media akses. Adapun diagram blok dari rangkaian modul GSM dapat dilihat pada Gambar 15.



SIM – Subscriber Identification Module.
MS – Mobile Station.
ME – Mobile Equipment.
BTS – Bare Transceiver System.
MSC – Mobile Services switching Center.
NSS – Network & Switching Subsystem.
PSTN – Public Switched Telephone Network.

Gambar 15. Skematik dari Sistem GSM

(<http://www.circuitstoday.com>)

Jaringan GSM terdiri dari beberapa *subsystem* yang memiliki fungsi dan interface-nya masing-masing. Jaringan GSM dibagi menjadi tiga bagian besar *subsystem*, yaitu:

1. *Mobile Station Subsystem* (MSS)

Mobile station (MS) terdiri dari *mobile equipment* (telepon seluler) dan kartu pintar yang disebut *subscriber identity module card* (SIM). *Mobile equipment* secara unik diidentifikasi oleh *international mobile equipment identity* (IMEI).

2. *Base Station Subsystem (BSS)*

Base station terdiri atas 2 bagian yaitu *Base Transceiver Station (BTS)* dan *Base Station Controller (BSC)*. *BTS* memiliki *transceiver* radio yang mendefinisikan sel dan menangani protokol hubungan radio dengan *MS*. *BTS* dan *BSC* berkomunikasi melalui *interface* udara atau hubungan radio. *BSC* mengatur *radio resources* untuk satu atau lebih *BTS* dan menangani *setup* saluran radio, *frequency hopping* dan proses *handover*.

3. *Network Switching Subsystem (NSS)*

Komponen utama dari *network switching subsystem (NSS)* adalah *mobile switching center (MSC)*. *MSC* melakukan *switching* hubungan antar sesama pemakai telepon seluler, dan antara pemakai telepon seluler dengan pemakaian telepon tetap (*PTSN* atau *ISDN*).

1. Layanan SMS pada Sistem GSM

SMS dikembangkan terutama sebagai alat pengirim informasi data konfigurasi dari *handset* GSM sebagai bagian dari protokol jaringan dan tidak lebih sekedar layanan tambahan daripada layanan utama sistem GSM yaitu layanan *voice* dan *switched data*. Namun pada akhirnya SMS menjadi sukses sebagai layanan *mesaging* paling populer di dunia. Berdasarkan mekanisme distribusi pesan SMS oleh aplikasi SMS, terdapat empat macam mekanisme pengiriman pesan, yaitu:

- a. *Pull*, yaitu pesan yang dikirimkan ke pengguna berdasarkan permintaan pengguna.
- b. *Push – event based*, yaitu pesan diaktivasi oleh aplikasi berdasarkan kejadian yang berlangsung.

- c. *Push – scheduled*, yaitu pesan yang diaktivasi oleh aplikasi berdasarkan waktu yang telah terjadwal.
- d. *Push – personal profile*, yaitu pesan yang diaktivasi oleh aplikasi berdasarkan profil dan preferensi dari pengguna.

SMS adalah data tipe *asynchronous message* yang pengiriman datanya dilakukan dengan mekanisme protokol *store and forward*. Hal ini berarti bahwa pengirim dan penerimaan SMS tidak perlu berada dalam status berhubungan (*connected/online*) satu sama lain ketika akan saling bertukar pesan SMS. Pengiriman pesan SMS secara *store and forward* berarti pengiriman pesan SMS menuliskan pesan dan nomor telepon tujuan dan kemudian mengirimkannya (*store*) ke *server* SMS (*SMS-Center*) yang kemudian bertanggung jawab untuk mengirimkan pesan tersebut (*forward*) ke nomor tujuan.

Keuntungan dari mekanisme *store and forward* pada SMS adalah penerima tidak perlu dalam status *online* ketika ada pengirim yang bermaksud mengirimkan pesan kepadanya. Kini SMS tidak terbatas untuk komunikasi antar manusia pengguna saja, namun juga bisa dibuat otomatis dikirim/diterima oleh peralatan untuk mencapai suatu tujuan tertentu. Namun untuk melakukannya, kita harus memahami dulu cara kerja SMS itu sendiri. Ketika pengguna mengirim SMS, maka pesan dikirim ke MSC melalui jaringan seluler yang tersedia yang meliputi *tower* BTS yang sedang meng-handle komunikasi pengguna, lalu ke BSC, kemudian sampai ke MSC. MSC kemudian mem-*forward* lagi SMS ke SMSC untuk disimpan.

SMSC kemudian mengecek (lewat HLR – *Home Location Register*) untuk mengetahui apakah *handphone* tujuan sedang aktif dan dimanakah *handphone* tujuan tersebut. Jika *handphone* sedang tidak aktif dan dimanakah *handphone* tujuan tersebut. Jika *handphone* sedang tidak aktif maka pesan tetap disimpan di SMSC itu sendiri, menunggu MSC memberitahukan bahwa *handphone* sudah aktif kembali untuk kemudian SMS dikirim dengan batas maksimum waktu tunggu yaitu *validity period* dari pesan SMS itu sendiri. Jika *handphone* tujuan aktif maka pesan disampaikan MSC lewat jaringan yang sedang mengendalikan penerima (BSC dan BTS).

Sebenarnya di dalam kebanyakan *handphone* dan GSM/CDMA modem terdapat duatu komponen *wireless* modem/*engine* yang dapat diperintah antara lain untuk mengirim suatu pesan SMS dengan protokol tertentu. Standar perintah tersebut dikenal sebagai *AT-Command*, sedangkan protokolnya disebut sebagai PDU (Protokol Data Unit). Melalui *AT-Command* dan PDU inilah kita dapat membuat komputer/mikrokontroler mengirim/menerima SMS secara otomatis berdasarakan program yang kita buat.

2. *AT-Command*

AT-Command merupakan perintah standar yang diterima oleh modem. Perintah AT (Hayes *AT-Command*) digunakan untuk berkomunikasi dengan terminal (modem) melalui gerbang serial pada komputer. *AT-Command* ini dipakai untuk memerintahkan telepon seluler mengirim dan menerima pesan SMS. Selain itu, *AT-Command* juga dapat dipakai untuk mengetahui atau membaca kondisi dari terminal seperti mengetahui kondisi sinyal, kondisi baterai, nama operator, lokasi,

menambah item pada daftar *telephone*, mengetahui model *telephone* selular yang dipakai, nomor IMEI (*Internasional Mobile Statition Equipment Identity*) dan informasi – informasi lainnya yang berhubungan dengan *telephone* selular tersebut.

Perintah – perintah *AT-Command* dikirimkan ke *telephone* selular dalam bentuk *string* (teks). Komunikasi data antara *telephone* selular dengan peripheral lainnya seperti mikrokontroler dilakukan secara serial menggunakan perintah-perintah AT (*Hayes AT-Command*) melalui komunikasi serial RS-232. Berikut adalah beberapa jenis perintah *AT-Command* pada Tabel 3.

Tabel 3. Jenis- Jenis Perintah AT-Command

Perintah	Fungsi
AT+CPBF	Mencari nomor telephone yang tersimpan
AT+CPBR	Membaca buku telephone
AT+CPBW	Menulis nomor telephone di buku telephone
AT+CMGF	Menyeting mode SMS teks atau PDU
AT+CMGF=0	Menyeting mode PDU
AT+CMGF=1	Menyeting mode SMS teks
AT+CMGS	Mengirim sebuah perintah SMS
AT+CMGR	Membaca sebuah pesan
AT+CMGR=1	Membaca sebuah pesan di alamat 1
AT+CMG	Melihat semua daftar SMS yang ada
AT+CMGD	Menghapus sebuah SMS
AT+CMNS	Menyeting sebuah lokasi penyimpanan SMS
AT+COPS?	Mengetahui sebuah nama provider kartu GSM
AT+CSCA	Mengetahui alamat SMS <i>Center</i>
AT+CGMI	Mengetahui nama dan jenis ponsel
AT+CGMM	Mengetahui jenis ponsel
AT+CBC	Mengetahui level baterai

(David Hutajulu, 2015)

L. Motor Servo

Motor servo adalah sebuah motor dengan sistem umpan balik tertutup di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor servo merupakan salah satu jenis motor DC. Berbeda dengan motor stepper, motor servo beroperasi secara *close loop*. Poros motor dihubungkan dengan rangkaian kendali, sehingga jika putaran poros belum sampai pada posisi yang diperintahkan maka rangkaian kendali akan terus mengoreksi posisi hingga mencapai posisi yang diperintahkan. Motor servo banyak digunakan pada peranti R/C (*Remote Control*) seperti mobil, pesawat, helikopter, dan kapal, penggerak pada kamera serta sebagai aktuator robot. Pada robot boat pengintai, motor servo digunakan sebagai pengendali kamera pengintai (Ahlina, 2015).

Adapun bentuk fisik dari motor servo yang dapat dilihat pada Gambar 16 sebagai berikut:



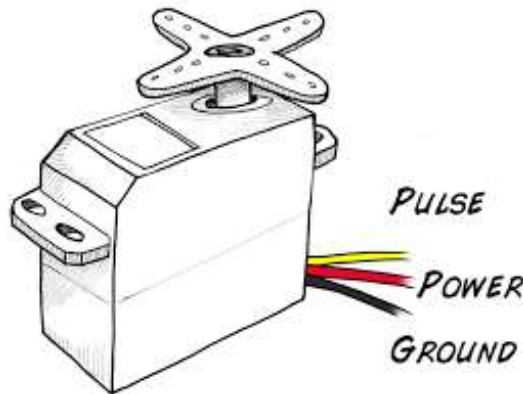
Gambar 16. Motor Servo DC

(Fahmi, 2010)

1. Kontruksi Motor Servo

Motor servo merupakan motor yang berputar lambat, dimana biasanya ditunjukkan oleh rate putarannya yang lambat, namun demikian memiliki torsi yang kuat karena pada internal *gear*-nya. Motor servo memiliki 3 kabel yaitu putih

sebagai I/O pin, merah sebagai VCC dan hitam sebagai *ground*. Dengan demikian, motor servo dapat dikontrol melalui kabel I/O yang berwarna putih. Berikut ini Pin *out* dari motor servo yang dapat dilihat pada Gambar 17 sebagai berikut:



Gambar 17. Pin out Motor Servo

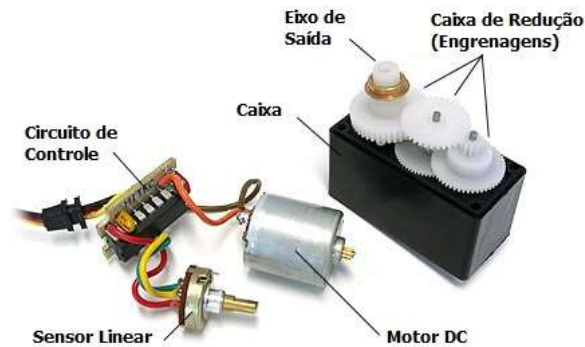
(Fahmi, 2010)

Berikut ini beberapa karakteristik motor servo (Ahlina, 2015) sebagai berikut:

- a. Terdapat tiga jalur yang meliputi jalur *power*, *ground*, *control*.
- b. Sinyal *control* digunakan untuk mengendalikan posisi.
- c. Operasional dari motor servo dikendalikan oleh pulsa sebesar 20 ms, dimana lebar pulsa antara 500 μ s dan 2400 μ s menyatakan akhir dari *range* sudut maksimum.
- d. Konstruksi di dalamnya meliputi *internal gear*, potensiometer, dan *feedback control*.

Di dalam motor servo terdapat potensiometer yang digunakan sebagai sensor posisi. Potensiometer tersebut dihubungkan dengan *output shaft* untuk mengetahui sudut posisi dari *output gear* pada motor servo. Ketika motor DC (*Direct Current*) berputar, maka *output shaft* juga berputar dan sekaligus memutar potensiometer.

Rangkaian control kemudian dapat membaca kondisi potensiometer tersebut untuk mengetahui posisi *actual shaft*. Jika posisinya sesuai dengan yang diinginkan, maka motor DC akan berhenti. Sudut operasi motor servo (*operating angle*) bervariasi tergantung jenis motor servo. Berikut ini konstruksi dari motor servo dapat dilihat pada Gambar 18 sebagai berikut:

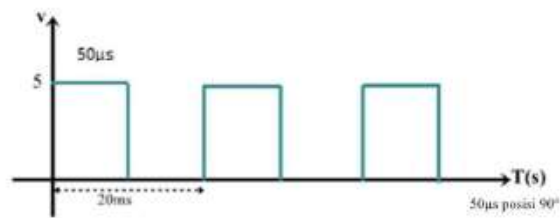


Gambar 18. Kontruksi Motor Servo
(Yama, 2009)

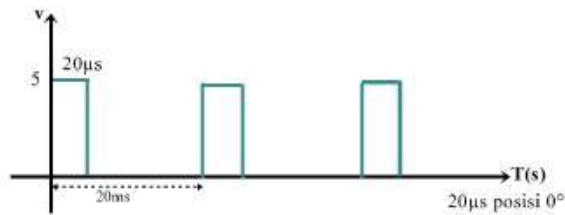
2. Jenis-Jenis Motor Servo

a. Motor Servo Standar

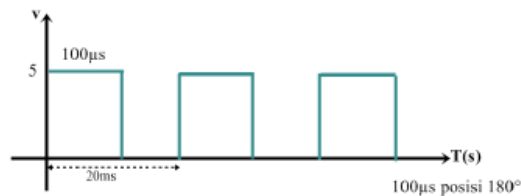
Motor servo jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180° . Pengaturan motor servo diperlukan untuk mengetahui gerakan dari motor servo dan pula yang harus kita berikan untuk bergerak ke kanan atau bergerak ke kiri. Teknik PWM (*Pulse Width Modulation*) digunakan untuk mengatur sudut motor servo jenis standard 180° ini dapat di lihat pada gambar 2.42 . Pada motor servo jenis standard 180° memiliki 3 sudut yaitu pada saat kondisi sudut 0° , 90° dan 180° .



(a)



(b)



(c)

Gambar 19. Teknik Pengaturan Sudut Pada Motor Servo

(Makruf, 2013)

Sudut dari motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Sebagai contoh, dengan pulsa $50 \mu\text{s}$ pada periode delay selebar 20 ms maka sudut dari motor servo akan berada pada posisi netral atau 90° sedangkan pada saat pulsa $\leq 20 \mu\text{s}$ pada periode *delay* selebar 20 ms maka sudut dari motor servo akan berada pada posisi 0° dan untuk pulsa $100 \mu\text{s}$ pada periode *delay* sebesar 20 ms maka sudut dari motor servo akan berada pada posisi 180° .

Pada motor servo standar 180° semakin lebar pulsa *off* maka akan semakin besar gerakan sumbu ke arah jarum jam dan semakin kecil pulsa *off* maka akan semakin besar gerakan sumbu berlawanan dengan arah jarum jam sehingga

semakin besar pulsa yang masuk melalui kaki pin motor servo maka semakin besar sudut yang dihasilkan. Berikut ini arah putaran motor servo standar yang ditunjukkan pada gambar berikut:



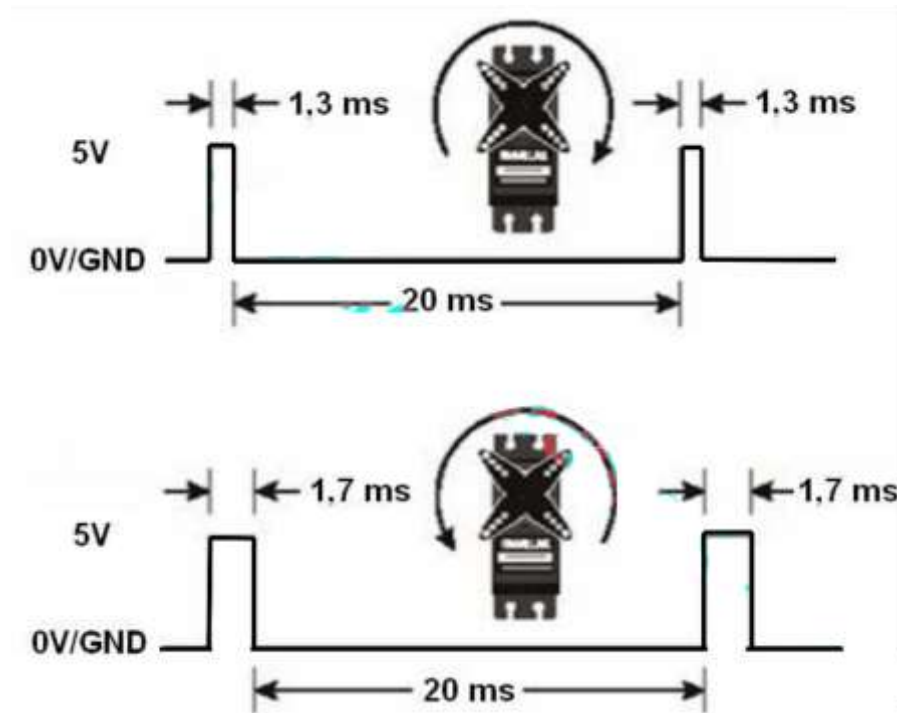
Gambar 20. Arah Putaran Motor Servo Standar

(Makruf, 2013)

b. Motor Servo Kontinu

Motor servo kontinu merupakan motor servo yang bagian feedback-nya dilepas sehingga motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar secara kontinu). Prinsip kerja dari motor servo *continuous* sedikit berbeda dari motor servo standar. Untuk menggerakkan motor servo ke kanan atau ke kiri, tergantung dari nilai *delay* yang diberikan. Agar servo pada posisi *center*, berikan pulsa 1.5 ms dan untuk pemberian pulsa ≤ 1.3 ms motor servo akan berputar searah jarum jam dengan besar putaran sumbu ditentukan oleh besar pulsa *On* pada motor sedangkan untuk membuat motor servo *continuous* berputar berlawanan dengan arah jarum jam dapat memberikan pulsa ≥ 1.7 ms, dan dengan besar pulsa *On* yang digunakan, dapat menentukan besar putaran untuk berlawanan dengan arah jarum jam. Teknik PWM (*Pulse Width Modulation*) pada motor servo *continuous* dapat dilihat pada Gambar 20. Pada

motor servo *continuous* mampu bergerak dua arah yaitu searah jarum jam dan berlawanan dengan arah jarum jam tanpa adanya batasan defleksi sudut putar (dapat berputar secara kontinyu sehingga motor ini berputar 360°).



Gambar 21. Arah Putaran Motor Servo *Continuous* 360°

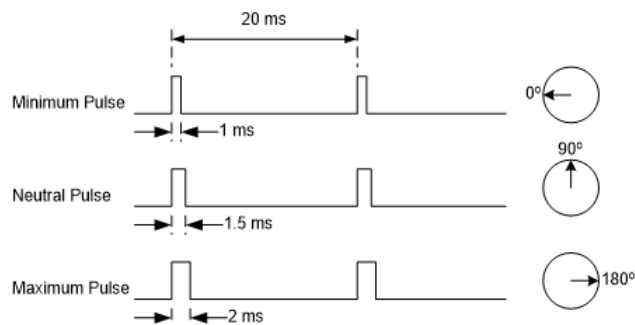
(Makruf, 2013)

Saat ini terdapat dua jenis motor servo, yaitu motor servo AC dan DC. Motor servo AC yaitu motor servo yang penggunaannya untuk digunakan pada arus yang tinggi atau beban berat, sehingga jenis ini sering diaplikasikan pada mesin-mesin industri. Sedangkan jenis motor servo DC biasanya lebih cocok digunakan pada aplikasi-aplikasi yang memiliki arus dan beban lebih kecil dibandingkan jenis motor servo AC.

Sementara apabila ditinjau dari rotasinya, terdapat dua jenis motor servo sebagai berikut:

1. Motor servo *standard* (*servo rotation* 180°) adalah jenis yang paling umum dari motor servo, dimana putaran poros outputnya terbatas hanya 90° kearah kanan dan 90° kearah kiri. Dengan kata lain total putarannya hanya setengah lingkaran atau 180° .
2. Motor servo *rotation continuous* merupakan jenis motor servo yang sebenarnya sama dengan jenis servo *standard*, hanya saja perputaran porosnya tanpa batasan atau dengan kata lain dapat berputar terus, baik ke arah kanan maupun kiri.

Pada dasarnya semua tipe servo menerima sinyal masukan dengan konsep yang sama, sinyal PWM (*Pulse Width Modulation*). Pada umumnya, suatu servo perlu menerima sinyal positif setiap 20 ms, dengan lebar-pulsa-positif maksimal 2 milidetik, dan minimalnya 1 milidetik (pada tipe Servo tertentu nilainya berbeda). Pada Gambar 23, ditunjukkan besar data sinyal PWM terhadap posisi sudut statis *output* poros dari tipe Servo Rotasi Sudut. Sedangkan pada Servo Rotasi Kontinu, lebar-pulsa-positif akan menentukan arah dan kecepatan putaran Servo.



Gambar 22. Pulsa Pengendalian Motor Servo Rotasi Sudut

(www.depokinstruments.com)

Motor servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM (*Pulse Width Modulation*) dengan frekuensi 50 Hz. Dimana pada saat sinyal dengan frekuensi 50 Hz tersebut dicapai pada kondisi *Ton duty cycle* 1.5 ms, maka rotor dari motor akan berhenti tepat di tengah-tengah (sudut 0°/netral). Pada saat *Ton duty cycle* dari sinyal yang diberikan kurang dari 1.5 ms, maka rotor akan berputar ke berlawanan arah jarum jam (*Counter Clock wise, CCW*) dengan membentuk sudut yang besarnya linier terhadap besarnya *Ton duty cycle*, dan akan bertahan diposisi tersebut. Dan sebaliknya, jika *Ton duty cycle* dari sinyal yang diberikan lebih dari 1.5 ms, maka rotor akan berputar searah jarum jam (*Clock Wise, CW*) dengan membentuk sudut yang linier pula terhadap besarnya *Ton duty cycle*, dan bertahan diposisi tersebut.

M. LCD (*Liquid Crystal Display*)

Menurut *Liquid Cristal Display* (LCD) adalah suatu media penampil dari bahan cairan kristal yang pengoperasiannya menggunakan sistem dot matriks (Andrianto, 2013). LCD banyak digunakan pada beberapa alat-alat elektronika seperti kalkulator, multimeter digital, jam digital dan sebagainya. LCD dapat dengan mudah dihubungkan dengan mikrokontroler Arduino Nano. Pada proyek akhir ini, LCD yang digunakan yaitu jenis LCD 20x4, dimana memiliki lebar *display* 4 baris, 20 kolom, dan memiliki 16 pin konektor. Adapun bentuk fisik dari LCD 20x4 ditunjukkan pada Gambar 23 sebagai berikut:



Gambar 23. LCD Karakter 20x4

(<http://site.gravitech.us/>)

Saat ini LCD banyak digunakan sebagai media penampil informasi. Hal ini disebabkan karena dibandingkan dengan media penampil lainnya, LCD hanya menggunakan arus yang kecil (beberapa micro ampere), sehingga alat atau sistem menjadi portable karena dapat menggunakan catu daya yang kecil. Selain itu, tampilan yang diperlihatkan dapat dibaca dengan mudah di bawah terang sinar matahari. Di bawah sinar cahaya yang remang-remang dalam kondisi gelap, sebuah lampu (led) harus dipasang dibelakang layar tampilan. LCD yang digunakan adalah jenis yang menampilkan data dengan 4 baris tampilan pada *display*. Kelebihan dari LCD ini adalah dapat menampilkan karakter *American Standard Code for Information Interchange* (ASCII), sehingga dapat memudahkan untuk membuat program tampilan, mudah dihubungkan dengan port I/O karena hanya menggunakan 8 bit data dan 3 bit kontrol, ukuran modul yang proporsional, daya yang digunakan relatif sangat kecil.

Operasi dasar pada LCD terdiri dari empat, yaitu instruksi mengakses proses *internal*, instruksi menulis data, instruksi membaca kondisi sibuk, dan instruksi membaca data. ROM pembangkit sebanyak 192 tipe karakter, tiap karakter dengan

huruf 5x7 *dot matrik*. Kapasitas pembangkit RAM 8 tipe karakter (membaca program), maksimum pembacaan 80 x 8 bit tampilan data. Perintah utama LCD adalah *display clear, cursor home, display on/off, display character blink, cursor shift, dan display shift* (Setiawan, 2011).

Berikut ini operasi dasar yang terdapat pada LCD *display* dapat dilihat pada Tabel 4 sebagai berikut:

Tabel 4. Operasi Dasar LCD

RS	R/W	Operasi
0	0	Input Instruksi ke LCD
0	1	Membaca <i>Status Flag</i> (DB ₇) dan alamat <i>counter</i> (DB ₀ ke DB ₆)
1	0	Menulis Data
1	1	Membaca Data

(Setiawan, 2011)

Adapun konfigurasi pin LCD 20x4 dapat dilihat pada Tabel 5 sebagai berikut:

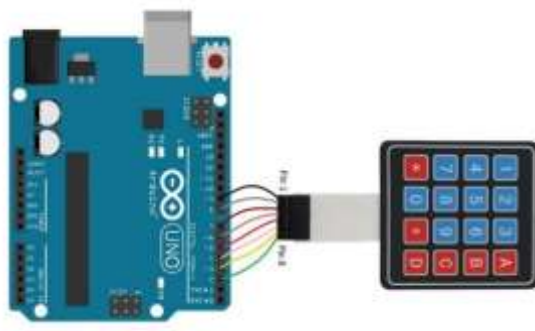
Tabel 5. Pinout LCD 20x4

No.	Nama Pin	Fungsi
1	VCC	+5V
2	GND	0V
3	VEE	Kontras LCD
4	RS	Register <i>select</i> , 0= <i>input</i> instruksi, 1= <i>input</i> data
5	R/W	1 = <i>Read</i> , 0 = <i>Write</i>
6	E	<i>Enable Clock</i>
7	D0	Data 0
8	D1	Data 1
9	D2	Data 2
10	D3	Data 3
11	D4	Data 4
12	D5	Data 5
13	D6	Data 6
14	D7	Data 7
15	LED+	Tegangan positif <i>backlight</i>
16	LED-	Tegangan negatif <i>backlight</i>

(Arianto, 2015)

N. Keypad

Keypad umumnya digunakan sebagai suatu *input* pada beberapa peralatan yang berbasis mikroprosesor atau mikrokontroler. *Keypad* sesungguhnya terdiri dari sejumlah saklar, yang terhubung sebagai baris dan kolom dengan susunan seperti yang ditunjukkan pada Gambar 25. Agar mikrokontroler dapat melakukan *scan keypad*, maka *port* mengeluarkan salah satu bit dari 4 bit yang terhubung pada kolom dengan logika *low* “0” dan selanjutnya membaca 4 bit pada baris untuk menguji jika ada tombol yang ditekan pada kolom tersebut. Sebagai konsekuensi selama tidak ada tombol yang ditekan, maka mikrokontroler akan melihat sebagai logika *high* “1” pada setiap pin yang terhubung ke baris. Terdapat beberapa jenis-jenis dari *keypad* yang sering digunakan yaitu jenis *keypad* dengan matriks 4x1, 4x2, dan 4x4. Pada alat ini *keypad* yang digunakan yaitu *keypad* dengan matriks 4x4 yang mana memiliki 16 tombol yang berfungsi dalam pengoperasian menu tampilan pada LCD (Akbar, 2007).



Gambar 24. Rangkaian Keypad dengan Arduino UNO

<http://www.theorycircuit.com>

O. LM2596

LM2596 merupakan sebuah *regulator converter* yang dapat digunakan untuk menurunkan tegangan arus searah. Regulator ini terbagi menjadi dua versi yaitu *adjustable* yang tegangan keluarannya dapat diatur dan *fixed voltage output* yang tegangan keluarannya sudah tetap. Tegangan *input* yang dapat diterima dari tegangan 4 volt hingga 35 volt yang bisa diatur keluarannya menjadi 1.23 volt sampai 30 volt dengan arus maksimal 3 ampere. Berikut ini bentuk fisik dari *converter* LM2596 dapat dilihat pada Gambar 25 sebagai berikut:



Gambar 25. Switching Power Supply LM2596

(<https://cdn3.volusion.com/>)

Adapun spesifikasi dari regulator LM2596 ini sebagai berikut.

1. *Input voltage* : DC 4 – 35 volt
2. *Output voltage* : DC 1.3 – 35 volt
3. *Output current* : rated 2A – 3 A
4. *Operatig temperatur* : -40 to +85 celcius
5. *Switching frequency* : 150 KHz
6. *Rectifier* : non-synchronous rectificaton
7. *Full load tempratur rise* : 40°C
8. *Dimensions* : panjang 4.3 cm x lebar 2cm x tinggi 1.4cm

P. Catu Daya

Catu daya adalah suatu rangkaian elektronika yang dapat menghasilkan tegangan keluar yang stabil baik berupa tegangan AC maupun tegangan DC. Sumber tegangan dan arus yang disediakan harus disesuaikan dengan konsumsi daya yang digunakan pada rangkaian agar tidak terjadinya kelebihan beban yang menyebabkan komponen bisa mengalami kerusakan. Namun, kemungkinan kerusakan tersebut dapat dicegah dengan pemilihan IC regulator yang sesuai dengan alat yang akan digunakan. Secara garis besar, catu daya elektrik dibagi menjadi dua macam sebagai berikut :

1. Catu Daya Linier

Catu daya linier merupakan jenis catu daya yang umum digunakan di pasaran yang digunakan untuk mengubah tegangan AC menjadi tegangan AC lain yang lebih kecil dengan menggunakan transformator. Tegangan ini kemudian disearahkan menggunakan rangkaian penyearah tegangan, dan pada bagian akhir rangkaian ditambahkan kapasitor sebagai pembantu menghaluskan tegangan keluaran sehingga tegangan DC yang dihasilkan oleh catu daya jenis ini tidak terlalu bergelombang. Selain menggunakan dioda sebagai penyearah, digunakan regulator tegangan sehingga tegangan yang dihasilkan lebih baik daripada rangkaian yang menggunakan dioda. Catu daya jenis ini dapat menghasilkan tegangan DC yang bervariasi antara 0 – 30 volt dengan arus antara 0 – 5 Ampere.

Berikut ini rangkaian dari catu daya linier yang dapat dilihat pada Gambar 26 sebagai berikut:



Gambar 26. Catu Daya Linier
(Steens Lasershow, 2008)

2. Catu Daya *Switching*

Tegangan AC yang masuk ke dalam rangkaian catu daya jenis ini akan langsung disearahkan oleh rangkaian penyearah tanpa menggunakan bantuan transformator. Cara menyearahkan tegangan tersebut adalah dengan menggunakan frekuensi tinggi antara 10 kHz hingga 1 MHz, dimana frekuensi ini jauh lebih tinggi daripada frekuensi AC yang sekitar 50 Hz. Pada catu daya jenis ini biasanya diberikan rangkaian *feedback* agar tegangan dan arus yang keluar dari rangkaian ini dapat dikontrol dengan baik.

Adapun rangkaian dari catu daya *switching* dapat dilihat pada Gambar 27 sebagai berikut:



Gambar 27. Catu Daya *Switching*
(Abdurrahman, 2017)

Adapun karakteristik dari IC regulator yang digunakan dalam catu daya yang dapat dilihat pada Tabel 6.

Tabel 6. Karakteristik IC Regulator

Tipe	V-out (Volt)	I-out (Ampere)			V-in (Volt)	
		78xx	78Lxx	78Mxx	Min	Max
7805	5	1	0,1	0,5	7,5	20
7806	6	1	0,1	0,5	8,6	21
7808	8	1	0,1	0,5	10,6	23
7809	9	1	0,1	0,5	11,7	24
7810	10	1	0,1	0,5	12,7	25
7812	12	1	0,1	0,5	14,8	27
7815	15	1	0,1	0,5	18	30
7818	18	1	0,1	0,5	21	33

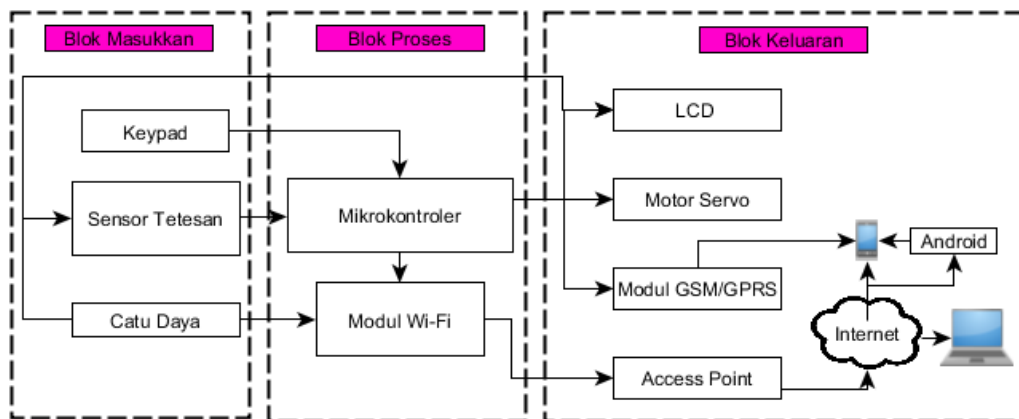
(Muhammad Dwi Maulana, 2017)

BAB III

KONSEP RANCANGAN

Automatic stoping and infusion monitoring with telemetry system based on android ini digunakan untuk memantau jumlah cairan infus setiap saat dan pemberhenti laju cairan infus saat batas habis. Pemantau jumlah tetesan infus menggunakan sensor tetesan yang memiliki akurasi tepat dan presisi. Alat ini dapat memantau secara langsung melalui LCD 20x4, website, ataupun aplikasi *smartphone* dengan *Operating System* (OS) Android. Pemberhenti laju cairan infus menggunakan motor servo sedangkan, sistem pemberitahuan melalui panggilan menggunakan modul modul GSM/GPRS. Komponen-komponen yang dipakai dalam perancangan alat ini harus memiliki ketahanan yang lama. Hal ini dimaksud agar alat dapat bekerja dengan baik apabila digunakan dalam jangka panjang.

Secara umum, proses perancangan pada *automatic stoping and infusion monitoring with telemetry system based on android* dapat dijelaskan pada Gambar 28 berikut dibawah ini.



Gambar 28. Diagram Blok Sistem

Pada gambar 28 dapat diketahui cara kerja dari alat tersebut pertama menyalakan saklar pada alat. kedua memilih infus yang digunakan menggunakan *keypad*. Ketiga memasukkan batas jumlah tetesan dan nomor telepon perawat atau dokter. Keempat tampil tetesan pada LCD, website, dan aplikasi android. Alat ini dapat menghentikan laju tetesan infus dan melakukan *miscall* jika infus mencapai batas maksimal, mengalami kemacetan, dan saat habis.

A. Identifikasi Kebutuhan

Tahapan dalam membuat *automatic stoping and infusion monitoring with telemetry system based on android*. Komponen yang dibutuhkan adalah sebagai berikut :

1. Bagian Catu Daya
 - a. Rangkaian *switching* dan regulator sebagai catu daya yang stabil untuk penunjang kerja alat.
 - b. *Step down adjustable* sebagai kontrol tegangan masuk supaya sesuai kebutuhan komponen.
2. Bagian Masukan
 - a. *Switch ON/OFF* untuk menyalakan alat.
 - b. *Push button* berfungsi untuk *me-reset* alat.
 - c. *Keypad 4x4* berfungsi sebagai kemudahan dalam pengontrolan alat.
 - d. Sensor cahaya photodiode dan laser diode berfungsi untuk mendeteksi tetesan.

3. Bagian Proses

- a. Komponen kendali sebagai pengendali sistem.
- b. Modul WiFi sebagai pengiriman data ke internet secara serial kemudian ditampilkan kedalam aplikasi.
- c. Arduino IDE sebagai aplikasi editor untuk pemrograman pada Arduino dan modul WiFi.

4. Bagian Keluaran

- a. LCD sebagai penampil tetesan infus setiap saat serta, untuk kemudahan dalam pengaturan alat.
- b. *ThingSpeak* sebagai penampil data tetesan infus pada tampilan *webpage*.
- c. *ThingView* sebagai penampilan data tetesan infus pada android.
- d. *Acces Point* sebagai pengatur lalu lintas data *client* untuk terhubung melalui jaringan.
- e. Motor servo sebagai penghenti laju infus saat infus mencapai batas habis.
- f. Modul GSM/GPRS sebagai sistem pemberitahuan melalui panggilan.

B. Analisis Kebutuhan

Berdasarkan analisis kebutuhan di atas, maka diperoleh beberapa kebutuhan, yang terdiri sebagai berikut :

1. Bagian Catu Daya

Catu daya yang digunakan pada alat ini adalah catu daya yang memiliki keluaran 12 volt. Pada catu daya tersebut, rangkaian nantinya dihubungkan dengan rangkaian dua buah *step down* LM2596. Kemudian tegangan dari 12 volt tersebut

diturunkan menjadi 9 volt dan 5 volt. Tegangan 9 volt tersebut digunakan untuk menyuplai *board* Arduino Mega 2560 sedangkan tegangan 5 volt digunakan untuk menyuplai komponen-komponen yang lain.

2. Bagian Masukan

Saklar atau *switch On/Off* dibutuhkan pada alat ini. *Switch* tersebut digunakan untuk pemutus arus dari rangkaian *switching*. Sementara *keypad* diperlukan untuk membantu *user* dalam memasukkan nomor telepon dan melakukan pemilihan infus. Hal ini disebabkan data yang diproses berasal dari sensor yang berbeda-beda sehingga pemberian *keypad* sangat diperlukan. Data masukan pada alat berasal dari hasil pengukuran sensor cahaya photodiode dan laser diode. Pada alat ini *push button ON* digunakan untuk melakukan *reset* alat yang dibuat apabila secara tiba-tiba mengalami *error system* atau pengguna menginginkan alat diatur ulang.

3. Bagian Proses

Proses merupakan bagian yang penting dalam menunjang bekerjanya semua komponen. Alat ini membutuhkan komponen yang dapat mengolah data masukan untuk selanjutnya dikirim ke bagian keluaran. Pada bagian ini digunakan Arduino Mega 2560 digunakan untuk kendali utama alat. Pemilihan Arduino Mega 2560 karena jenis ini memiliki empat pasang pin serial yang mana dirasakan sudah cukup sesuai keperluan. Modul WiFi yang digunakan menggunakan Wemos D1 Mini yang sudah terintegrasi dengan ESP8266 yang berfungsi sebagai penerima data tetesan infus yang diperoleh dari Arduino Mega 2560. Selanjutnya data tersebut dikirim secara *real time* ke ThingSpeak maupun ThingView. Arduino IDE merupakan aplikasi pemrograman yang digunakan untuk di-*upload* ke *chip*

Arduino. Pemilihan aplikasi ini karena selain *freeware*, *open source*, dan umum digunakan dalam pembuatan program dengan *board* Arduino.

4. Bagian Keluaran

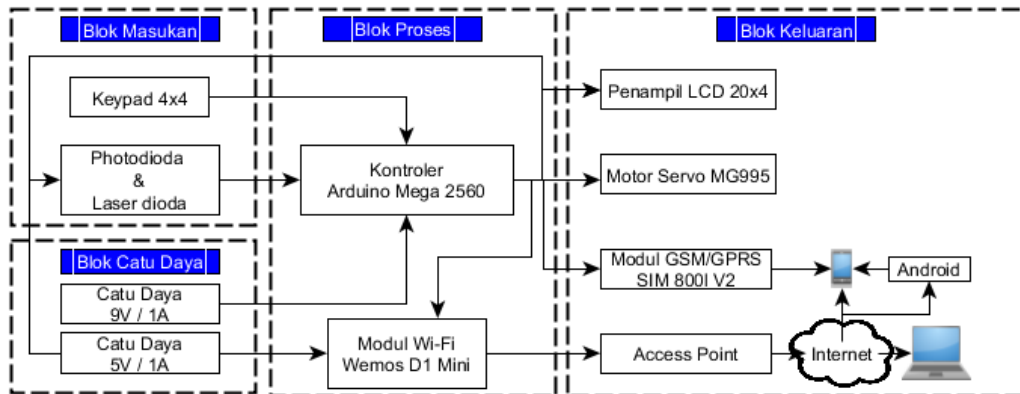
Komponen utama pada bagian keluaran yaitu LCD *display*, *ThingSpeak*, dan *ThingView*. Ketiganya media penampil tersebut dipilih karena dapat memudahkan *user* dalam memantau cairan infus pada berbagai *device* yang ada. Namun dari ketiga media tersebut terdapat kelebihan dan kekurangan masing-masing seperti LCD *display* yang data hasil pengukurannya hanya dapat dilihat secara langsung, *ThingSpeak* dan aplikasi *ThingView* yang mana hanya dapat dilihat dengan terhubung jaringan internet serta data yang masuk hanya 15 detik sekali yang tampil. *Access Point* pada bagian ini digunakan sebagai pengatur lalu lintas data *client* untuk terhubung melalui jaringan internet.

Access Point yang digunakan berasal dari *wireless hotspot* pada *smartphone* atau melalui pemancar WiFi pada *router*. Sehingga apabila mengakses website tersebut perlu dihubungkan ke jaringan WiFi terlebih dahulu. Motor servo MG955 digunakan untuk menghentikan laju cairan infus pada batas habis, karena memiliki torsi yang cukup. Modul GSM/GPRS SIM800L berfungsi sebagai sistem yang memberitahukan lewat panggilan telepon infus macet atau mengalami kehabisan. Adapun keseluruhan kebutuhan komponen baik utama maupun penunjang pada *automatic stopping and infusion monitoring with telemetry system based on android* ini disajikan pada Tabel 7.

Tabel 7. Kebutuhan Komponen

No.	Bagian	Nama Komponen	Spesifikasi	Jumlah
1.	Catu Daya	IC Regulator	LM7812	1 buah
		Dioda	IN4002	4 buah
		Capasitor	22Pf	1 buah
			2200Uf	1 buah
		LM2596 +9V	9 volt	1 buah
		LM2596 +5V	5 volt	1 buah
Switching 12V/3A	-	1 buah		
2.	Masukan	<i>Switch ON/OFF</i>	40 mm x 30 mm	1 buah
		<i>Push Button</i>	30 mm x 15 mm	1 buah
		<i>Keypad</i>	4 kolom x 4 baris	1 buah
		Laser dioda	Warna merah	2 buah
		Sensor cahaya photodiode	Diameter 5mm	2 buah
3.	Proses	Mikrokontroler	Arduino Mega2560	1 buah
		Modul WiFi	Wemos D1 Mini	1 buah
4.	Keluaran	LCD	LCD 20x4	1 buah
		Website	ThingSpeak	-
		Aplikasi	ThingView	-
		<i>Acces Point</i>	Jaringan wifi	-
		Motor Servo	MG995	2 buah
		Modul GSM/GPRS	SIM 8001 V2	1 buah
5.	Bahan Pendukung	Modul IIC	LCD 20x4	1 buah
		PCB	15 cm x 15 cm	1 buah
		<i>Kabel Jumper</i>	-	Secukupnya
		<i>Box akrilik chamber</i>	70 mm x 30 mm x 30 mm	2 buah
		<i>Box akrilik kontrol</i>	156 mm x 136 mm x 96 mm	1 buah
		<i>Box akrilik servo</i>	70 mm x 50 mm x 75 mm	2 buah
		Mur, baut, dan <i>spacer</i>	M2, M3, M4, M5, M6, M7	Secukupnya

C. Blok Diagram



Gambar 29. Blok Diagram Keseluruhan

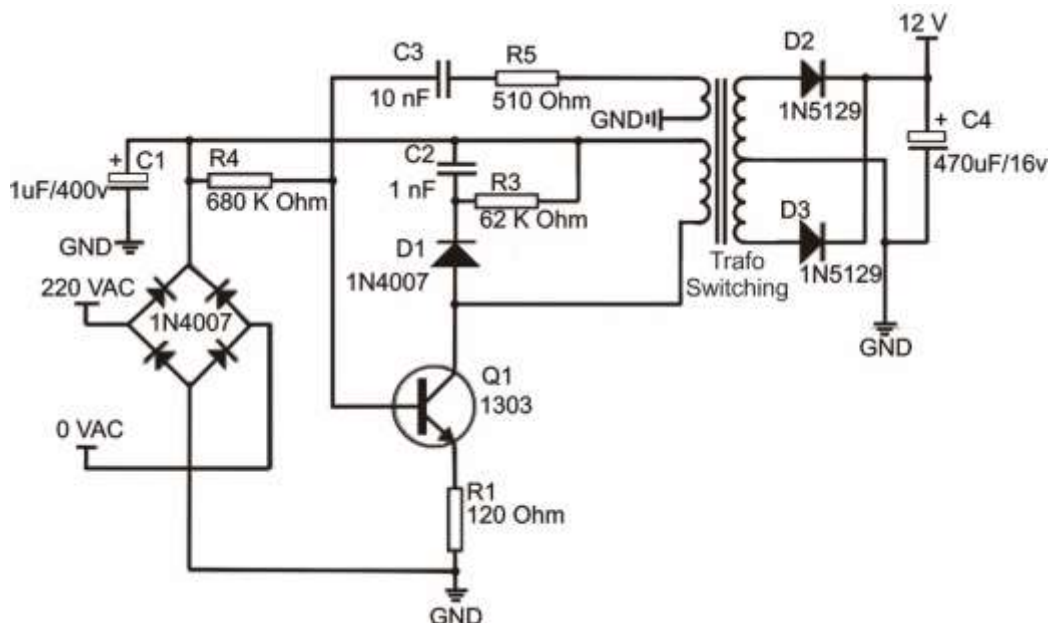
Gambar 29 adalah blok diagram rangkaian sistem secara keseluruhan dengan proses yang diaplikasikan pada *automatic stoping and infusion monitoring with telemetry system based on android*. Alur kerja pengukuran alat yaitu setelah pada saat menu tetesan dan nomor telepon dimasukkan maka sensor photodiode dengan laser diode dan modul GSM GPRS SIM800L mulai aktif bekerja untuk melakukan penghitungan tetesan infus dengan hasilnya ditampilkan pada LCD *display*. Setelah ditampilkan pada LCD *display* maka data tersebut dikirimkan ke modul WiFi untuk ditampilkan pada *Thingspeak* dan *Thingview*. Pada saat infus mengalami macet atau cairannya tak terdeteksi (habis) maka selang infus secara otomatis menghentikan laju cairan dengan cara membuat tekukan pada selang infus tersebut menggunakan motor servo dan apabila selang sudah tertekuk dengan baik maka alat melakukan panggilan ke nomor yang telah dimasukkan. Alat tersebut bekerja secara terus-menerus hingga tombol *switch* ditekan pada kondisi OFF.

D. Perancangan Sistem

Perancangan *automatic stoping and infusion monitoring with telemetry system based on android* terdiri menjadi beberapa blok sebagai berikut:

1. Blok Rangkaian Catu Daya

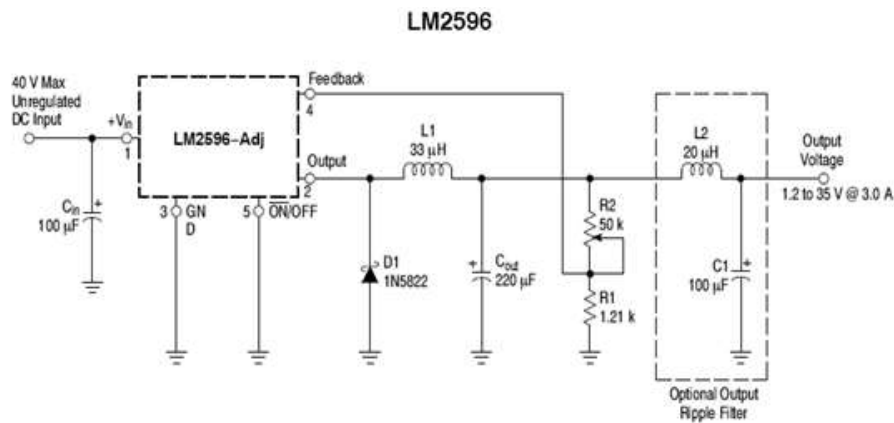
Catu daya utama yang digunakan pada alat ini adalah *power supply switching* 12V/3.5A, kemudian tegangan diturunkan menggunakan *step down adjustable*. Alat ini menggunakan dua buah *step down* LM2596. LM2596 yang pertama diatur tegangannya menjadi 9V dan LM2596 kedua diatur menjadi 5V. Tegangan 9V digunakan untuk menyuplai Arduino Mega 2560, sedangkan tegangan 5V digunakan untuk sensor, motor servo, modul wifi, dan modul GSM. Pemilihan jenis tersebut dikarenakan tegangan mudah diatur, stabil, tahan panas, dan dapat digunakan dalam jangka panjang.



Gambar 30. Rangkaian *Switching Power Supply* 12 Volt

(<http://zoniaelektro.net>)

Gambar 30 merupakan desain skematik rangkaian *switching power supply* 12 volt. Prinsip kerja rangkaian tersebut adalah menyearahkan tegangan AC 220 volt dan menggunakan dioda *bridge* D1 dan kapasitor C1 sebagai filternya. Kemudian tegangan DC yang masih bertegangan tinggi tersebut dibentuk menggunakan sistem regulator PWM tentunya dengan *power regulator* transistor (Q1) yang berfungsi sebagai pengendali transformator. Setelah itu *output* dari transformator yang masih berupa tegangan AC dengan frekuensi yang masih tinggi kemudian tegangan tersebut diproses dan disearahkan menggunakan sistem penyearah setengah gelombang dengan menggunakan filter kapasitor bernilai kecil.



Gambar 31. Rangkaian Step Down LM2596 Adjustable

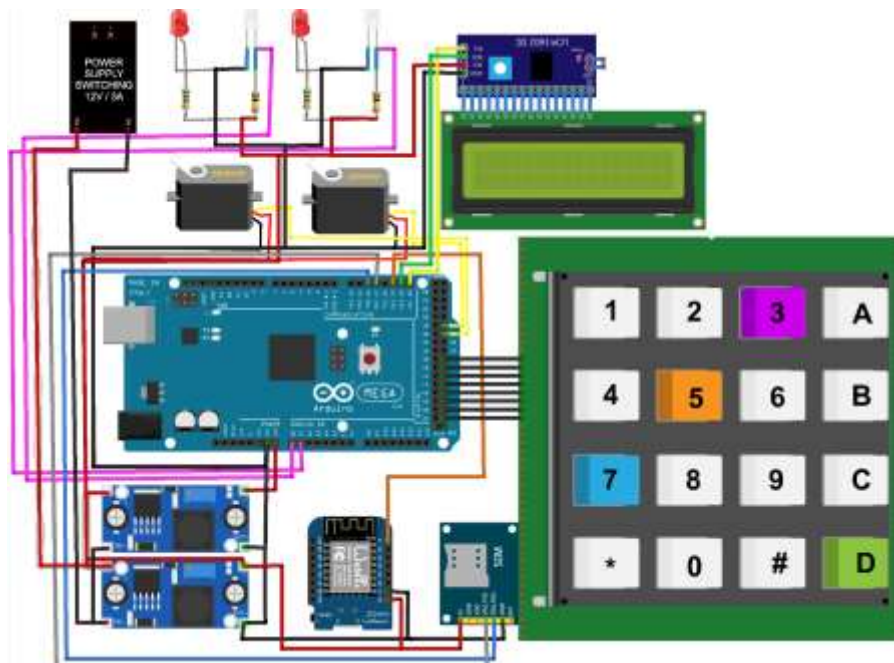
(Marsono, 2014)

Step down ini menggunakan IC LM2596 untuk mengubah tingkatan tegangan (*voltage level*) arus searah menjadi lebih rendah dibanding tegangan masukannya. Tegangan masukan (*input voltage*) dapat dialiri tegangan berapapun antara 3 volt - 40 volt yang diubah menjadi tegangan yang lebih rendah di antara 1,5 volt - 35 volt. Besar arus berkelanjutan (*continuous current*) yang dapat ditangani modul elektronika ini sebesar $\pm 1,5A$ dengan arus puncak 3A.

Arus 3A hanya untuk waktu yang sangat singkat, nilai 3A ini jangan dijadikan acuan. Tegangan keluaran yang diinginkan dapat disetel dengan memutar sekrup pada potensiometer dengan catatan perbedaan tegangan antara tegangan masukan dengan tegangan keluaran minimal 1,5 volt. Pada alat ini menggunakan dua buah LM2596. LM2596 pertama diatur menjadi 9 volt untuk menyuplai Arduino Mega 2560, sedangkan LM2596 kedua diatur menjadi 5 volt sebagai catu daya modul sensor *infrared*, Modul GSM/GPRS SIM8001 V2, motor servo, dan Wemos D1 Mini.

2. Blok Alur Rangkaian Mikrokontroler

Pada *automatic stopping and infusion monitoring with telemetry system based on android* terdiri dari keypad 4x4, sensor photodiode dengan laser diode, Arduino Mega 2560, Wemos D1 Mini, motor servo, modul GSM/GPRS SIM8001 V2, dan LCD 20x4 yang jika dirangkai menjadi satu seperti Gambar 32 sebagai berikut:



Gambar 32. Alur Rangkaian Mikrokontroler

Rangkaian terdiri dari dua jenis masukan yang pertama menggunakan *keypad* 4x4. *Keypad* 4x4 terdiri dari 16 *pushbutton* yang tersusun secara paralel yang terhubung pada 8 pin digital Arduino Mega 2560 yaitu 53, 51, 49, 47, 45, 43, 41, dan 39. *Keypad* digunakan untuk memilih infus yang dipantau kemudian memasukkan nomor telepon yang dapat dihubungi oleh alat ini. Masukan yang kedua menggunakan dua buah sensor cahaya photodiode dengan laser diode. Sensor ini terhubung ke Arduino Mega 2560 pada pin *analog output* 0 dan 1. Wemos D1 Mini terhubung ke mikrokontroler secara serial. Pin mikrokontroler TX1 (D19) terhubung dengan pin Wemos D1 mini yaitu RX2 (D7) kemudian Wemos D1 mini mengirim datanya ke ThingSpeak dan ThingView untuk penampilan datanya.

Keluaran dari alat ini adalah dua buah motor servo yang digunakan untuk penjepitan pada selang infus yang terhubung pada pin 31 dan 33. LCD 20x4 sebagai penampil dalam kemudahan dalam memasukkan nomor telepon. LCD ini yang terhubung ke pin SDA(20) dan SCL(21). Modul SIM800L digunakan karena bentuknya yang kecil dan membutuhkan tegangan yang stabil 5V. Modul SIM800L ini terhubung ke Arduino Mega 2560 sebagai indikator saat infus habis atau macet dengan melakukan panggilan.

E. Langkah Pembuatan Alat

Langkah pembuatan proyek akhir *automatic stoping and infusion monitoring with telemetry system based on android* ini terdiri dari pembuatan PCB, pemasangan komponen pada PCB, perancangan tempat sensor tetes infus, perancangan *box*, pemasangan rangkaian ke dalam *box*, serta pemasangan mekanik *box chamber* infus dan penghenti infus.

1. Pembuatan PCB

Langkah pertama dalam pembuatan PCB adalah membuat *layout* menggunakan aplikasi ISIS pada Proteus 7 Profesional. Setelah *layout* selesai dibuat maka selanjutnya yaitu mencetak *layout* tersebut menggunakan kertas *art paper* dengan menggunakan printer laser. Apabila proses penyablonan telah selesai maka dilakukan pengecekan jalur rangkaian yang telah menempel. Hal ini dimaksudkan untuk meminimalisir adanya terputusnya jalur dalam penyablonan tersebut. Setelah semua jalur menempel pada tempatnya maka dilakukan proses pelarutan menggunakan campuran HCl, H₂O₂, dan H₂O. Perbandingan penggunaan HCl : H₂O₂ : H₂O₃ adalah 1 : 2 : 3. Setelah proses pelarutan selesai, maka PCB harus dibersihkan menggunakan air bersih agar cairan sisa-sisa pelarutan hilang. Proses selanjutnya yaitu pelubangan PCB menggunakan mesin bor pada setiap kaki komponen yang ada pada *layout*.

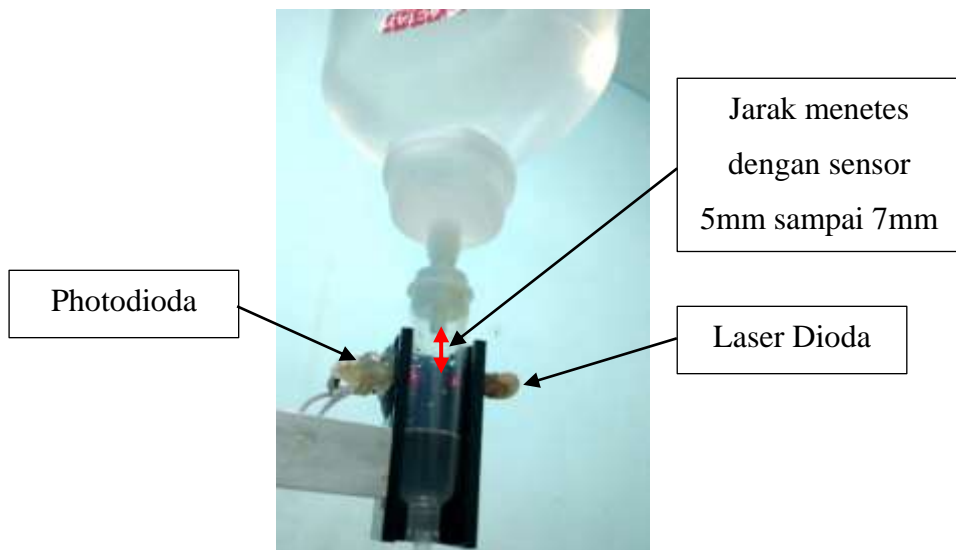
2. Pemasangan Komponen

Apabila proses pelubangan selesai, maka langkah berikutnya adalah pemasangan komponen sesuai dengan jalur komponen masing-masing. Setelah semua komponen terpasang dengan benar maka dilakukan proses penyolderan pada setiap kaki komponen tersebut menggunakan timah. Selesai seluruh komponen telah terpasang maka dilakukan pengujian jalur-jalur komponen menggunakan multimeter supaya dapat mengetahui jika ada kesalahan dalam penyolderan.

3. Perancangan Tempat Sensor Tetes Infus

Tempat sensor tetes infus dibuat dari bahan akrilik dengan bagian tengah sebagai tempat *drip chamber*, sedangkan pada bagian kiri untuk sensor photodiode

dan bagian kanan untuk laser dioda. Photodioda dan laser dioda ini harus sejajar penempatannya dan menempel pada *drip chamber* supaya dapat mendeteksi tetesan dengan akurasi tinggi. Jarak sensor dengan tetesan pada *drip chamber* juga tidak boleh terlalu naik atau turun, karena mengakibatkan perubahan nilai ADC. Jarak yang baik penempatan sensor adalah 5mm sampai 7mm. Selain itu, cahaya terlalu cerah juga harus dihindarkan dari sensor tetesan karena kan sensor tidak mendeteksi tetesan jika ada banyaknya cahaya yang masuk. Pada gambar 33 adalah contoh penempatan sensor tetesan.



Gambar 33. Bentuk Fisik Tempat Sensor Tetes Infus

4. Pemasangan Rangkaian pada *Box Control*

Setelah rangkaian selesai dibuat maka selanjutnya dapat pemasangan dan meletakkan komponen-komponen PCB ke dalam *box* yang sudah dibuat. Pemasangan tersebut harus disesuaikan dengan keadaan luas dalam *box* tersebut dan ditata secara rapi.

5. Perancangan *Box*

Pada alat ini terdiri dari dua buah bagian *box*. Kedua *box* tersebut dibuat dari bahan akrilik karena menyesuaikan dengan mekanik yang telah dibuat. *Box* pertama adalah bagian penghenti laju cairan infus. *Box* penghenti laju infus sangat minimalis karena fungsinya hanya menekuk selang infus dengan bantuan motor servo. *Box* kedua adalah bagian *box control system*. Pada *box control system* terdiri atas dua bagian dalam dan luar. Bagian dalam terdiri atas semua komponen utama seperti Arduino Mega 2560, Wemos D1 ini, Modul SIM800L, *step down* LM2596, dan *switching power supply*. Sedangkan bagian luar terdiri dari LCD 20x4, keypad 4x4, *switch ON/OFF*, *push on*, dan kabel AC. Adapun bentuk fisik dari *box* penghenti laju infus dapat dilihat pada Gambar 35 dan *box control system* Gambar 34 di bawah ini.



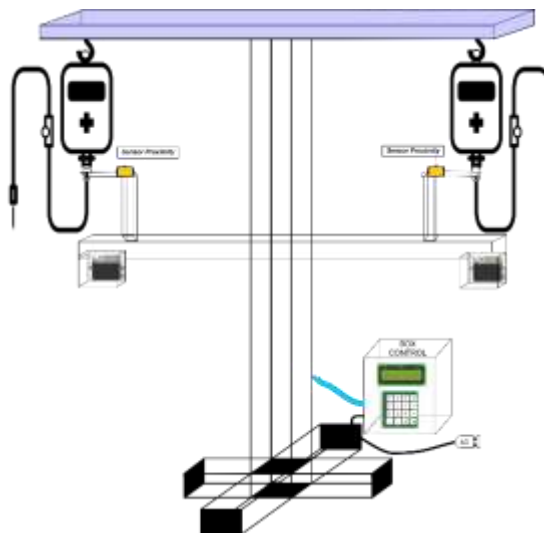
Gambar 34. Bentuk Fisik *Box* Penghenti Laju Infus



Gambar 35. Bentuk Fisik *Box Control System*

6. Pemasangan Mekanik *Box Chamber* Infus dan Penghenti Infus

Setelah langkah pembuatan *box* selesai dan hasilnya sesuai dengan yang diharapkan, maka selanjutnya adalah pemasangan mekanik alat yang telah dibuat. Berikut ini merupakan desain mekanik dari penghitung tetesan infus yang ditunjukkan pada Gambar 37.



Gambar 36. Desain Alat Penghitung Jumlah Tetesan dan Pemberhenti Laju Cairan Infus

Tiang infus terbuat dari bahan *stainless steel* dan alumunium. *Stainless steel* digunakan sebagai penyangga utama tiang, sedangkan alumunium digunakan sebagai tiang yang bersifat fleksibel sehingga tiang ini dapat dinaikan dan diturunkan sesuai kebutuhan. Tiang ini memiliki tinggi ± 150 cm sampai 180 cm. Pada bagian dalam *box* infus terdapat modul sensor photodioda dan laser dioda yang berada di *drip chamber*. Sensor photodioda dan laser dioda berfungsi sebagai pendeteksi tetesan setiap saat, mengetahui kemacetan infus, dan infus habis. Letak modul sensor ini harus di samping dan menempel pada *drip chamber* infus supaya sensor stabil, berfungsi dengan baik, dan kemudahan dalam pemakaiannya. Tiang ini juga memiliki sistem penghenti infus yang berada pada bawah tempat sensor. Penghenti infus ini menggunakan motor servo dengan menekukan bagian selangnya sehingga dapat mencegah darah pada pasien tidak naik ke kantong infus saat infus habis.

F. Perangkat Lunak

1. Arduino IDE

Perangkat lunak yang digunakan adalah *software* Arduino IDE yang digunakan untuk membuat program *hardware* Arduino. Arduino IDE adalah *software* yang sangat canggih ditulis dengan menggunakan *Java*. Adapun terdiri dari sebagai berikut:

- a. Editor program, sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing*.

- b. *Compiler*, sebuah modul yang mengubah kode program (bahasa Processing) menjadi kode biner. Bagaimanapun sebuah *microkontroler* tidak bisa memahami bahasa *Processing*. Yang bisa dipahami oleh *microkontroler* adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.
- c. *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam memori di dalam papan Arduino.

Sebuah kode program Arduino umumnya disebut dengan istilah *sketch* yang mana *sketch* digunakan secara bergantian dengan “kode program” dimana keduanya memiliki arti yang sama. Berikut ini adalah contoh tampilan IDE Arduino dengan sebuah *sketch* yang sedang diedit.



Gambar 37. Tampilan dari Software Arduino IDE

2. Algoritma Program

- a. *Start*
- b. Tampil menu pilihan infus di LCD
- c. Jika menu pengukuran tetesan infus 1
 - 1) Tekan huruf “B” pada *keypad* 4x4
 - 2) Penunjuk mengarah ke “Infus 1”
 - 3) Pilih “Infus 1”
 - 4) Tekan huruf “C” pada *keypad* 4x4

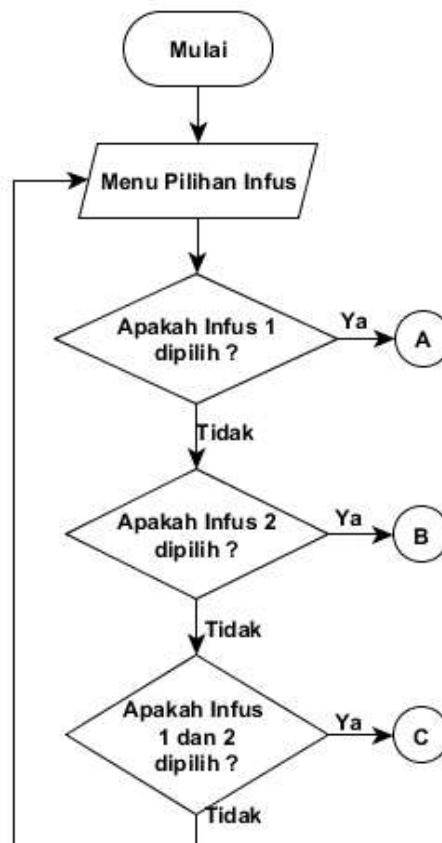
- 5) Tampil “Batas Tetesan Infus 1” di LCD
 - 6) Masukkan “Batas Tetesan Infus 1” sesuai kondisi infus di LCD
 - 7) Simpan data
 - 8) Tampil “Masukan Nomor Telepon 1” di LCD
 - 9) Masukkan nomor telepon 1 yang terdiri dari 1 sampai 13 digit
 - 10) Tekan huruf “C” pada *keypad*
 - 11) Simpan data
 - 12) Tampil tetesan infus 1 di LCD
- d. Jika menu pengukuran tetesan infus 2
- 1) Tekan huruf “B” pada *keypad* 4x4
 - 2) Penunjuk mengarah ke “Infus 2”
 - 3) Pilih “Infus 2”
 - 4) Tekan huruf “C” pada *keypad* 4x4
 - 5) Tampil “Batas Tetesan Infus 2” di LCD
 - 6) Masukkan “Batas Tetesan Infus 2” sesuai kondisi infus di LCD
 - 7) Simpan data
 - 8) Tampil “Masukan Nomor Telepon 2” di LCD
 - 9) Masukkan nomor telepon 1 yang terdiri dari 1 sampai 13 digit
 - 10) Tekan huruf “C” pada *keypad*
 - 11) Simpan data
 - 12) Tampil tetesan infus 2 di LCD
- e. Jika menu pengukuran tetesan infus 1 dan tetesan infus 2
- 1) Tekan huruf “B” pada *keypad* 4x4

- 2) Pilih “Infus 1”
 - 3) Masukkan batas tetes Infus 1
 - 4) Simpan data
 - 5) Tekan huruf “D” pada *kepad*
 - 6) Pilih “Infus 2”
 - 7) Masukkan batas tetes Infus 2
 - 8) Simpan data
 - 9) Tekan huruf “D” pada *kepad*
 - 10) Pilih “Infus 1 dan 2”
 - 11) Tampil nomor telepon 1 di LCD
 - 12) Masukkan nomor telepon 1 yang terdiri dari 1 sampai 13 digit
 - 13) Tekan huruf “C” pada *keypad*
 - 14) Simpan data
 - 15) Tampil nomor telepon 2 di LCD
 - 16) Masukkan nomor telepon 2 yang terdiri dari 1 sampai 13 digit
 - 17) Tekan huruf “C” pada *keypad*
 - 18) Simpan data
 - 19) Tampil tetesan infus 1 dan infus 2 di LCD
- f. Aktifkan *access point* jaringan *hotspot handphone*
 - g. Sambung perangkat pada jaringan
 - h. Buka alamat web <https://thingspeak.com/channels/351806>
 - i. Buka aplikasi ThingView
 - j. Masukkan *id channel* 351806

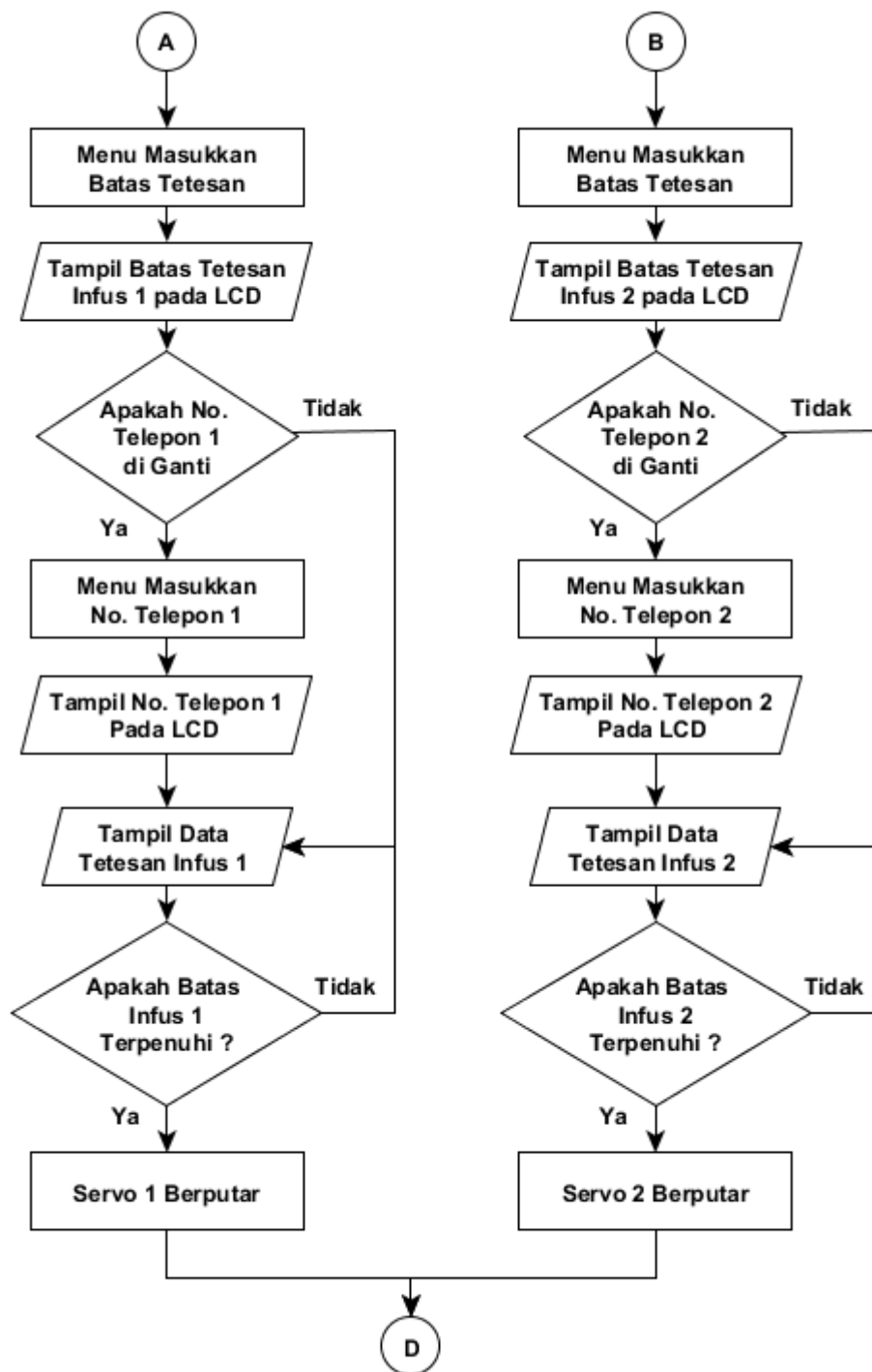
- k. Lihat hasil pengukuran
- l. Jika infus mencapai batas habis atau sudah tidak menetes selama 1 menit maka laju infus terhenti
- m. Selesai

G. Flow Chart

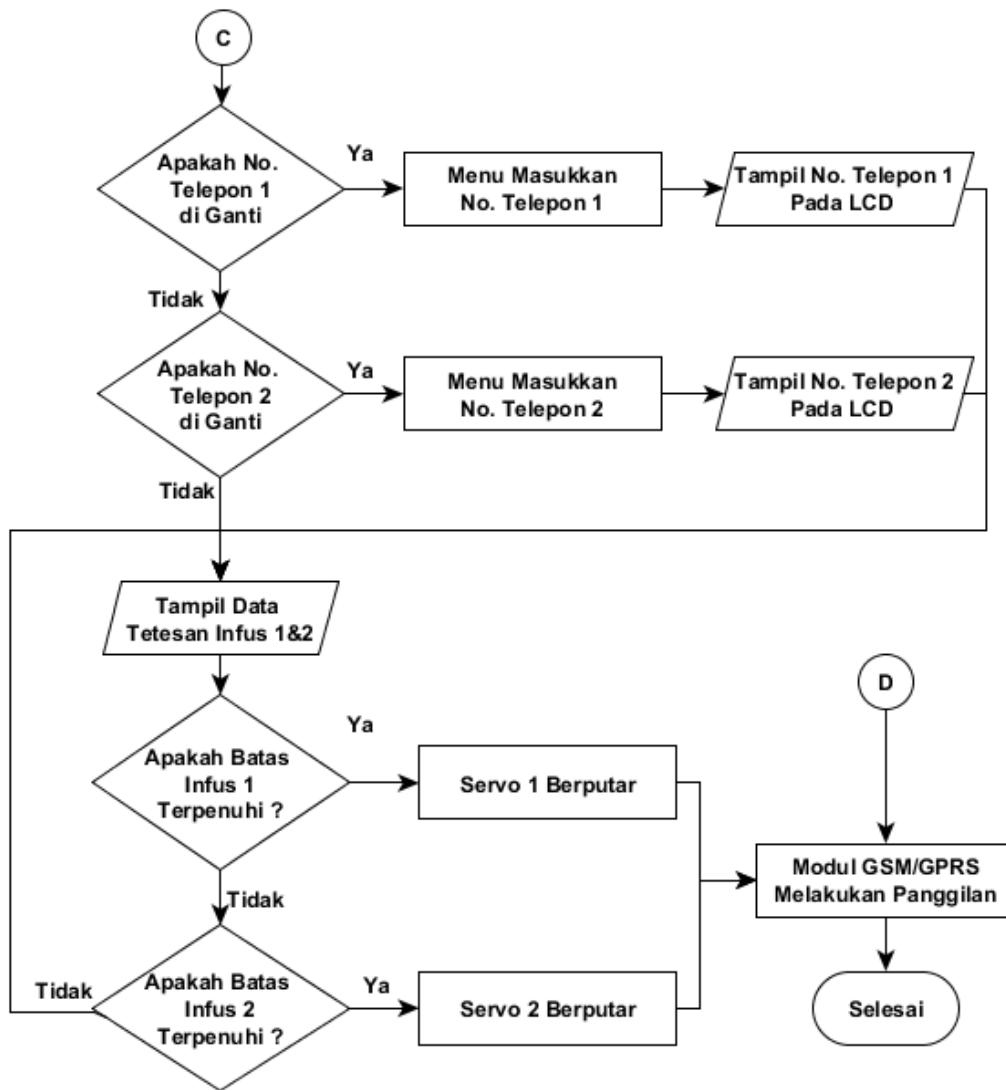
Berikut ini diagram alir dari program *automatic stoping and infusion monitoring with telemetry system based on android* yang terdiri dari tiga bagian. Bagian-bagian tersebut ditampilkan pada Gambar 38, 39, dan 49. Pada gambar tersebut, setiap bagian menu masukkan data diwakilkan dengan *process*.



Gambar 38. Flowchart Program bagian Pertama



Gambar 39. Flowchart Program bagian Kedua



Gambar 40. Flow Chart Program bagian Ketiga

H. Spesifikasi Alat

Automatic stoping and infusion monitoring with telemetry system based on android memiliki spesifikasi sebagai berikut:

1. Alat ini terbuat dari besi *stainless steel* dan aluminium sebagai tiang infus dengan tinggi antara 150 cm sampai 180 cm.

2. Tempat sensor cahaya photodiode dan laser diode diletakkan bersebelahan pada sisi kiri dan kanan pada *drip chamber*.
3. Penjepit infus menggunakan motor servo yang berbentuk seperti *box* pada bagian tengah.
4. Modul sensor cahaya photodiode dengan laser diode dipasang pada bagian samping *drip chamber* mendeteksi tetesan dengan pembacaan secara digital.
5. Terdapat *keypad* 4x4 yang digunakan untuk mempermudah memasukkan nomor telepon perawat.
6. Pengendali sistem menggunakan Arduino Mega 2560 .
7. Sumber tegangan masukan berasal dari *power supply switching* 12V/3.5A.
8. Media penampil hasil pengukur dapat dilihat melalui LCD *display*, *ThingSpeak*, dan *ThingView*.
9. Terdapat sistem notifikasi dapat melalui panggilan telepon kepada tenaga medis yang ada.
10. Menggunakan IP Address global dari Modul WiFi ESP8266 sendiri.

I. Pengujian Alat

1. Uji Fungsional

Pengujian ini dilakukan dengan cara menguji setiap alat berdasarkan karakteristik dan fungsi masing-masing. Pengujian ini dilakukan untuk mengetahui apakah setiap bagian dari perangkat telah bekerja sesuai dengan fungsi dan keinginan.

2. Uji unjuk kerja

Pengujian unjuk kerja alat dilakukan dengan cara melihat unjuk kerja alat. Hal-hal yang perlu di amati antara lain : rangkaian sensor tetesan menggunakan photodiode dan laser diode dengan Arduino Mega 2560, motor servo dengan Arduino Mega 2560, SIM8001V2 dengan Arduino Mega 2560, Wemos D1 Mini dengan LCD 20x4, *website*, dan *android*.

K. Pengoperasian Alat

Adapun prosedur pengoperasian alat ini dapat dilakukan dengan cara sebagai berikut:

1. Pastikan alat terhubung dengan tegangan listrik AC 220 volt.
2. Aktifkan tombol *switch on/off* yang telah terpasang pada posisi ON yang letaknya berada di bagian bawah LCD 20x4.
3. Pilih infus yang digunakan dengan tombol “B” untuk memilihnya.
4. Kemudian tekan tombol “C” untuk memilih infus.
5. Tekan tombol “D” untuk kembali.
6. Setelah itu masukkan nomor telepon tenaga medis yang memantau infus pasien menggunakan *keypad* 4x4.
7. Kemudian tekan tombol “C” pada *keypad* 4x4 apabila nomor yang dimasukkan sudah benar jika salah tekan tombol “A” untuk mengubahnya.
8. Setelah menekan tombol C maka alat otomatis bekerja dan dapat memantau melalui LCD *display*.

9. Kemudian aktifkan *hotspot* Wifi yang sesuai dengan program. WiFi digunakan untuk *aces point* dalam pengiriman data ke internet.
10. Setelah mengaktifkan *hotspot* WiFi dapat dilakukan pemantauan di website dan android.
 - a. Pemantauan di website dapat dilakukan dengan cara sebagai berikut :
 - 1) Buka *browser* di laptop atau *smartphone*
 - 2) Masukkan alamat web <https://thingspeak.com/channels/351806>
 - 3) Kemudian klik “ENTER”
 - 4) Kemudian tampil jumlah tetesan
 - b. Pemantauan di android dapat dilakukan dengan cara sebagai berikut :
 - 1) Buka aplikasi ThingView di *smartphone* yang menggunakan OS android
 - 2) Masukkan *id channel* [351806](https://thingspeak.com/channels/351806)
 - 3) Kemudian klik “SEARCH”
 - 4) Kemudian tampil jumlah tetesan setiap 60 detik sekali.
11. Tunggu sampai tetesan infus sampai habis, maka selang pada infus diteuk menggunakan motor servo,
12. Kemudian alat melakukan panggilan ke nomor yang telah dimasukkan selama 15 detik.
13. Saat penggantian kantong infus *switch* harus dalam kondisi *Off*.
14. Jika kondisi infus masih batas aman. Alat tidak mengirim SMS, menjepitan selang, ataupun melakukan panggilan.
15. Jika selesai mempergunakan alat maka jangan lupa mastikan tombol *switch* dalam kondisi *Off* supaya alat aman dan tidak terjadi hal yang diinginkan.

BAB IV

PENGUJIAN DAN PEMBAHASAN

Pengujian terhadap *automatic stoping and infusion monitoring with telemetry system based on android* ini dilakukan untuk mengetahui kinerja baik masing-masing komponen dan keseluruhan alat. Hasil dari pengujian alat tersebut diharapkan mampu mendapatkan data yang valid dan mengetahui apakah alat sudah bekerja sesuai dengan yang diharapkan.

A. Hasil Pengujian

1. Hasil Pengujian Tegangan Catu Daya

Pengujian catu daya dilakukan bertujuan untuk mengukur besarnya tegangan yang dibutuhkan pada setiap blok rangkaian. Pengujian ini dilakukan untuk memastikan tegangan yang keluar pada rangkaian tidak melebihi tegangan yang dibutuhkan sehingga dapat menghindarkan terjadinya kerusakan-kerusakan pada beberapa komponen yang digunakan. Untuk memastikan tegangan keluaran tersebut tidak *over voltage* maka dilakukan pengukuran sebanyak 5 kali setiap masing-masing agar dapat dipastikan tegangan yang keluar tetap stabil pada semua kondisi. Pengukuran tegangan catu daya tanpa beban, maupun dengan beban menggunakan multimeter digital SINHWA DT830B.

Tabel 8. Hasil Pengujian Tegangan Catu Daya Tanpa Beban

No.	Pengukuran pada	Pengukuran ke-	V-out (Volt)	V-out Terbaca (Volt)	Selisih Tegangan (Volt)	Error (%)
1.	<i>Switching Power Supply</i> 12V/3A	1	12	12.10	0.10	0.83
		2	12	12.10	0.10	0.83
		3	12	12.10	0.10	0.83
		4	12	12.10	0.10	0.83
		5	12	12.10	0.10	0.83

2.	<i>Step down</i> LM2596 <i>adjustable</i> +9 VDC	1	9	9.27	0.27	2.91
		2	9	9.27	0.27	2.91
		3	9	9.27	0.27	2.91
		4	9	9.27	0.27	2.91
		5	9	9.27	0.27	2.91
3.	<i>Step down</i> LM2596 <i>adjustable</i> +5 VDC	1	5	5.12	0.12	2.34
		2	5	5.12	0.12	2.34
		3	5	5.12	0.12	2.34
		4	5	5.12	0.12	2.34
		5	5	5.12	0.12	2.34

Tabel 9. Hasil Pengujian Tegangan Catu Daya dengan Beban

No.	Pengukuran pada	Pengukuran ke-	V-Out (Volt)	V-Out Terbaca (Volt)	Selisih Tegangan (Volt)	Error (%)
1.	<i>Switching Power Supply</i> 12V/3A	1	12	11.93	0.07	0.58
		2	12	11.93	0.07	0.58
		3	12	11.93	0.07	0.58
		4	12	11.93	0.07	0.58
		5	12	11.93	0.07	0.58
2.	<i>Step down</i> LM2596 <i>adjustable</i> +9VDC	1	9	8.97	0.03	0.33
		2	9	8.97	0.03	0.33
		3	9	8.97	0.03	0.33
		4	9	8.97	0.03	0.33
		5	9	8.97	0.03	0.33
3.	<i>Step down</i> LM2596 <i>adjustable</i> +5VDC	1	5	4.87	0.13	2.67
		2	5	4.87	0.13	2.67
		3	5	4.87	0.13	2.67
		4	5	4.87	0.13	2.67
		5	5	4.87	0.13	2.67

2. Hasil Pengujian Modul Sensor Tetesan

Pengujian modul sensor tetesan dilakukan dengan cara meletakkan sensor pada bagian samping *drip chamber* untuk mengetahui jumlah tetesan setiap menitnya. Pada pengujian ini bertujuan untuk mengetahui perbandingan hasil pembacaan sensor dengan jumlah tetesan secara perhitungan. Pada pengujian ini menggunakan pembanding dengan jenis infus makro yang mempunyai faktor tetes 1 ml dengan hasil 20 tetes/menit.

Tabel 10. Hasil Pengujian Sensor Tetesan Infus 1

Volume Infus (ml)	Jumlah Tetesan dengan sensor (TPM)	Jumlah tetesan dengan perhitungan (TPM)	Selisih Pengukuran	Error (%)
50	901	1000	99	10
100	2113	2000	113	5,35
200	3928	4000	72	1,8
300	5894	6000	106	1,7
400	8087	8000	87	1,07
500	9125	10000	125	1,37
Rata-Rata Kesalahan				3,5

Tabel 11. Hasil Pengujian Sensor Tetesan Infus 2

Volume Infus (ml)	Jumlah Tetesan dengan sensor (TPM)	Jumlah tetesan dengan perhitungan (TPM)	Selisih Pengukuran	Error (%)
50	1056	1000	56	5,3
100	1932	2000	68	3,52
200	3919	4000	81	2,06
300	6121	6000	121	1,98
400	7903	8000	97	1,22
500	9892	10000	108	1,09
Rata-Rata Kesalahan				2,52

3. Hasil Pengujian Penghenti Laju Infus

Sistem penghenti laju infus ini dibuat dan diletakkan pada bagian bawah *box* infus. Cara kerjanya dengan cara menekukkan atau menekan selang infus agar laju cairan infus dapat terhenti. Pengujian ini dilakukan dengan membandingkan jumlah tetesan yang dimasukkan apakah sesuai dengan jumlah tetesan infus hingga terhenti atau tidak.

Tabel 12. Hasil Pengujian Penghenti Laju Infus 1

No.	Batas infus yang dimasukkan	Infus berhenti pada tetesan ke-	Selisih Pengukuran	Error (%)
1.	110	110	0	0
2.	115	115	0	0
3.	100	100	0	0
4.	150	150	0	0
5.	200	200	0	0

Tabel 13. Hasil Pengujian Penghenti Laju Infus 2

No.	Batas infus yang dimasukkan	Infus berhenti pada tetesan ke-	Selisih Pengukuran	Error (%)
1.	25	25	0	0
2.	50	50	0	0
3.	75	75	0	0
4.	85	85	0	0
5.	99	99	0	0

4. Hasil Pengujian Sistem Notifikasi Melalui Panggilan

Pada pengujian ini bertujuan untuk mengetahui kinerja dari sistem pemberitahuan melalui panggilan. Proses pengujian dilakukan dengan cara memasukkan nilai batas infus untuk mengetahui respon sistem panggilan pada alat.

Tabel 14. Hasil Pengujian Sistem Notifikasi Melalui Panggilan Pada Infus 1

No.	Batas Infus yang Dimasukan	Panggilan Saat Tetesan Ke-	Selisih Pengukuran	Error (%)
1.	110	110	0	0
2.	115	115	0	0
3.	100	100	0	0
4.	150	150	0	0
5.	200	200	0	0

Tabel 15. Hasil Pengujian Sistem Notifikasi Melalui Panggilan Pada Infus 2

No.	Batas Infus yang Dimasukan	Panggilan Saat Tetesan Ke-	Selisih Pengukuran	Error (%)
1.	25	25	0	0
2.	50	50	0	0
3.	75	75	0	0
4.	85	85	0	0
5.	99	99	0	0

5. Hasil Pengujian Tampilan Tetesan

Pengujian pada tampilan ini berguna untuk mengetahui apakah terdapat kesalahan pada tampilan LCD, ThingsSpeak, dan ThingView. Pengujian ini dilakukan dengan menggunakan sample sebanyak 5 kali. Pengujian ini dilakukan dengan cara membandingkan hasil tampilan antara LCD dengan media penampil lainnya dan sebaiknya apakah sesuai atau tidak.

Tabel 16. Hasil Pengujian Tampilan LCD dengan website Thingspeak

No.	Jumlah Tetesan Infus pada LCD	Jumlah Tetesan Infus pada Website	Selisih Pengukuran	Error (%)
1.	31	31	0	0
2.	62	62	0	0
3.	124	124	0	0
4.	183	183	0	0
5.	241	241	0	0

Tabel 17. Hasil Pengujian Tampilan LCD dengan aplikasi Thingview

No.	Jumlah Tetesan Infus pada LCD	Jumlah Tetesan Infus pada Android	Selisih Pengukuran	Error (%)
1.	31	31	0	0
2.	62	62	0	0
3.	124	124	0	0
4.	183	183	0	0
5.	241	241	0	0

Tabel 18. Hasil Pengujian Tampilan ThingSpeak dengan ThingView

No.	Jumlah Tetesan Infus pada Website	Jumlah Tetesan Infus pada Android	Selisih Pengukuran	Error (%)
1.	31	31	0	0
2.	62	62	0	0
3.	124	124	0	0
4.	183	183	0	0
5.	241	241	0	0



6. Hasil Pengujian Jangkauan Transmisi Data Wemos D1 Mini





Tabel 19. Hasil Pengujian Jangkauan Transmisi Data

No.	Jarak (Kilometer)	Kesanggupan Akses		Sumber Tempat Pengambilan Data
		Dapat	Tidak	
1.	5	V		UNY
2.	17	V		Prambanan
3.	25	V		Bantul
4.	39	V		Klaten
5.	64	V		Solo
6.	113	V		Semarang
7.	257	V		Tuban

7. Hasil Pengujian LCD

Tabel 20. Hasil Pengujian LCD Display

No.	Pengujian	Hasil Tampilan	Keterangan
1.	Tampilan <i>Booting</i>		Sesuai
2.	Menu Tetesan		Sesuai

3.	Menu Batas Tetesan		Sesuai
4.	Menu <i>input</i> nomor Telepon		Sesuai
5.	Menu Tampil Data Tetesan		Sesuai
6.	Menu Cek Sensor		Sesuai

8. Hasil Pengujian Unjuk Kerja

Pengujian unjuk kerja dilakukan dengan cara menguji sistem alat secara keseluruhan yang dibutuhkan. Tujuan pengujian ini untuk mengetahui kekurangan dan kelebihan dari sistem yang telah dibuat. Pengujian ini dilakukan mulai dari menguji pengukuran sensor yang digunakan, tampilan, dan komunikasi transmisi datanya. Prosedur pengujian dilakukan dengan memilih pilihan menu infus yang tersedia, apakah infus 1, infus 2, infus 1 dan infus 2, atau tampil data menggunakan *keypad* yang tersedia. Selanjutnya dari data tersebut kemudian dikirimkan menuju bagian penampil agar dapat diketahui hasilnya. Adapun hasil pengujian unjuk kerja ini adalah sebagai berikut :

- a. Pengujian sensor photodiode dan laser diode

Tabel 21. Hasil Pengujian sensor photodiode dan laser diode Pada Infus 1

Volume Infus (ml)	Jumlah Tetesan dengan modul sensor <i>infrared</i>	Jumlah tetesan dengan perhitungan	Selisih Pengukuran	Error (%)
50	901	1000	99	10
100	2113	2000	113	5,35
200	3928	4000	72	1,8
300	5894	6000	106	1,7
400	8087	8000	87	1,07
500	9125	10000	125	1,37
Rata-rata error				3,5

Tabel 22. Hasil Pengujian sensor photodiode dan laser diode Pada Infus 2

Volume Infus (ml)	Jumlah Tetesan dengan modul sensor <i>infrared</i>	Jumlah tetesan dengan perhitungan	Selisih Pengukuran	Error (%)
50 ml	1056	1000	56	5,3
100 ml	1932	2000	68	3,52
200 ml	3919	4000	81	2,06
300 ml	6121	6000	121	1,98
400 ml	7903	8000	97	1,22
500 ml	9892	10000	108	1,09
Rata-rata error				2,52

b. Pengujian Motor Servo

Pengujian ini bertujuan untuk mengetahui putaran servo apakah sudah sesuai dengan program atau belum. Motor servo yang digunakan berfungsi untuk menghentikan laju infus dengan cara menekukkan atau menekan pada bagian selang infus. Cara kerja pengujian ini dengan cara memasukan batasan infus yang ingin dihentikan kemudian apakah motor servo memberikan respon memutar 180° ataupun tidak.

Tabel 23. Hasil Pengujian Motor Servo Pada Infus 1

No.	Batas Tetesan yang di-input-kan	Derajat Putaran Motor Servo pada Program (°)	Derajat Putaran Motor Servo pada Alat (°)	Selisih Putaran	Error (%)	Kondisi Laju Tetesan
1.	110	180	180	0	0	Terhenti
2.	115	180	180	0	0	Terhenti
3.	100	180	180	0	0	Terhenti
4.	150	180	180	0	0	Terhenti
5.	200	180	180	0	0	Terhenti

Tabel 24. Hasil Pengujian Motor Servo Pada Infus 2

No.	Jumlah Tetesan yang di Masukkan	Derajat Putaran Motor Servo pada Program (°)	Derajat Putaran Motor Servo pada Alat (°)	Selisih Putaran	Error (%)	Kondisi Tetesan
1.	25	180	180	0	0	Terhenti
2.	50	180	180	0	0	Terhenti
3.	75	180	180	0	0	Terhenti
4.	85	180	180	0	0	Terhenti
5.	99	180	180	0	0	Terhenti

c. Pengujian Modul GSM SIM800L

Pengujian ini bertujuan untuk mengetahui jika saat infus mendekati batas maksimum habis yang dimasukan, apakah alat tersebut memberikan notifikasi panggilan kepada *user*.

Tabel 25. Hasil Pengujian Modul SIM800L Pada Infus 1

Batas Jumlah Tetesan Yang Masukkan	Batas Jumlah Tetesan Infus Pada Alat	Selisih Batas Jumlah Tetesan	Keberhasilan Modul SIM800L Melakukan Panggilan		Error (%)
			Berhasil	Tidak	
110	50	0	V		0
115	125	0	V		0
100	500	0	V		0
150	1000	0	V		0
200	10000	0	V		0

Tabel 26. Hasil Pengujian Modul SIM800L Pada Infus 2

Batas Jumlah Tetesan Yang Masukkan	Batas Jumlah Tetesan Infus Pada Alat	Selisih Batas Jumlah Tetesan	Keberhasilan Modul SIM800L Melakukan Panggilan		Error (%)
			Berhasil	Tidak	
25	25	0	V		0
50	75	0	V		0
75	101	0	V		0
85	1025	0	V		0
99	9057	0	V		0

d. Hasil Pengujian Unjuk Kerja Tampilan ThingSpeak dan ThingView

Tabel 27. Hasil Pengujian Unjuk Kerja Tampilan pada website ThingSpeak

Percobaan	Tampilan Hasil Pengukuran	
	Berhasil	Tidak
1	V	
2	V	
3	V	
4	V	
5	V	
6	V	
7	V	
8	V	
9	V	
10	V	

Tabel 28. Hasil Pengujian Unjuk Kerja Tampilan ThingView

Percobaan	Tampilan Hasil Pengukuran	
	Berhasil	Tidak
1	V	
2	V	
3	V	
4	V	
5	V	
6	V	
7	V	
8	V	
9	V	
10	V	

B. Pembahasan

Berdasarkan hasil pengujian dari *automatic stoping and infusion monitoring with telemetry system based on android* dapat diperoleh hasil bahwa secara kuantitas alat yang dibuat sudah dapat bekerja dengan baik dan sesuai dengan fungsinya. Pada pengukuran beberapa rangkaian sistem terdapat perbedaan selisih antara hasil pengukuran dengan teori yang ada. Hal ini dimungkinkan disebabkan karena 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

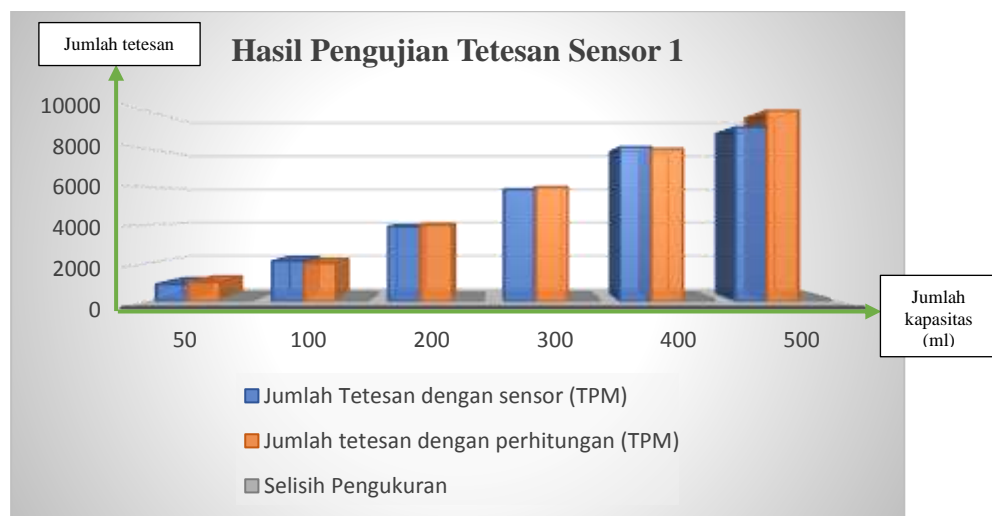
Pengukuran tegangan catu daya menggunakan multimeter digital dilakukan terhadap catu daya yang mempunyai tegangan 12 volt, 9 volt, dan 5 volt. Pada tiap-tiap tegangan tersebut dilakukan pengukuran sebanyak 5 kali menggunakan multimeter digital. Hal ini bertujuan agar memastikan bahwa nantinya alat dapat bekerja dengan baik dalam jangka panjang. Pada pengukuran tegangan 12 volt diperoleh hasil pengukuran sebesar 12.1 volt dengan rata-rata *error* adalah 0.83%. Pada tegangan 9 volt diperoleh hasil pengukuran sebesar 9.27 volt dengan nilai rata-rata *error* sebesar 2.91%. Sedangkan pada rangkaian catu daya 5 volt, rata-rata tegangan yang diukur memiliki nilai sebesar 5.12 volt dengan rata-rata *error* 2.34%.

b. Dengan Beban

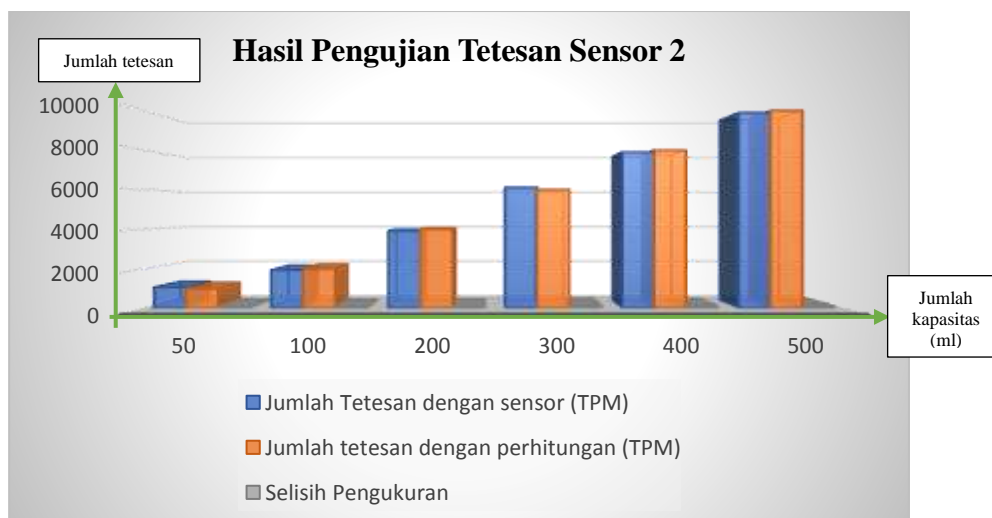
Pada pengukuran tegangan catu daya dengan beban sebanyak lima kali menggunakan multimeter digital diperoleh bahwa pada tegangan 12 volt diperoleh

hasil bahwa rata-rata tegangan sebesar 11,93 volt dengan rata-rata *error* sebesar 0,58%. Berbeda pada rangkaian *step down* pada 9 volt dan 5 volt. Pada rangkaian *step down* 9 volt diperoleh hasil pengukuran dengan beban sebesar 8.97 volt dengan rata-rata *error* 0.33%. Sedangkan pada rangkaian catu daya *switching* 5 volt, rata-rata tegangan yang dihasilkan sebesar 4.87 volt dengan rata-rata *error* sebesar 2.34%.

2. Analisa Pengujian Sensor Tetesan



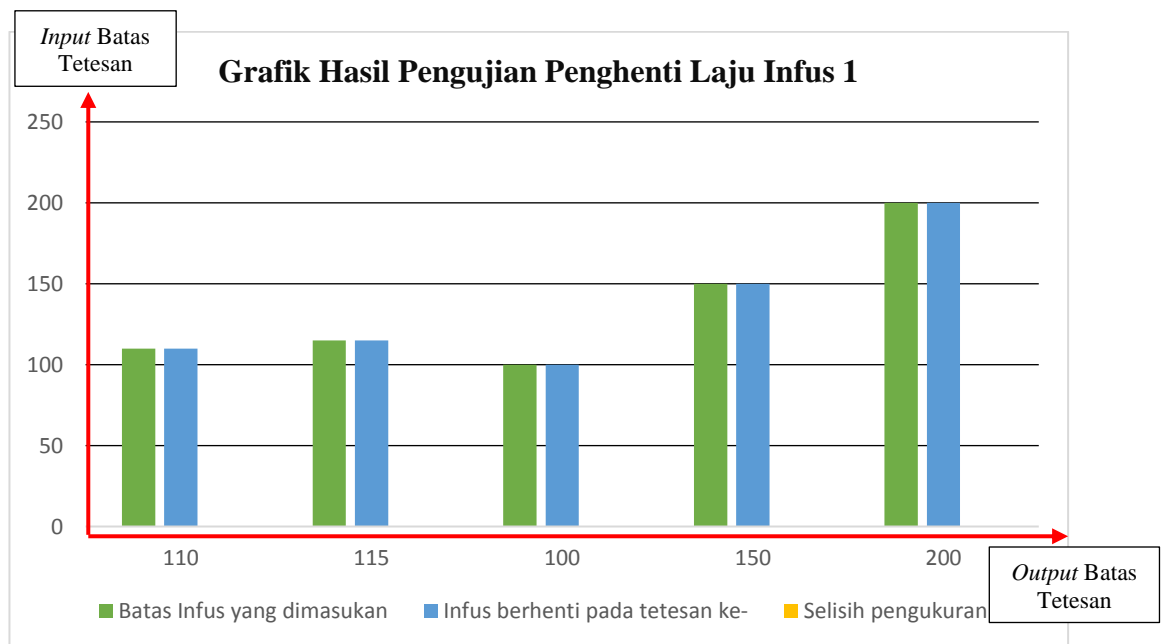
Gambar 41. Hasil Pengujian Tetesan Sensor 1



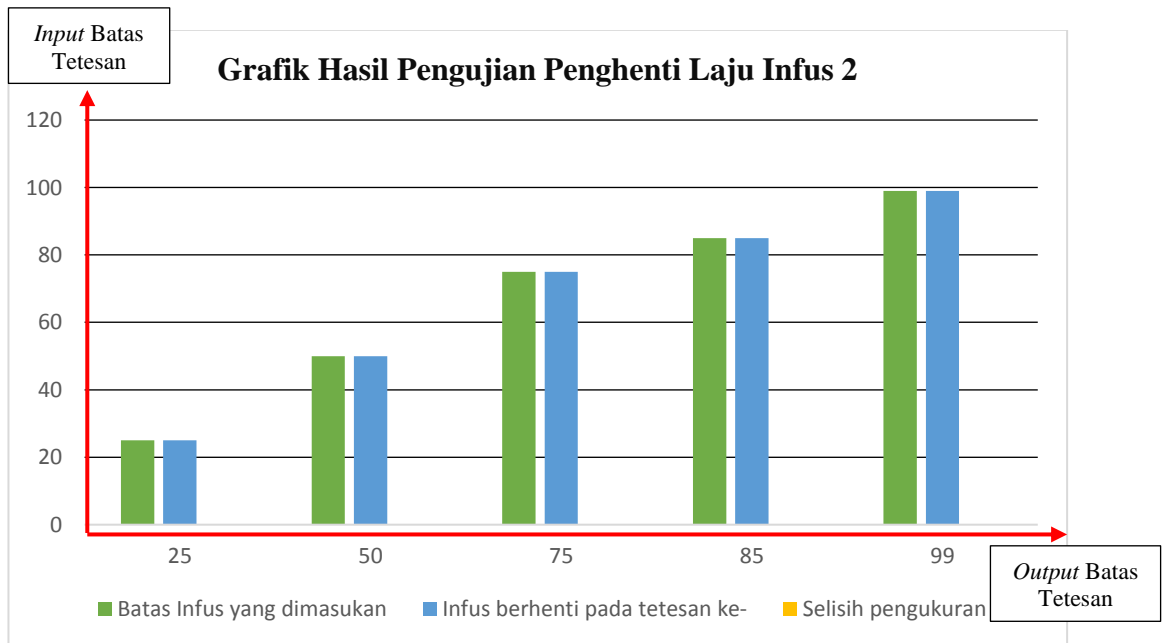
Gambar 42. Hasil Pengujian Tetesan Sensor 2

Dari hasil pengujian diperoleh hasil bahwa pengukuran jumlah tetesan pada sensor 1 diperoleh rata-rata *error* sebesar 3,5%. Sedangkan pada pengukuran jumlah tetesan pada sensor 2 diperoleh rata-rata *error* sebesar 2,52%. Hal ini disebabkan karena tetesan infus yang terlalu banyaknya cahaya yang masuk ke photodiode atau penempatan sensor yang kurang sesuai. Selain itu kecepatan tetesan infus juga dapat mempengaruhi nilai ADC yang dibaca oleh sensor. Semakin cepat tetesan maka semakin tinggi nilai ADC serta semakin cepat waktu pembacaan ADC-nya. Jika tetesan semakin lambat maka perubahan ADC kecil namun waktu untuk pembacaan ADC-nya lambat.

3. Analisa Pengujian Penghenti Laju Infus



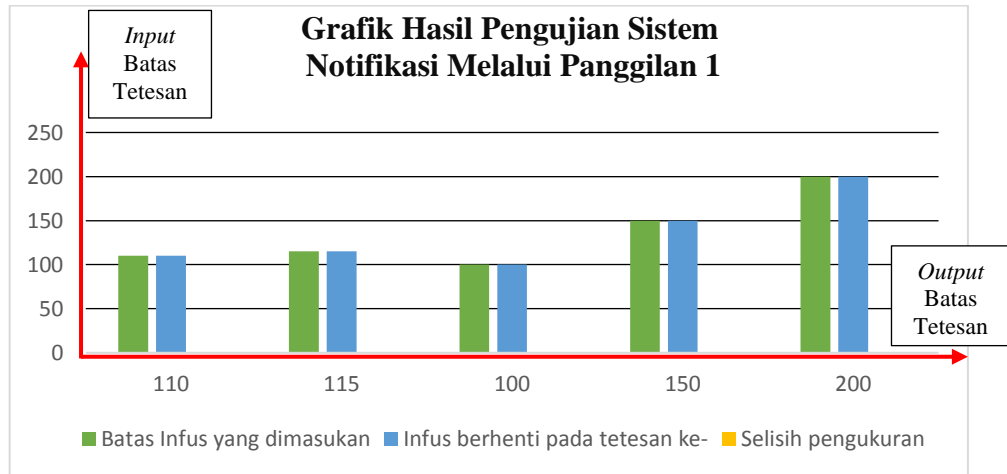
Gambar 43. Hasil Pengujian Penghenti Laju Infus 1



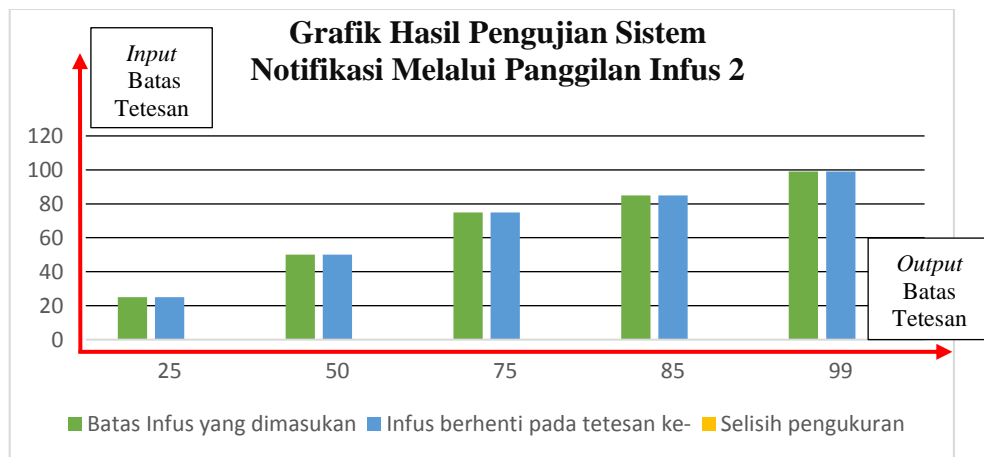
Gambar 44. Hasil Pengujian Penghenti Laju Infus 2

Dari hasil percobaan penghenti infus baik pada infus 1 dan infus 2 diperoleh bahwa hasil pengukuran alat sangat baik. Hal ini dibuktikan dengan pemasukan batas nilai dengan respon motor servo. Ketika masukan diberi batas 100 tetesan maka penghenti infus berhenti pada tetesan yang ke-100. Pengujian ini bukan hanya diberi masukan batas tetesan sebesar 100 namun diberi masukan batas tetes acak dan penghenti infus dapat menghentikan infus pada batas tetesan yang dimasukkan menggunakan *keypad* 4x4. Dari semua pengujian terhadap penghenti infus 1 dan infus 2 tidak terdapat *error*.

4. Analisa Pengujian Sistem Pemberitahuan Melalui Panggilan



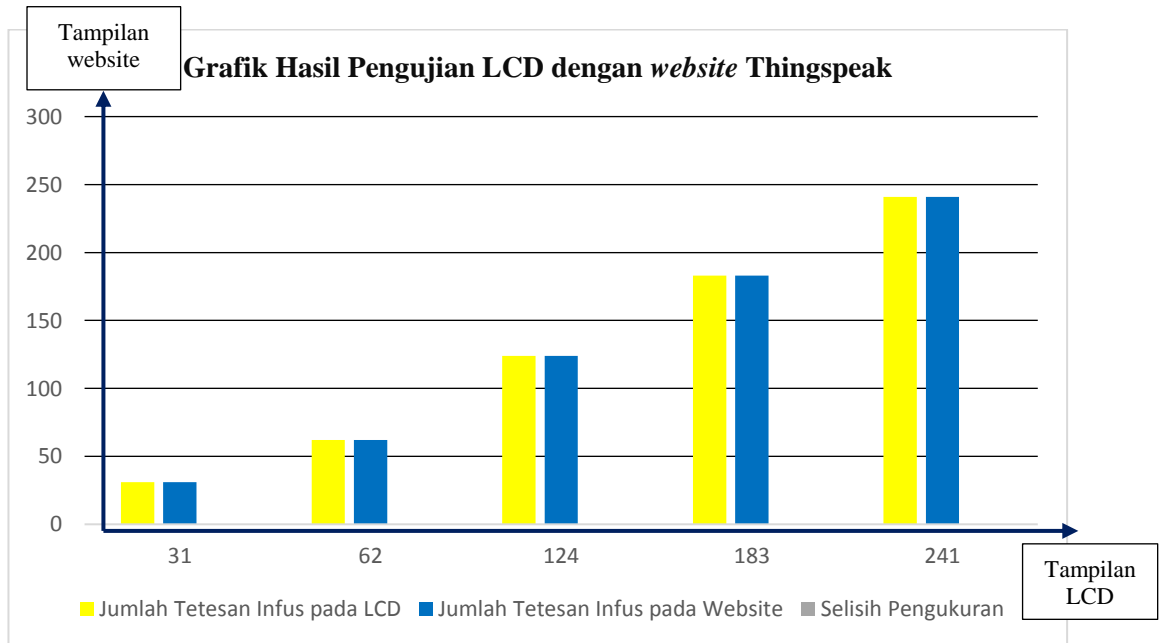
Gambar 45. Hasil Pengujian Sistem Notifikasi Melalui Panggilan 1



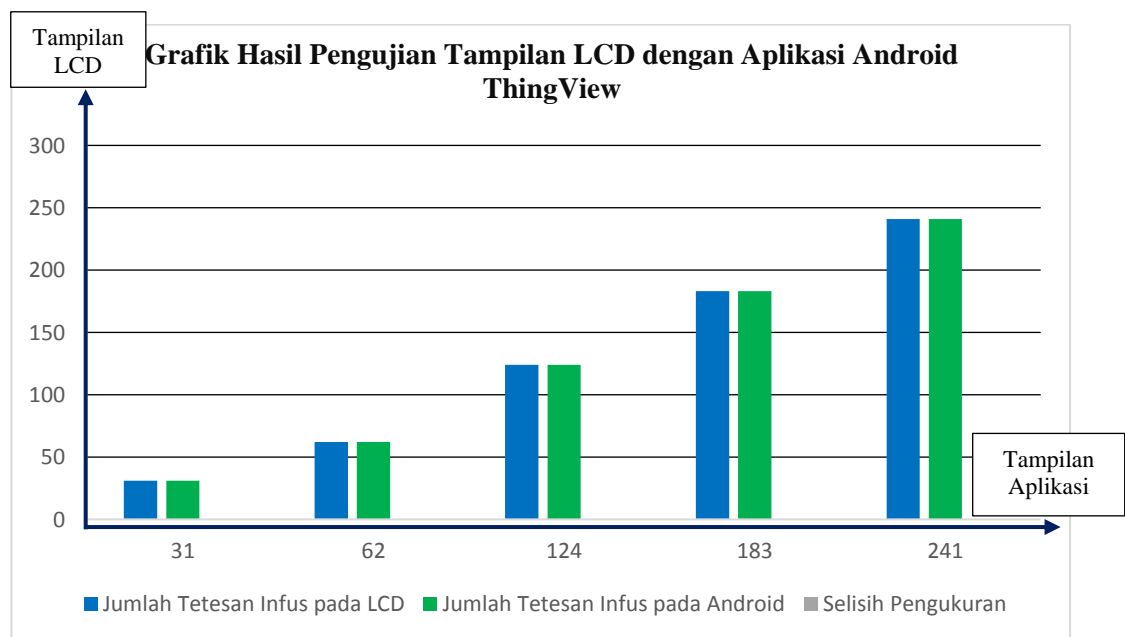
Gambar 46. Hasil Pengujian Sistem Notifikasi Melalui Panggilan Infus 2

Pada hasil percobaan ini data yang dimasukkan adalah sama dengan data penghenti infus 1 dan 2. Karena alat ini melakukan panggilan langsung ketika laju infus terhenti oleh motor servo. Jadi data yang didapat oleh pengujian ini tidak memiliki perbedaan dengan pengujian penghenti infus. Nilai *error* dari sistem pemberitahuan melalui panggilan 1 dan 2 ini adalah sama dengan penghenti infus yaitu 0%.

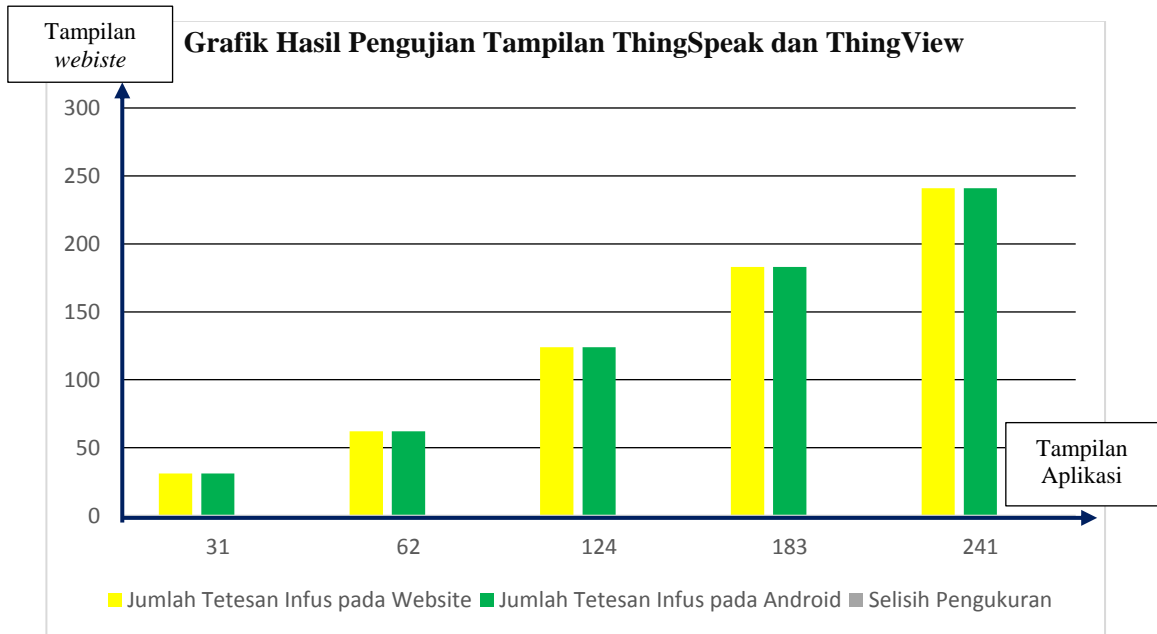
5. Analisa Pengujian Tampilan Tetesan



Gambar 47. Hasil Pengujian LCD dengan ThingSpeak



Gambar 48. Hasil Pengujian Tampilan LCD dengan Thingview



Gambar 49. Hasil Pengujian Tampilan ThingSpeak dengan ThingView

Pada pengujian ini tidak terdapat perbedaan antara tampilan pada LCD dengan ThingSpeak dan ThingView sehingga dapat disimpulkan bahwa tampilan dapat berfungsi dengan baik dengan nilai *error* sebesar 0%. Kendati demikian, proses pengambilan data tetesan pada alat ini dilakukan setiap 15 detik sekali. Selain itu data tetesan yang masuk ke website ataupun aplikasi android hanya data 15 detik yang tampil pada LCD. Hal ini disebabkan karena website dan aplikasi yang digunakan masih bersifat *free account* sehingga fitur-fiturnya masih terbatas.

6. Analisa Pengujian Jangkauan Transmisi Data

Pengujian jarak jangkauan ini berfungsi untuk mengetahui apakah sensor dapat diakses oleh perangkat lain dengan jarak tertentu. Pengujian ini dilakukan dengan cara mengirimkan data dari Wemos D1 Mini untuk ditampilkan hasilnya pada website dan aplikasi android lalu selanjutnya dilakukan uji akses dengan jarak jangkauan tertentu. Parameter jarak jangkauan yang digunakan maksimal sebesar

257 kilometer. Dari pengujian yang telah dilakukan tersebut, dapat disimpulkan bahwa website dapat diakses darimana pun asalkan perangkat masih terhubung dengan koneksi internet. Apabila perangkat tersebut tidak dapat diakses dimungkinkan disebabkan karena perangkat tidak terkoneksi internet sehingga tidak dapat menampilkan data hasil deteksinya.

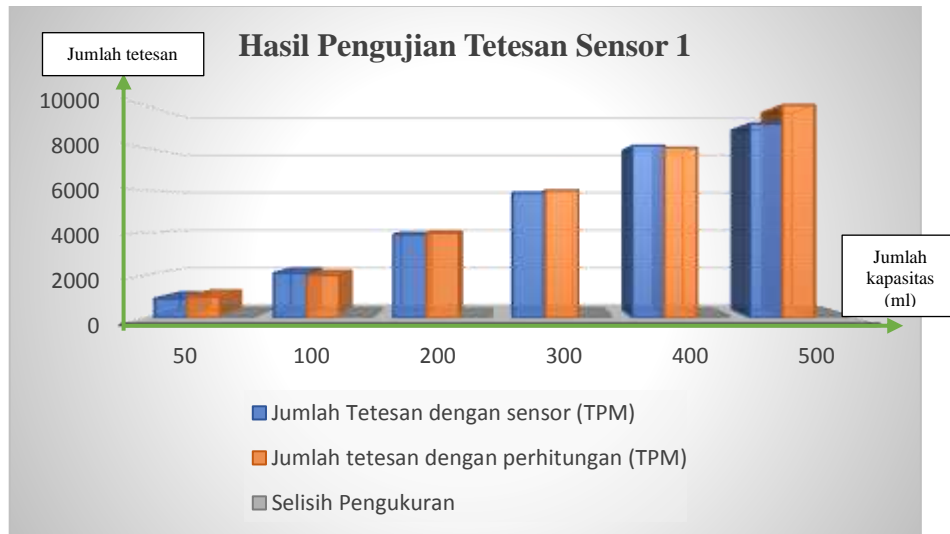
7. Analisa Pengujian LCD

Pada pengujian ini LCD yang digunakan dapat berjalan dengan baik sesuai dengan yang diharapkan. LCD *display* sudah mampu menampilkan karakter-karakter sesuai dengan yang diperintahkan oleh *chip* mikrokontroler Arduino Mega 2560, diantaranya seperti menu tampilan *booting*, tampilan menu pilihan infus, dan menu tampil data tetesan. Pengujian LCD dengan cara membiarkannya dalam kondisi tampil data tetesan selama sekitar 7 jam. Meskipun demikian, LCD tidak mengalami kendala apapun baik peredupan *backlight* ataupun lainnya sehingga diperoleh *error* sebesar 0%.

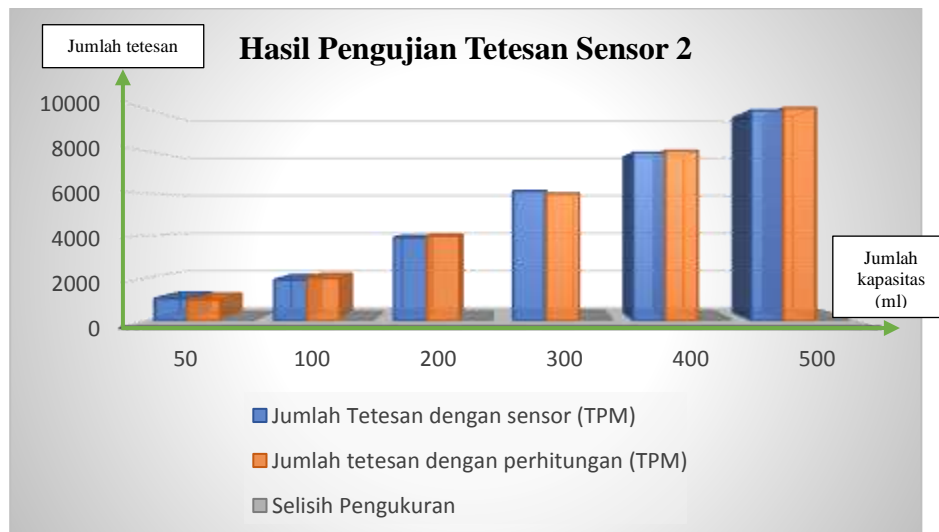
8. Analisa Pengujian Unjuk Kerja

Berdasarkan pengujian unjuk kerja yang dilakukan dengan menggunakan botol infus yang dapat diisi dengan air bersih maka diperoleh data sebagai berikut:

- a. Penggunaan catu daya selama pengujian unjuk kerja alat tidak mengalami kendala ataupun kerusakan sejenisnya.
- b. Penggunaan sensor tetesan infus mengalami kesalahan pembacaan yang cukup kecil.



Gambar 50. Hasil Pengujian Tetesan Pada Sensor 1



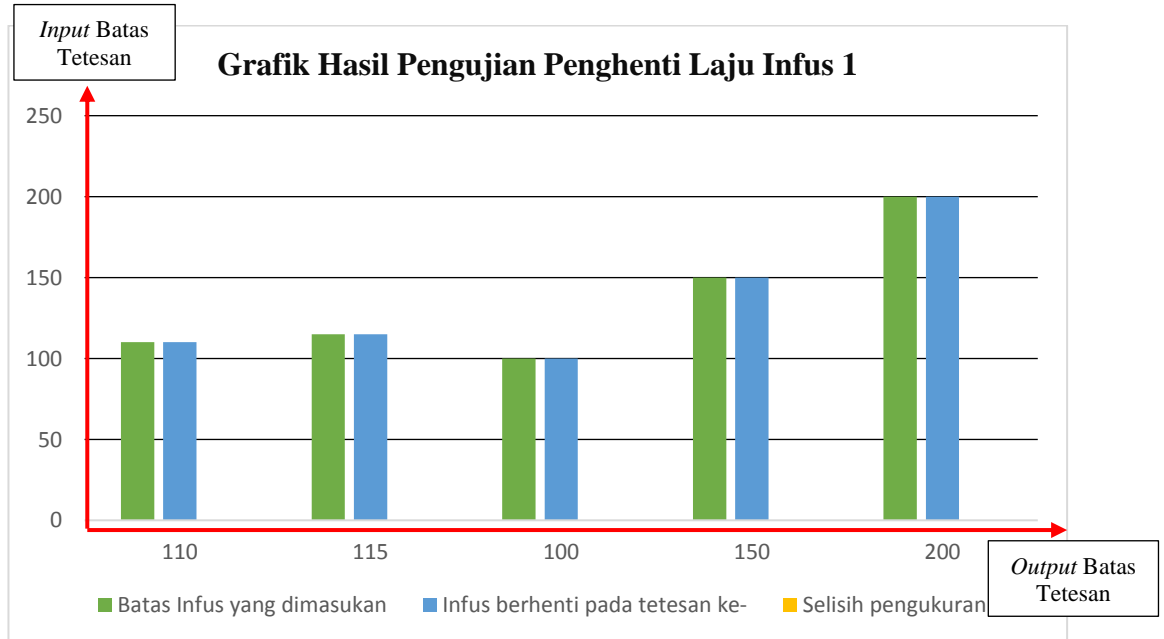
Gambar 51. Hasil Pengujian Tetesan Pada Sensor 2

Dari lima kali pengujian pada masing-masing sensor didapatkan bahwa rata-rata *error* pada sensor 1 sebesar 3,5%, sedangkan pada sensor 2 sebesar 2,52%.

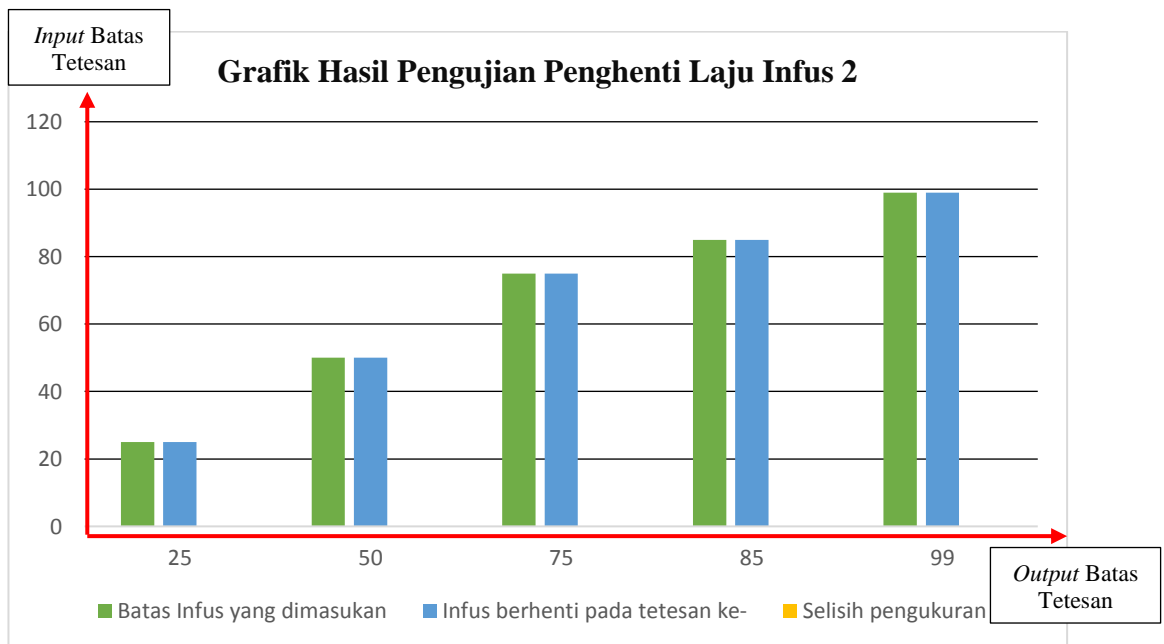
$$error = \frac{10+5,35+1,8+1,7+1,07+1,37}{6} \times 100\% = 3,5\% \quad (\text{Infus 1})$$

$$error = \frac{5,3+3,52+2,06+1,98+1,22+1,09}{6} \times 100\% = 2,52\% \quad (\text{Infus 2})$$

c. Penghentian laju infus dapat bekerja dengan baik



Gambar 52. Hasil Pengujian Penghentian Laju Infus 1



Gambar 53. Hasil Pengujian Penghentian Laju Infus 2

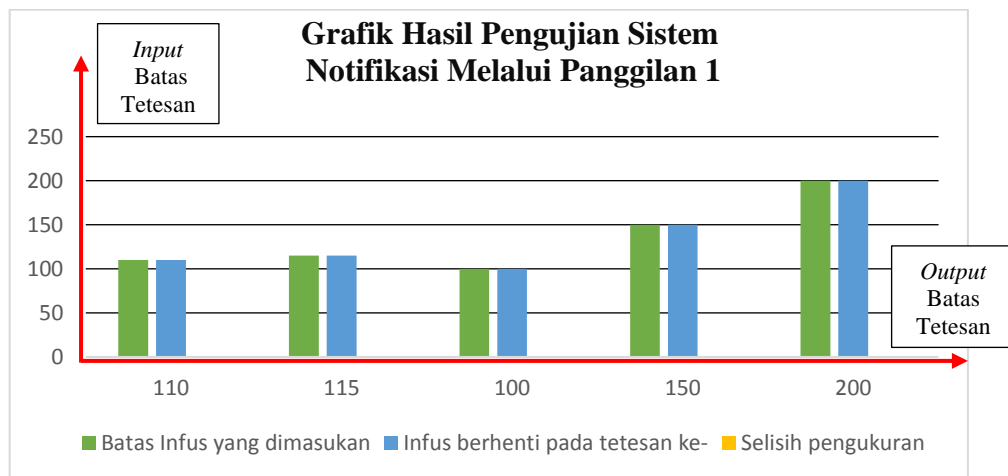
sDari hasil pengujian yang telah dilakukan didapatkan bahwa rata-rata *error* hasil pembacaan sebesar 0%.

$$error = \frac{0+0+0+0+0}{5} \times 100\% = 0\% \text{ (Motor Servo 1)}$$

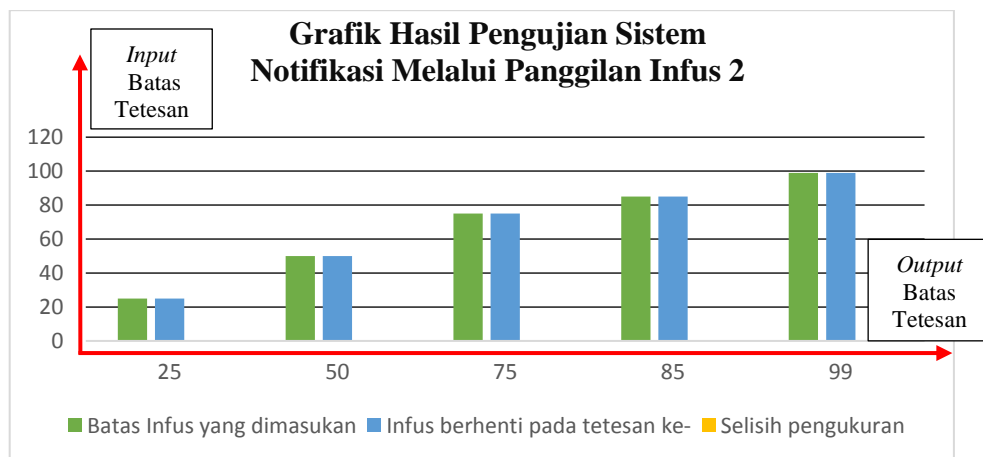
$$error = \frac{0+0+0+0+0}{5} \times 100\% = 0\% \text{ (Motor Servo 2)}$$

Sehingga didapatkan bahwa rata-rata *error* sebesar 0% pada tiap-tiap motor servo. Hal ini disebabkan karena pemilihan motor servo, tegangan dan arus yang diterima sesuai dengan kebutuhan.

d. Sistem notifikasi panggilan dapat bekerja dengan baik



Gambar 54. Hasil Pengujian Sistem Notifikasi Melalui Panggilan Infus 1



Gambar 55. Hasil Pengujian Sistem Notifikasi Melalui Panggilan Infus 2

Pada pengujian ini menggunakan variabel yang sama pada penghenti infus 1 dan 2. Hal ini disebabkan karena alat ini melakukan panggilan langsung ketika laju infus terhenti oleh motor servo. Sehingga data yang didapat pada pengujian ini tidak memiliki perbedaan. Dari enam kali pengujian didapatkan bahwa nilai *error* sebagai berikut:

$$error = \frac{0+0+0+0+0}{5} \times 100\% = 0\% \text{ (Panggilan Infus 1)}$$

$$error = \frac{0+0+0+0+0}{5} \times 100\% = 0\% \text{ (Panggilan Infus 2)}$$

Hal ini disebabkan karena tegangan yang didapat pada modul GSM ini sesuai dengan nilai pada *datasheet* sehingga proses notifikasi dapat lancar tanpa mengalami gangguan.

e. Tampilan Website dan Aplikasi Android dapat bekerja dengan baik

Berdasarkan hasil pengujian sebanyak 10 kali pada website dan aplikasi diperoleh data bahwa website dan aplikasi dapat bekerja sesuai tujuan dan fungsinya sebagai media penampil hasil pengukuran sensor-sensor yang digunakan. Dengan kata lain bahwa hasil pengujian unjuk kerja pada website dan aplikasi memiliki *error* sebesar 0%.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan terhadap *automatic stoping and infusion monitoring with telemetry system based on android*, maka dapat disimpulkan sebagai berikut :

1. *Automatic stoping and infusion monitoring with telemetry system based on android* ini sudah dapat dikembangkan untuk pendeteksi jumlah tetesan infus dan pemberhenti infus dengan baik. Pembuatannya terdiri *hardware* dan *software*. Pada *hardware* digunakan komponen elektronika sensor cahaya photodioda dan laser dioda, *keypad* 4x4, modul WiFi Wemos D1 Mini, motor servo, modul GSM SIM800L V2, serta LCD *display* yang keseluruhan kerjanya dikendalikan oleh Arduino Mega 2560. Pada perancangan dan pembuatan *software* terdapat pembuatan diagram alir kerja dan *source code* program dengan menggunakan bahasa pemrograman C yang dimasukkan pada Arduino Mega 2560 serta ThingSpeak dan ThingView dalam sistem pengaksesan Wemos D1 Mini.
2. Dari hasil pengujian *automatic stoping and infusion monitoring with telemetry system based on android* ini dapat berjalan dengan baik dengan rata-rata *error* pada jumlah tetesan infus 1 sebesar 3,5%, sedangkan pada jumlah tetesan infus 2 sebesar 2,52%. Sementara pada sistem penghenti laju infus 1 dan 2 yang menggunakan motor servo serta sistem notifikasi alat juga didapatkan rata-rata *error* sebesar 0%. Penampilan hasil pengukuran menggunakan LCD *display*,

website, dan aplikasi android juga tidak mengalami kendala. Pada pengujian unjuk kerja ini dilakukan menggunakan lima buah sampel jumlah tetesan yang berbeda untuk diukur pada tiap-tiap sensor. Batas tetesan yang digunakan adalah 5 kali yang nilainya acak mulai dari satuan sampai ribuan. Kesalahan umumnya disebabkan oleh pembacaan ADC sensor yang kurang akurat dan presisi sehingga data yang terbaca tidak sesuai dengan perhitungan yang telah dilakukan.

B. Keterbatasan Alat

Automatic stoping and infusion monitoring with telemetry system based on android ini memiliki keterbatasan dalam sistem kerjanya antara lain :

1. Jumlah objek infus yang dapat dipantau hanya berjumlah 2 buah.
2. Posisi *box controller* kurang ergonomis karena letaknya yang terpisah dengan alat sehingga diperlukan tempat untuk meletakkannya.
3. Ukuran *box controller* yang kurang minimalis.
4. Sistem penampilan data hasil deteksi masih menggunakan ThingSpeak dan ThingView.
5. Membutuhkan roda pada bawah tiang untuk kemudahan pemindahannya.
6. Masih membutuhkan sumber daya PLN sehingga tidak dapat dipindahkan saat alat digunakan.
7. Peletakan sensor yang dilem bakar sehingga rawan terjadinya lepasnya sensor dalam jangka waktu tertentu.

8. Tiang infus yang digunakan belum menggunakan bahan *stainless steel* secara penuh.
9. Tempat sensor berbentuk balok sehingga tidak sesuai dengan *drip chamber* yang bentuknya tabung.
10. Pengecekan pulsa tidak bisa melalui alat secara langsung.
11. Membutuhkan jaringan Wi-Fi dari luar.
12. Belum adanya konektor antara *box* kontroler dengan sensor tetesan dan motor servo.

C. Penelitian Lanjutan

Berdasarkan keterbatasan waktu, kemampuan dan dana sehingga dirasakan masih banyak kekurangan dalam pengerjaan alat yang dibuat ini, maka dari itu penulis menyarankan sebagai berikut:

1. Penambahan modul SD Card pada Arduino Mega 2560 yang digunakan sebagai media penyimpanan data hasil pendeteksian jumlah tetesan.
2. Pengembangan website dinamis sebagai media penampil hasil deteksi.
3. Pembuatan *box controller* yang lebih minimalis.
4. Pembuatan aplikasi android yang lebih spesifik, menarik, dan adanya tampilan infus tinggal berapa persen saat digunakan.
5. Sensor ADC dapat dikembangkan dengan menambah IC komparator supaya sensitivitas dapat diatur melalui potensiometer.
6. Penambahan pengukuran dari dua infus menjadi empat infus supaya port pada arduino dapat digunakan semaksimalnya.

7. Pembuatan tempat sensor berbentuk tabung dan lebih presisi.
8. Penambahan roda pada tiang infus supaya dapat dipindah tempat dengan mudah.
9. Penggunaan baterai Li-Po supaya alat dapat dipindah walau saat dipakai oleh pasien.
10. Penggunaan OLED atau TFT LCD sebagai penampil akan lebih menarik dan efisien.

DAFTAR PUSTAKA

- Ansel, H.C. (1989). *Pengantar Bentuk Sediaan Farmasi edisi ke 4*. Jakarta: Penerbit Universitas Indonesia.
- Arnita, Rina. (2017). *Hubungan Tindakan Perawat pada Pemasangan Infus dalam Mencegah Infeksi Nosokomial “Flebitis” di RSUD dr. R.M Djoelham Binjai*. Medan: Universitas Sumatera Utara.
- Assuncao R., Barbosa P., dkk. (2014). *Developing the Control System of a Syringe Infusion Pump*. Porto-Portugal: Plythechnic of Porto-School of Engineering.
- Barnett, C. C. (2007). *Embedded C Programming and the Atmel AVR 2nd edition*. Canada: Nelson Education, Ltd.
- Cara Harian. (2017). *Cara Menghitung Tetesan Infus Mikro Dan Makro*. Diakses pada tanggal 5 Desember 2017 dari <http://caraharian.com/cara-menghitung-tetes-an-infus.html>
- Depkes. (1995). *Farmakope Indonesia*. Jakarta: Departemen Kesehatan Republik Indonesia.
- Faranux. (2016). *SIM800L V2.0 5V Wireless GSM GPRS MODULE Quad-Band MOD33*. Diakses pada tanggal 11 November 2017 dari <http://www.faranux.com/product/sim800l-v2-0-5v-wireless-gsm-gprs-module-quad-band/>
- Hobby Component. (2014). *Wemos D1 Mini Pro ESP8266 Development Board*. Diakses pada tanggal 11 November 2017 dari <http://hobbycomponents.com/development-boards/864-wemos-d1-mini-pro-esp8266-development-board>
- Hutalaju, David. (2015). *Perancangan Sistem Akuisisi Data dengan Pengukuran Kecepatan Angin, Suhu, dan Kelembapan Udara Jarak Jauh*. Medan: Universitas Sumatera Utara.
- Kata Pena. (2014). *Penyakit-Penyakit yang Membutuhkan Infus*. Diakses pada tanggal 10 November 2017 dari <https://www.katapena.info/2016/10/penyakit-penyakit-yang-membutuhkan-infus.html>

- Keohane, Hayes, dkk. (2005). *Intravenous Medication Safety and Smart Infusion System*. California: Journal of Infusion Nursing, 28, 322-323.
- Kusyati, Eni. (2006). *Ketrampilan dan Prosedur Laboratorium Keperawatan Dasar*. Jakarta: EGC.
- Lachman, Lieberman, dkk. (1994). *Teori dan Praktek Farmasi Industri II*. Jakarta: Penerbit Universitas Indonesia.
- Maxim Integrated. (2011). *Choosing the Right I-Wire® Master for Embedded Applications*. Diakses pada tanggal 11 November 2017 dari <https://www.maximintegrated.com/en/app-notes/index.mvp/id/4206>
- Mazidi, Ali Muhammad. (2011). *The Microcontroller and Embedded System: Using Assembly and C*. New Jersey: Pearson Education, inc.
- Mubarak, Iqbal Mubarak. (2007). *Buku ajar Kebutuhan Dasar Manusia : teori & aplikasi dalam praktik*. Jakarta: EGC.
- Muslim Abdy, Setiawan Iwan, dkk. (2012). *Monitoring cairan Infus Menggunakan Radio Frekuensi YS 1020 UB dengan Frekuensi 433 MHz*. Semarang: Universitas Diponegoro, 7, 1-3.
- Nugroho, Ardiyanto (2014). *Monitoring Tetesan Infus Berbasis Mikrokontroler Atmega16*. Universitas Muhammadiyah Surakarta, Surakarta.
- Potter, Patricia A. (2005). *Buku ajar Fundamental Keperawatan : konsep, proses, dan praktik*. Jakarta: EGC.
- Priyambodo, B. (2007). *Manajemen Farmasi Industri*. Yogyakarta: Global Pustaka Utama.
- Putro, M. R. D. (2014). *Rancang Bangun Sistem Informasi Monitoring Antrian Pada Koperasi Setia Bhakti Wanita Berbasis Web*. Surabaya: STIKOM.
- R. R., Saeed. (2013). *Design and Implementation of a Microcontroller based Infusion Pump System With Bluetooth Wireless Remote Monitoring*. Lefkosa-Cyprus: Near East University.
- R. S., Hadioetomo (1985). *Mikrobiologi Dasar dalam Praktek*. Jakarta: PT. Gramedia.

- S. A. Kokoh Bagus (2009). *Rancang Bangun Alat Pengaturan Jumlah Tetesan Infus pada Pasien dan Monitoring Jarak Jauh dengan PC*. Surabaya: Politeknik Elektronika Negeri Surabaya, 5, 1-3.
- Suyadi. (2012). *Komunikasi Serial dan Port Serial (COM)*. Surakarta: Universitas Muhammadiyah Surakarta.
- Tinta Merah. (2015). *Pengertian Pasien*. Diakses pada tanggal 9 November 2017 dari <https://tintahmerah.wordpress.com/2015/06/23/pengertian-pasien/>
- Zona Elektro. (2013). *Switching Power Supply*. Diakses 12 November 2017 dari <http://zoniaelektro.net/switching-power-supply/>

LAMPIRAN

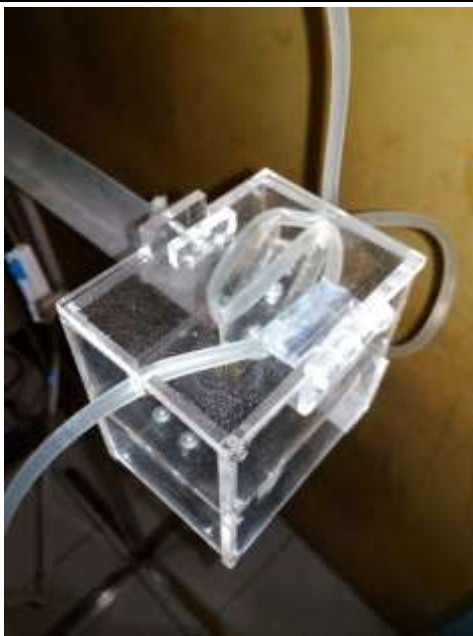
Lampiran 1. Foto Alat



Posisi sensor tetesan dari bawah



Posisi sensor Tetesan dari samping



Posisi selang infus saat ditekuk motor servo



Posisi selang infus sebelum ditekuk motor servo



Antena modul GSM/GPRS pada samping kiri *box controller*



Port USB Arduino Mega 2560 pada samping kanan *box controller*



Posisi *SIM card* pada modul GSM/GPRS menghadap keatas



Pemasukan batas tetesan di LCD dan tampilan diaplikasi Android



Pemasukan nomor telepon di LCD dan tampilan diaplikasi Android



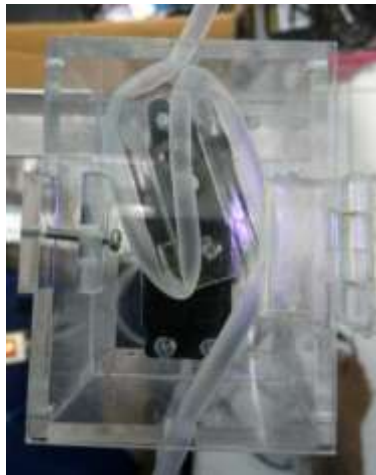
Tampilan jumlah tetesan di LCD dan aplikasi Android



Perubahan nilai tetesan diaplikasi android setiap satu menit



Proses telpon atau *miscall* saat mencapai batas habis atau macet



Posisi selang infus 1 saat dihentikan



Posisi selang infus 2 saat dihentikan



**Pengukuran jumlah tetesan infus 1
dengan gelas ukur**



**Pengukuran jumlah tetesan infus 2
dengan gelas ukur**



Lampiran 2. Program Alat

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
#include <EEPROM.h>
#include <StopWatch.h>
#include <Servo.h>
LiquidCrystal_I2C lcd_mbahe(0x3F ,2,1,0,4,5,6,7,3, POSITIVE);
Servo servo_mbahe1;
Servo servo_mbahe2;
StopWatch sw_millis;
StopWatch sw_micros (StopWatch::MICROS);
StopWatch sw_secs (StopWatch::SECONDS);

int eNo_telp1[13];
int eNo_telp2[13];
int ebatas_tetesan1[10];
int ebatas_tetesan2[10];
int ebatas_tetes11=51;
int ebatas_tetes12=52;
int ebatas_tetes13=53;
int ebatas_tetes21=54;
int ebatas_tetes22=55;
int ebatas_tetes23=56;
int etombol_BT1=57;
int etombol_BT2=58;
int add_eNo_telp1[13];
int add_eNo_telp2[13];
int alamat_eNo_telp1;
int No_telp1[13];
int No_telp2[13];
int batas_tetes1[10];
int batas_tetes2[10];
int No_telp_gabung1[13];
int No_telp_gabung2[13];
int nomer_telp1;
int nomer_telp2;
int read_BT1;
int read_BT2;
int read_etombol_BT1;
int read_etombol_BT2;
int nomer_telp_gabung;
int tombol=0;
int tombol_BT1=0;
int tombol_BT2=0;
int count_hapus=0;
```

```

int count_input_nomer=0;
int menu_infus=1;
int menu_nomerHp_infus1=1;
int menu_nomerHp_infus2=1;
int simpan_nomer[13];
int simpan_batas_tetesan[10];
int count_waktu;
int baris_lcd;
int sensor_tick1 = A0;
int sensor_tick2 = A1;
int led_tick1 = 4;
int led_tick2 = 5;
int sensorValue1=HIGH;
int sensorValue2=HIGH;
int tetesan1=0;
int tetesan2=0;
int tetesan_permenit1=0;
int tetesan_permenit2=0;
const byte baris = 4;
const byte kolom = 4;
char hexaKeys[baris][kolom] =
{{'1','4','7','*'},{'2','5','8','0'},{'3','6','9','#'},{'A','B','C','D'}};
String simpan[13]="";
String simpan_batas_tetes1="";
String simpan_batas_tetes11="";
String simpan_batas_tetes12="";
String simpan_batas_tetes13="";
String simpan_batas_tetes2="";
String simpan_batas_tetes21="";
String simpan_batas_tetes22="";
String simpan_batas_tetes23="";
String path_No_telp1[13]="";
String path_No_telp2[13]="";
String epath_No_telp1="";
String epath_No_telp2="";
String epath_No_telp="";

byte cinta[8] =
{0b00000,0b01010,0b11111,0b11111,0b11111,0b01110,0b00100,0b00000};
byte kursor_atas_bawah[8] =
{0b00100,0b01110,0b11111,0b00000,0b00000,0b00000,0b00000,0b00000};
byte booting[8] =
{0b00000,0b01110,0b11111,0b11111,0b11111,0b11111,0b01110,0b00000};
byte lineup[8] =
{0b11111,0b00000,0b00000,0b00000,0b00000,0b00000,0b00000,0b00000};
byte minim[8]
={0b01110,0b01110,0b01110,0b01110,0b11111,0b01110,0b00100,0b00000};

```

```

byte pin_baris[baris]      = {53, 51,49, 47};
byte pin_kolom[kolom]     = {45, 43, 41, 39};
Keypad customKeypad      = Keypad( makeKeymap(hexaKeys), pin_baris, pin_kolom,
baris, kolom);

int batas_tetesan1=0;
int batas_tetesan2=0;
int batas_tetes_fix1=0;
int batas_tetes_fix2=0;
long int simpan_batas_tetes_fix1;
long int read_batas_tetes_fix1;
int simpan_batas_tetes_fix11;
int simpan_batas_tetes_fix12;
int simpan_batas_tetes_fix13;
long int simpan_batas_tetes_fix2;
long int read_batas_tetes_fix2;
int simpan_batas_tetes_fix21;
int simpan_batas_tetes_fix22;
int simpan_batas_tetes_fix23;
int bool_infus1=0;
int bool_infus2=0;
int bool_kembali=0;
int bool_previous1=0;
int bool_previous2=0;
int bool_previous3=0;
int hari = 0;
int jam = 0;
int menit = 0;
int detik = 0;
float waktu=0;
float teta_waktu=0;
int teta_waktu_infus1;
int teta_waktu_infus2;
int batas_waktu_infus_macet=360;

void setup()
{
  lcd_mbahe.begin (20,4);
  Serial.begin(9600);
  Serial3.begin(9600);
  Serial2.begin(9600);
  lcd_mbahe.createChar(1, cinta);
  lcd_mbahe.createChar(2, kursor_atas_bawah);
  lcd_mbahe.createChar(3, booting);
  lcd_mbahe.createChar(4, lineup);
  lcd_mbahe.createChar(5, minim);
  pinMode(sensor_tick1, INPUT);

```

```

pinMode(sensor_tick2, INPUT);
pinMode(led_tick1, INPUT);
pinMode(led_tick2, INPUT);
servo_mbahe1.attach(35);
servo_mbahe2.attach(37);
servo_mbahe1.write(0);
servo_mbahe2.write(0);
Serial.print("No_telp1 = ");
for(nomer_telp1=1;nomer_telp1<14;nomer_telp1++){
  No_telp1[nomer_telp1]=EEPROM.read(nomer_telp1);
  Serial.print(No_telp1[nomer_telp1]);
  Serial.print(",");
}
Serial.println("");
Serial.print("No_telp2 = ");
for(nomer_telp2=14;nomer_telp2<27;nomer_telp2++){
  No_telp2[nomer_telp2]=EEPROM.read(nomer_telp2);
  Serial.print(No_telp2[nomer_telp2]);
  Serial.print(",");
}
Serial.println("");
Serial.print("Batas Tetesan 1 = ");
for(read_BT1=31;read_BT1<36;read_BT1++){
  batas_tetes1[read_BT1]=EEPROM.read(read_BT1);
  Serial.print(batas_tetes1[read_BT1]);
  Serial.print(",");
}

batas_tetes_fix1=batas_tetes1[31]+batas_tetes1[32]+batas_tetes1[33]+batas_tetes1
[34]+batas_tetes1[35];
Serial.print("\t");
Serial.print(batas_tetes_fix1);
Serial.println("");
Serial.print("Batas Tetesan 2 = ");
for(read_BT2=41;read_BT2<46;read_BT2++){
  batas_tetes2[read_BT2]=EEPROM.read(read_BT2);
  Serial.print(batas_tetes2[read_BT2]);
  Serial.print(",");
}

batas_tetes_fix2=batas_tetes2[41]+batas_tetes2[42]+batas_tetes2[43]+batas_tetes2
[44]+batas_tetes2[45];
Serial.print("\t");
Serial.print(batas_tetes_fix2);
Serial.println("");

read_etombol_BT1=EEPROM.read(etombol_BT1);

```

```

    read_etombol_BT2=EEPROM.read(etombol_BT2);
    Serial.print("Tombol Batas Tetesan 1 = ");
    Serial.print(read_etombol_BT1);
    Serial.println("");
    Serial.print("Tombol Batas Tetesan 2 = ");
    Serial.print(read_etombol_BT2);
    Serial.println("");
    read_batas_tetes_eeprom();
    silahkanTunggu();
}

void loop()
{
// timer_berjalan();
    utama();
}

void utama()
{
    atas:
    while(1)
    {
        char tombole_mbahe_utama = customKeypad.getKey();
        lcd_mbahe.setCursor(4,0);
        lcd_mbahe.print("PILIHAN INFUS  ");
        lcd_mbahe.setCursor(1,1);
        lcd_mbahe.print("Infus 1 cek sensor");
        lcd_mbahe.setCursor(12,2);
        lcd_mbahe.write(5);
        lcd_mbahe.write(5);
        lcd_mbahe.write(5);
        lcd_mbahe.write(5);
        lcd_mbahe.write(5);
        lcd_mbahe.write(5);
        lcd_mbahe.setCursor(1,2);
        lcd_mbahe.print("Infus 2  ");
        lcd_mbahe.setCursor(1,3);
        lcd_mbahe.print("Infus 1&2 tekan D");
        if(tombole_mbahe_utama=='B')
        {
            menu_infus++;
            if(menu_infus>3)menu_infus=1;
        }
        if(tombole_mbahe_utama=='D')
        {
            lcd_mbahe.clear();

```

```

delay(400);
while(1)
{
    tampil_hanya_tetesan();
    char tombol_mbahe_utama = customKeypad.getKey();
    if(tombol_mbahe_utama=='D')
    {
        lcd_mbahe.clear();
        delay(200);
        goto atas;
    }
}
}
if(menu_infus==1)
{
    lcd_mbahe.setCursor(0,1);lcd_mbahe.write(1);
    lcd_mbahe.setCursor(0,2);lcd_mbahe.print(" ");
    lcd_mbahe.setCursor(0,3);lcd_mbahe.print(" ");
    if(tombol_mbahe_utama=='C')
    {
        input_BT1:
        servo_mbahe1.write(0);
        lcd_mbahe.clear();
        delay(400);
        lcd_mbahe.setCursor(0,0);
        lcd_mbahe.print("BATAS TETESAN 1");
        while(1)
        {
            input_batas_tetesan();
            if(bool_kembali==1){while(1){bool_kembali=0;goto atas;}}
            else{while(1){goto lanjut_gan1;}}
        }
        lanjut_gan1:
        while(1)
        {
            Serial.println("Apakah Nomer Diganti???");
            for(nomer_telp1=1;nomer_telp1<14;nomer_telp1++)
            {
                No_telp1[nomer_telp1]=EEPROM.read(nomer_telp1);
                path_No_telp1[nomer_telp1]=(String)No_telp1[nomer_telp1];
                Serial.print(No_telp1[nomer_telp1]);
                Serial.print(",");
            }
            Serial.println("");
        }
        epath_No_telp=path_No_telp1[1]+path_No_telp1[2]+path_No_telp1[3]+path_No_telp1[4

```

```

]+path_No_telp1[5]+path_No_telp1[6]+path_No_telp1[7]+path_No_telp1[8]+path_No_telp1[9]+path_No_telp1[10]+path_No_telp1[11]+path_No_telp1[12];
    lcd_mbahe.setCursor(0,0);
    lcd_mbahe.print("Apakah Nomer Diganti");
    lcd_mbahe.setCursor(0,1);
    lcd_mbahe.print(epath_No_telp);
    char tombole_mbahe_utama = customKeypad.getKey();
    if(tombole_mbahe_utama=='B')
    {
        menu_nomerHp_infus1++;
        if(menu_nomerHp_infus1>2)menu_nomerHp_infus1=1;
    }
    if(tombole_mbahe_utama=='D')
    {
        while(1){tombol_BT1=0;goto input_BT1;}
    }
    if(menu_nomerHp_infus1==1)
    {
        lcd_mbahe.setCursor(0,3);
        lcd_mbahe.print("[YA]                TIDAK");
        if(tombole_mbahe_utama=='C')
        {
            lcd_mbahe.clear();
            delay(200);
            input_no_telp();
            if(bool_previous1==1){while(1){bool_previous1=0;goto
lanjut_gan1;}}
            goto eksekusi1;
        }
    }
    if(menu_nomerHp_infus1==2)
    {
        lcd_mbahe.setCursor(0,3);
        lcd_mbahe.print("YA                [TIDAK]");
        if(tombole_mbahe_utama=='C')
        {
            lcd_mbahe.clear();
            delay(200);
            eksekusi1:
            while(1)
            {
                tampil_tetesan();
                if(tetesan1>read_batas_tetes_fix1)
                {
                    servo_mbahe1.write(180);
                    lcd_mbahe.clear();
                    goto panggil1;
                }
            }
        }
    }
}

```

```

    }
    if(teta_waktu_infus1>batas_waktu_infus_macet){
        servo_mbahe1.write(180);
        lcd_mbahe.clear();
        goto panggil1;
    }
}
panggil1:
while(1)
{
    eksekusi_panggilan();
    tetesan1=0;
    tetesan2=0;
    tetesan_permenit1=0;
    tetesan_permenit2=0;
    tombol_BT1=0;
    tombol_BT2=0;
    goto selesai;
}
}
}
}
}
if(menu_infus==2)
{
    lcd_mbahe.setCursor(0,1);lcd_mbahe.print(" ");
    lcd_mbahe.setCursor(0,2);lcd_mbahe.write(1);
    lcd_mbahe.setCursor(0,3);lcd_mbahe.print(" ");
    if(tombol_mbahe_utama=='C')
    {
        input_BT2:
        servo_mbahe2.write(0);
        lcd_mbahe.clear();
        delay(400);
        lcd_mbahe.setCursor(0,0);
        lcd_mbahe.print("BATAS TETESAN 2");
        while(1)
        {
            input_batas_tetesan();
            if(bool_kembali==1){while(1){bool_kembali=0;goto atas;}}
            else{while(1){goto lanjut_gan2;}}
        }
        lanjut_gan2:
        while(1)
        {
            Serial.println("Apakah Nomer Diganti???");

```

```

for(nomer_telp2=14;nomer_telp2<27;nomer_telp2++)
{
    No_telp2[nomer_telp2]=EEPROM.read(nomer_telp2);
    path_No_telp2[nomer_telp2]=(String)No_telp2[nomer_telp2];
    Serial.print(No_telp2[nomer_telp2]);
    Serial.print(",");
}
Serial.println("");

epath_No_telp=path_No_telp2[14]+path_No_telp2[15]+path_No_telp2[16]+path_No_telp
2[17]+path_No_telp2[18]+path_No_telp2[19]+path_No_telp2[20]+path_No_telp2[21]+pa
th_No_telp2[22]+path_No_telp2[23]+path_No_telp2[24]+path_No_telp2[25];

    lcd_mbahe.setCursor(0,0);
    lcd_mbahe.print("Apakah Nomer Diganti");
    lcd_mbahe.setCursor(0,1);
    lcd_mbahe.print(epath_No_telp);
    char tombole_mbahe_utama = customKeypad.getKey();
    if(tombole_mbahe_utama=='B')
    {
        menu_nomerHp_infus2++;
        if(menu_nomerHp_infus2>2)menu_nomerHp_infus2=1;
    }

    if(tombole_mbahe_utama=='D')
    {
        while(1){tombol_BT2=0;goto input_BT2;}
    }
    if(menu_nomerHp_infus2==1)
    {
        lcd_mbahe.setCursor(0,3);
        lcd_mbahe.print("[YA]          TIDAK");
        if(tombole_mbahe_utama=='C')
        {
            lcd_mbahe.clear();
            delay(200);
            input_no_telp();
            if(bool_previous2==1){while(1){bool_previous2=0;goto
lanjut_gan2;}}
            goto eksekusi2;
        }
    }
    if(menu_nomerHp_infus2==2)
    {
        lcd_mbahe.setCursor(0,3);
        lcd_mbahe.print("YA          [TIDAK]");
        if(tombole_mbahe_utama=='C')
        {

```

```

        lcd_mbahe.clear();
        delay(200);
        eksekusi2:
        while(1)
        {
            tampil_tetesan();
            if(tetesan2>read_batas_tetes_fix2)
            {
                servo_mbahe2.write(180);
                lcd_mbahe.clear();
                goto panggil2;
            }
            if(teta_waktu_infus2>batas_waktu_infus_macet){
                servo_mbahe2.write(180);
                lcd_mbahe.clear();
                goto panggil2;
            }
        }
        panggil2:
        while(1)
        {
            eksekusi_panggilan();
            tetesan1=0;
            tetesan2=0;
            tetesan_permenit1=0;
            tetesan_permenit2=0;
            tombol_BT1=0;
            tombol_BT2=0;
            goto selesai;
        }
    }
}

if(menu_infus==3)
{
    lcd_mbahe.setCursor(0,1);lcd_mbahe.print(" ");
    lcd_mbahe.setCursor(0,2);lcd_mbahe.print(" ");
    lcd_mbahe.setCursor(0,3);lcd_mbahe.write(1);
    if(tombola_mbahe_utama=='C')
    {
        servo_mbahe1.write(0);servo_mbahe2.write(0);
        lcd_mbahe.clear();
        delay(400);
        kembali1:
        while(1)

```

```

    {
        Serial.println("INFUS 1 & INFUS 2");
        Serial.println("Apakah Nomer Diganti???");
        for(nomer_telp_gabung=1;nomer_telp_gabung<14;nomer_telp_gabung++)
        {
            No_telp_gabung1[nomer_telp_gabung]=EEPROM.read(nomer_telp_gabung);
No_telp_gabung2[nomer_telp_gabung]=EEPROM.read(nomer_telp_gabung+13);

path_No_telp1[nomer_telp_gabung]=(String)No_telp_gabung1[nomer_telp_gabung];

path_No_telp2[nomer_telp_gabung]=(String)No_telp_gabung2[nomer_telp_gabung];
            Serial.println(No_telp_gabung1[nomer_telp_gabung]);
            Serial.println(No_telp_gabung2[nomer_telp_gabung]);
            Serial.print(",");
        }
        Serial.println("");

        epath_No_telp1=path_No_telp1[1]+path_No_telp1[2]+path_No_telp1[3]+path_No_telp1[
4]+path_No_telp1[5]+path_No_telp1[6]+path_No_telp1[7]+path_No_telp1[8]+path_No_t
elp1[9]+path_No_telp1[10]+path_No_telp1[11]+path_No_telp1[12];

        epath_No_telp2=path_No_telp2[1]+path_No_telp2[2]+path_No_telp2[3]+path_No_telp2[
4]+path_No_telp2[5]+path_No_telp2[6]+path_No_telp2[7]+path_No_telp2[8]+path_No_t
elp2[9]+path_No_telp2[10]+path_No_telp2[11]+path_No_telp2[12];

        lcd_mbahe.setCursor(0,0);
        lcd_mbahe.print("Apakah Nomer Diganti");
        lcd_mbahe.setCursor(0,1);
        lcd_mbahe.print(epath_No_telp1);
        lcd_mbahe.setCursor(0,2);
        lcd_mbahe.print(epath_No_telp2);
        char tombole_mbahe_utama = customKeypad.getKey();
        if(tombole_mbahe_utama=='B')
        {
            menu_nomerHp_infus2++;
            if(menu_nomerHp_infus2>2)menu_nomerHp_infus2=1;
        }
        if(menu_nomerHp_infus2==1)
        {
            lcd_mbahe.setCursor(0,3);
            lcd_mbahe.print("[YA]          TIDAK");
            if(tombole_mbahe_utama=='C')
            {
                kembali2:
                lcd_mbahe.clear();
                delay(200);
                menu_infus=1;
            }
        }
    }

```

```

        input_no_telp();
        if(bool_previous1==1){while(1){tombol=0;bool_previous1=0;goto
kembali1;}}
        menu_infus=2;
        input_no_telp();
        if(bool_previous2==1){while(1){tombol=0;bool_previous2=0;goto
kembali2;}}
        menu_infus=3;
        goto kembali_eksekusi;
    }
}
if(menu_nomerHp_infus2==2)
{
    lcd_mbahe.setCursor(0,3);
    lcd_mbahe.print("YA          [TIDAK]");
    if(tombole_mbahe_utama=='C')
    {
        lcd_mbahe.clear();
        delay(200);
        kembali_eksekusi:
        while(1)
        {
            tampil_tetesan();
            if(bool_infus1==0)
            {
                if(tetesan1>read_batas_tetes_fix1)
                {
                    tetesan1=read_batas_tetes_fix1;
                    servo_mbahe1.write(180);
                    lcd_mbahe.clear();
                    bool_infus1=1;
                    goto panggil3;
                }
                if(teta_waktu_infus1>batas_waktu_infus_macet){
                    servo_mbahe1.write(180);
                    lcd_mbahe.clear();
                    bool_infus1=1;
                    goto panggil3;
                }
            }
        }
        if(bool_infus2==0)
        {
            if(tetesan2>read_batas_tetes_fix2)
            {
                tetesan2=read_batas_tetes_fix2;
                servo_mbahe2.write(180);
                lcd_mbahe.clear();
            }
        }
    }
}

```

```

        bool_infus2=1;
        goto panggil3;
    }
    if(teta_waktu_infus2>batas_waktu_infus_macet){
        servo_mbahe2.write(180);
        lcd_mbahe.clear();
        bool_infus2=1;
        goto panggil3;
    }
}
}
panggil3:
while(1)
{
    eksekusi_panggilan();
    if(bool_infus1==2&&bool_infus2==2)
    {
        bool_infus1=0;
        bool_infus2=0;
        tetesan1=0;
        tetesan2=0;
        tetesan_permenit1=0;
        tetesan_permenit2=0;
        tombol_BT1=0;
        tombol_BT2=0;
        goto selesai;
    }
    else{goto kembali_eksekusi;}
}
}
}
}
selesai:

while(1){lcd_mbahe.clear();tetesan1=0;tetesan2=0;teta_waktu_infus1=0;teta_waktu_
infus2=0;tetesan_permenit1=0;tetesan_permenit2=0;break;}
}
}
}
}

void panggil(int lama_panggilan)
{
    if(menu_infus==1)
    {
        Serial.print("Nomer Telepon 1:");
        for(nomer_telp1=1;nomer_telp1<14;nomer_telp1++)

```

```

{
    No_telpl[nomer_telpl]=EEPROM.read(nomer_telpl);
    path_No_telpl[nomer_telpl]=(String)No_telpl[nomer_telpl];
    Serial.print(No_telpl[nomer_telpl]);
    Serial.print(",");
}
Serial.println("");

epath_No_telpl=path_No_telpl[1]+path_No_telpl[2]+path_No_telpl[3]+path_No_telpl[4]
]+path_No_telpl[5]+path_No_telpl[6]+path_No_telpl[7]+path_No_telpl[8]+path_No_te
lp1[9]+path_No_telpl[10]+path_No_telpl[11]+path_No_telpl[12];
    Serial.println("Miscall dimulai ke");
    Serial.print("Nomer:");
    Serial.println(epath_No_telpl);
    lcd_mbahe.setCursor(0,0);
    lcd_mbahe.print("Miscall dimulai ke");
    lcd_mbahe.setCursor(0,1);
    lcd_mbahe.print("Nomer:");
    lcd_mbahe.print(epath_No_telpl);
    delay(500);
    String path_calling="ATD"+epath_No_telpl+"\r\n";
    Serial.print(path_calling);
    Serial2.print(path_calling);
    for(int i=0;i<=lama_panggilan;i++)
    {
        for(int a=0;a<20;a++)
        {
            lcd_mbahe.setCursor(a,2);
            lcd_mbahe.write(1);
            delay(50);
        }
        lcd_mbahe.setCursor(0,2);
        lcd_mbahe.print("                ");
    }
    Serial2.print("ATH\r\n");
    lcd_mbahe.clear();
    Serial.println("Miscall Selesai");
    lcd_mbahe.setCursor(0,0);
    lcd_mbahe.print("Miscall Selesai");
}
if(menu_infus==2)
{
    Serial.print("Nomer Telepon 2:");
    for(nomer_telpl2=14;nomer_telpl2<27;nomer_telpl2++)
    {
        No_telpl2[nomer_telpl2]=EEPROM.read(nomer_telpl2);
        path_No_telpl2[nomer_telpl2]=(String)No_telpl2[nomer_telpl2];
    }
}

```

```

        Serial.print(No_telp2[nomer_telp2]);
        Serial.print(",");
    }
    Serial.println("");

    epath_No_telp=path_No_telp2[14]+path_No_telp2[15]+path_No_telp2[16]+path_No_telp
    2[17]+path_No_telp2[18]+path_No_telp2[19]+path_No_telp2[20]+path_No_telp2[21]+pa
    th_No_telp2[22]+path_No_telp2[23]+path_No_telp2[24]+path_No_telp2[25];
    Serial.println("Miscall dimulai ke");
    Serial.print("Nomer:");
    Serial.println(epath_No_telp);
    lcd_mbahe.setCursor(0,0);
    lcd_mbahe.print("Miscall dimulai ke");
    lcd_mbahe.setCursor(0,1);
    lcd_mbahe.print("Nomer:");
    lcd_mbahe.print(epath_No_telp);
    delay(500);
    String path_calling="ATD"+epath_No_telp+"\r\n";
    Serial.print(path_calling);
    Serial2.print(path_calling);
    for(int i=0;i<=lama_panggilan;i++)
    {
        for(int a=0;a<20;a++)
        {
            lcd_mbahe.setCursor(a,2);
            lcd_mbahe.write(1);
            delay(50);
        }
        lcd_mbahe.setCursor(0,2);
        lcd_mbahe.print("                ");
    }
    Serial2.print("ATH\r\n");
    lcd_mbahe.clear();
    Serial.println("Miscall Selesai");
    lcd_mbahe.setCursor(0,0);
    lcd_mbahe.print("Miscall Selesai");
}
if(menu_infus==3)
{
    Serial.println("INFUS 1 & INFUS 2");
    Serial.println("INFUS 1 & INFUS 2");
    Serial.println("Apakah Nomer Diganti??");
    for(nomer_telp_gabung=1;nomer_telp_gabung<14;nomer_telp_gabung++)
    {
        No_telp_gabung1[nomer_telp_gabung]=EEPROM.read(nomer_telp_gabung);
        No_telp_gabung2[nomer_telp_gabung]=EEPROM.read(nomer_telp_gabung+13);
    }
}

```



```

void kirim_data(int data1,int data2)
{
  if(menu_infus==1)
  {
    Serial3.print(123);
    Serial3.print(',');
    Serial3.print(data1);
    Serial3.print(',');
    Serial3.println(1111);
  }
  if(menu_infus==2)
  {
    Serial3.print(123);
    Serial3.print(',');
    Serial3.print(1111);
    Serial3.print(',');
    Serial3.println(data2);
  }
  if(menu_infus==3)
  {
    Serial3.print(123);
    Serial3.print(',');
    Serial3.print(data1);
    Serial3.print(',');
    Serial3.println(data2);
  }
}

void input_no_telp()
{
  while(1)
  {
    if(menu_infus==1)
    {
      lcd_mbahe.setCursor(0,0);
      lcd_mbahe.print("MASUKAN NO.TELEPON 1");
    }
    if(menu_infus==2)
    {
      lcd_mbahe.setCursor(0,0);
      lcd_mbahe.print("MASUKAN NO.TELEPON 2");
    }
    char tombole_mbahe = customKeypad.getKey();
    lcd_mbahe.setCursor(7,3);lcd_mbahe.print("-"
);lcd_mbahe.setCursor(12,3);lcd_mbahe.print("-");

```

```

if(tombol==1){lcd_mbahe.setCursor(3,3);lcd_mbahe.print(simpan_nomer[tombol]);}

if(tombol==2){lcd_mbahe.setCursor(4,3);lcd_mbahe.print(simpan_nomer[tombol]);}

if(tombol==3){lcd_mbahe.setCursor(5,3);lcd_mbahe.print(simpan_nomer[tombol]);}

if(tombol==4){lcd_mbahe.setCursor(6,3);lcd_mbahe.print(simpan_nomer[tombol]);}

if(tombol==5){lcd_mbahe.setCursor(8,3);lcd_mbahe.print(simpan_nomer[tombol]);}

if(tombol==6){lcd_mbahe.setCursor(9,3);lcd_mbahe.print(simpan_nomer[tombol]);}

if(tombol==7){lcd_mbahe.setCursor(10,3);lcd_mbahe.print(simpan_nomer[tombol]);}

if(tombol==8){lcd_mbahe.setCursor(11,3);lcd_mbahe.print(simpan_nomer[tombol]);}

if(tombol==9){lcd_mbahe.setCursor(13,3);lcd_mbahe.print(simpan_nomer[tombol]);}

if(tombol==10){lcd_mbahe.setCursor(14,3);lcd_mbahe.print(simpan_nomer[tombol]);}

if(tombol==11){lcd_mbahe.setCursor(15,3);lcd_mbahe.print(simpan_nomer[tombol]);}

if(tombol==12){lcd_mbahe.setCursor(16,3);lcd_mbahe.print(simpan_nomer[tombol]);}

if(tombole_mbahe=='0' || tombole_mbahe=='1' || tombole_mbahe=='2' || tombole_mbahe=='3'
|| tombole_mbahe=='4' || tombole_mbahe=='5' || tombole_mbahe=='6' || tombole_mbahe=='7'
|| tombole_mbahe=='8' || tombole_mbahe=='9' || tombole_mbahe=='*' || tombole_mbahe=='#
')
{
    if(count_hapus==1 || count_hapus>1)
    {
        Serial.println(tombole_mbahe);
        simpan[tombol]=(String)tombole_mbahe;
        simpan_nomer[tombol]=simpan[tombol].toInt();
        if(menu_infus==1)
        {
            EEPROM.write(eNo_telpl[tombol]=tombol, simpan_nomer[tombol]);
        }
        if(menu_infus==2)
        {
            EEPROM.write(eNo_telpl2[tombol]=tombol+13, simpan_nomer[tombol]);
        }
        lcd_mbahe.setCursor(tombol+3,1);
        lcd_mbahe.print(tombole_mbahe);
        lcd_mbahe.setCursor(tombol+1,2);
        lcd_mbahe.print("  ");
    }
}

```

```

    lcd_mbahe.setCursor(tombol+3,2);
    lcd_mbahe.write(2);
    lcd_mbahe.setCursor(tombol+4,2);
    lcd_mbahe.print(" ");
    count_hapus=0;
}
else
{
    tombol++;
    if(tombol>13)tombol=13;
    Serial.println(tombole_mbahe);
    simpan[tombol]=(String)tombole_mbahe;
    simpan_nomer[tombol]=simpan[tombol].toInt();
    if(menu_infus==1)
    {
        EEPROM.write(eNo_telp1[tombol]=tombol, simpan_nomer[tombol]);
    }
    if(menu_infus==2)
    {
        EEPROM.write(eNo_telp2[tombol]=tombol+13, simpan_nomer[tombol]);
    }
    lcd_mbahe.setCursor(tombol+3,1);
    lcd_mbahe.print(tombole_mbahe);
    lcd_mbahe.setCursor(tombol+1,2);
    lcd_mbahe.print(" ");
    lcd_mbahe.setCursor(tombol+3,2);
    lcd_mbahe.write(2);
    lcd_mbahe.setCursor(tombol+4,2);
    lcd_mbahe.print(" ");
}
}
if(tombole_mbahe=='A')
{
    count_hapus++;
    if(count_hapus==1)
    {
        Serial.println(tombole_mbahe);
        simpan[tombol]=(String)tombole_mbahe;
        simpan_nomer[tombol]=simpan[tombol].toInt();
        if(menu_infus==1)
        {
            EEPROM.write(eNo_telp1[tombol]=tombol, simpan_nomer[tombol]);
        }
        if(menu_infus==2)
        {
            EEPROM.write(eNo_telp2[tombol]=tombol+13, simpan_nomer[tombol]);
        }
    }
}

```

```

    simpan_nomer[tombol]=0;
    lcd_mbahe.setCursor(tombol+3,1);
    lcd_mbahe.print(" ");
    lcd_mbahe.setCursor(tombol,2);
    lcd_mbahe.print(" ");
    lcd_mbahe.setCursor(tombol+2,2);
    lcd_mbahe.write(2);
    lcd_mbahe.setCursor(tombol+3,2);
    lcd_mbahe.print(" ");
}
if(count_hapus>1)
{
    tombol--;
    Serial.println(tombol_mbahe);
    simpan[tombol]=(String)tombol_mbahe;
    simpan_nomer[tombol]=simpan[tombol].toInt();
    if(menu_infus==1)
    {
        EEPROM.write(eNo_telp1[tombol]=tombol, simpan_nomer[tombol]);
    }
    if(menu_infus==2)
    {
        EEPROM.write(eNo_telp2[tombol]=tombol+13, simpan_nomer[tombol]);
    }
    lcd_mbahe.setCursor(tombol+3,1);
    lcd_mbahe.print(" ");
    lcd_mbahe.setCursor(tombol,2);
    lcd_mbahe.print(" ");
    lcd_mbahe.setCursor(tombol+2,2);
    lcd_mbahe.write(2);
    lcd_mbahe.setCursor(tombol+3,2);
    lcd_mbahe.print(" ");
    if(tombol==1)
    {
        count_hapus=0;
        tombol=0;
    }
}
}
if(tombol_mbahe=='D')
{
    lcd_mbahe.clear();
    delay(400);
    if(menu_infus==1)
    {
        bool_previous1=1;
    }
}

```

```

        if(menu_infus==2)
        {
            bool_previous2=1;
        }
        break;
    }
    if(tombole_mbahe=='C')
    {
        lcd_mbahe.clear();
        tombol=0;
        delay(200);
        goto selesai;
    }
}
selesai:
while(1){lcd_mbahe.clear();break;}
}

void eksekusi_panggilan()
{
    lcd_mbahe.clear();
    delay(200);
    baris_lcd=0;
    while(1)
    {
        panggil(20); // Satuan Detik
        goto selesai;
    }
    selesai:
    while(1)
    {
        tombol=0;
        lcd_mbahe.clear();
        break;
    }
}

void sensor_tetesan()
{
    sensorValue1 = analogRead(sensor_tick1);
    int inf1 =map(sensorValue1,0,1023,0,255);
    sensorValue2 = analogRead(sensor_tick2);
    int inf2 =map(sensorValue2,0,1023,0,255);
    if(inf1>=20)
    {
        digitalWrite(led_tick1, HIGH);
        tetesan1++;
    }
}

```

```

Serial.print("Sensor 1 Netes: ");
Serial.print(tetesan1);Serial.println(">>\t");Serial.print(infl);
Serial.print("\t");
teta_waktu_infus1=0;
}
else{digitalWrite(led_tick1, LOW);Serial.print("Sensor 1 Ora Netes");
teta_waktu_infus1++;

Serial.print(teta_waktu_infus1);Serial.print(">>\t");Serial.print(infl);Serial.p
rint("\t");
}
if(inf2>=50)
{
digitalWrite(led_tick2, HIGH);
tetesan2++;
Serial.print("Sensor 2 Netes: ");
Serial.print(tetesan2);Serial.print(">>\t");Serial.println(inf2);
teta_waktu_infus2=0;
}
else{digitalWrite(led_tick2, LOW);Serial.print("Sensor 2 Ora Netes");
teta_waktu_infus2++;
Serial.print(teta_waktu_infus2);Serial.print(">>\t");Serial.println(inf2);
}
delay(0);
}
void tampil_tetesan()
{
sensor_tetesan();
read_batas_tetes_eeeprom();
timer_berjalan();
kirim_data(tetesan1,tetesan2);
lcd_mbahe.setCursor(0,0);
lcd_mbahe.print("Jumlah Tetesan/menit");
if(menu_infus==1)
{
lcd_mbahe.setCursor(0,1);
lcd_mbahe.print("I1:");
lcd_mbahe.print(tetesan1);
lcd_mbahe.print(">");
lcd_mbahe.print("(");
lcd_mbahe.print(read_batas_tetes_fix1);
lcd_mbahe.print(")");
lcd_mbahe.print("(");
if(detik==0){tetesan_permenit1=tetesan1;}
lcd_mbahe.print(tetesan_permenit1);
lcd_mbahe.print("t/m");
lcd_mbahe.setCursor(0,2);

```

```

    lcd_mbahe.print("I2:OFF");
}
if(menu_infus==2)
{
    lcd_mbahe.setCursor(0,1);
    lcd_mbahe.print("I1:OFF");
    lcd_mbahe.setCursor(0,2);
    lcd_mbahe.print("I2:");
    lcd_mbahe.print(tetesan2);
    lcd_mbahe.print(">");
    lcd_mbahe.print("(");
    lcd_mbahe.print(read_batas_tetes_fix2);
    lcd_mbahe.print(")");
    lcd_mbahe.print("(");
    if(detik==0){tetesan_permenit2=tetesan2;}
    lcd_mbahe.print(tetesan_permenit2);
    lcd_mbahe.print("t/m");
}
if(menu_infus==3)
{
    if(detik==0){tetesan_permenit1=tetesan1;tetesan_permenit2=tetesan2;}
    lcd_mbahe.setCursor(0,1);
    lcd_mbahe.print("I1:");
    lcd_mbahe.print(tetesan1);
    lcd_mbahe.print(">");
    lcd_mbahe.print("(");
    lcd_mbahe.print(read_batas_tetes_fix1);
    lcd_mbahe.print(")");
    lcd_mbahe.print("(");
    lcd_mbahe.print(tetesan_permenit1);
    lcd_mbahe.print("t/m");
    lcd_mbahe.setCursor(0,2);
    lcd_mbahe.print("I2:");
    lcd_mbahe.print(tetesan2);
    lcd_mbahe.print(">");
    lcd_mbahe.print("(");
    lcd_mbahe.print(read_batas_tetes_fix2);
    lcd_mbahe.print(")");
    lcd_mbahe.print("(");
    lcd_mbahe.print(tetesan_permenit2);
    lcd_mbahe.print("t/m");
}
}

void tampil_hanya_tetesan()
{
    sensor_tetesan();
}

```

```

timer_berjalan();
kirim_data(tetesan1,tetesan2);
lcd_mbahe.setCursor(0,0);
lcd_mbahe.print("Jumlah Tetesan/menit");
lcd_mbahe.setCursor(0,1);
lcd_mbahe.print("Infus 1 :");
lcd_mbahe.print(tetesan1);
lcd_mbahe.setCursor(0,2);
lcd_mbahe.print("Infus 2 :");
lcd_mbahe.print(tetesan2);
}

void bersih2_layar()
{
    lcd_mbahe.setCursor(0,0);
    lcd_mbahe.print("                ");
    lcd_mbahe.setCursor(0,1);
    lcd_mbahe.print("                ");
    lcd_mbahe.setCursor(0,2);
    lcd_mbahe.print("                ");
    lcd_mbahe.setCursor(0,3);
    lcd_mbahe.print("                ");
    delay(100);
}

void silahkanTunggu()
{
    int i;
    for (i = 0 ; i < 1; i ++ )
    {
        bersih2_layar();
        lcd_mbahe.setCursor(0,0);
        lcd_mbahe.print("_____");
        lcd_mbahe.setCursor(0,1);
        lcd_mbahe.print(" *^Smart Infusion^* ");
        lcd_mbahe.setCursor(0,2);
        lcd_mbahe.print("Moh Sirojul Aziis");
        for(int a=0;a<20;a++)
        {
            lcd_mbahe.setCursor(a,2);
            lcd_mbahe.write(4);
        }
        for(int b=0;b<20;b++)
        {
            lcd_mbahe.setCursor(b,3);
            lcd_mbahe.write(3);
            delay(10);
        }
    }
}

```



```

Serial.print(simpan_batas_tetes_fix23);
Serial.print("\t\t\t");
Serial.print(simpan_batas_tetes2);
Serial.print("\t>>>\t");
Serial.println(simpan_batas_tetes_fix2);
EEPROM.write(ebatas_tetes21, simpan_batas_tetes_fix21);
EEPROM.write(ebatas_tetes22, simpan_batas_tetes_fix22);
EEPROM.write(ebatas_tetes23, simpan_batas_tetes_fix23);
}

if(tombale_mbahe=='0' || tombale_mbahe=='1' || tombale_mbahe=='2' || tombale_mbahe=='3'
|| tombale_mbahe=='4' || tombale_mbahe=='5' || tombale_mbahe=='6' || tombale_mbahe=='7'
|| tombale_mbahe=='8' || tombale_mbahe=='9' || tombale_mbahe=='*' || tombale_mbahe=='#
')
{
  if(count_hapus==1 || count_hapus>1)
  {
    if(menu_infus==1)
    {
      Serial.println(tombale_mbahe);
      simpan[tombol_BT1]=(String)tombale_mbahe;
      simpan_batas_tetesan[tombol_BT1]=simpan[tombol_BT1].toInt();
      EEPROM.write(ebatas_tetesan1[tombol_BT1]=tombol_BT1+30,
simpan_batas_tetesan[tombol_BT1]);
      EEPROM.write(etombol_BT1, tombol_BT1);
      lcd_mbahe.setCursor(tombol_BT1+7,1);
      lcd_mbahe.print(tombale_mbahe);
      lcd_mbahe.setCursor(tombol_BT1+5,2);
      lcd_mbahe.print(" ");
      lcd_mbahe.setCursor(tombol_BT1+7,2);
      lcd_mbahe.write(2);
      lcd_mbahe.setCursor(tombol_BT1+8,2);
      lcd_mbahe.print(" ");
      count_hapus=0;
    }
    if(menu_infus==2)
    {
      Serial.println(tombale_mbahe);
      simpan[tombol_BT2]=(String)tombale_mbahe;
      simpan_batas_tetesan[tombol_BT2]=simpan[tombol_BT2].toInt();
      EEPROM.write(ebatas_tetesan2[tombol_BT2]=tombol_BT2+40,
simpan_batas_tetesan[tombol_BT2]);
      EEPROM.write(etombol_BT2, tombol_BT2);
      lcd_mbahe.setCursor(tombol_BT2+7,1);
      lcd_mbahe.print(tombale_mbahe);
      lcd_mbahe.setCursor(tombol_BT2+5,2);
      lcd_mbahe.print(" ");
    }
  }
}

```

```

        lcd_mbahe.setCursor(tombol_BT2+7,2);
        lcd_mbahe.write(2);
        lcd_mbahe.setCursor(tombol_BT2+8,2);
        lcd_mbahe.print(" ");
        count_hapus=0;
    }
}
else
{
    if(menu_infus==1)
    {
        tombol_BT1++;
        if(tombol_BT1>5)tombol_BT1=5;
        Serial.println(tombole_mbahe);
        simpan[tombol_BT1]=(String)tombole_mbahe;
        simpan_batas_tetesan[tombol_BT1]=simpan[tombol_BT1].toInt();
        EEPROM.write(ebatas_tetesan1[tombol_BT1]=tombol_BT1+30,
simpan_batas_tetesan[tombol_BT1]);
        EEPROM.write(etombol_BT1, tombol_BT1);
        lcd_mbahe.setCursor(tombol_BT1+7,1);
        lcd_mbahe.print(tombole_mbahe);
        lcd_mbahe.setCursor(tombol_BT1+5,2);
        lcd_mbahe.print(" ");
        lcd_mbahe.setCursor(tombol_BT1+7,2);
        lcd_mbahe.write(2);
        lcd_mbahe.setCursor(tombol_BT1+8,2);
        lcd_mbahe.print(" ");
    }
    if(menu_infus==2)
    {
        tombol_BT2++;
        if(tombol_BT2>5)tombol_BT2=5;
        Serial.println(tombole_mbahe);
        simpan[tombol_BT2]=(String)tombole_mbahe;
        simpan_batas_tetesan[tombol_BT2]=simpan[tombol_BT2].toInt();
        EEPROM.write(ebatas_tetesan2[tombol_BT2]=tombol_BT2+40,
simpan_batas_tetesan[tombol_BT2]);
        EEPROM.write(etombol_BT2, tombol_BT2);
        lcd_mbahe.setCursor(tombol_BT2+7,1);
        lcd_mbahe.print(tombole_mbahe);
        lcd_mbahe.setCursor(tombol_BT2+5,2);
        lcd_mbahe.print(" ");
        lcd_mbahe.setCursor(tombol_BT2+7,2);
        lcd_mbahe.write(2);
        lcd_mbahe.setCursor(tombol_BT2+8,2);
        lcd_mbahe.print(" ");
    }
}

```

```

    }
}
if(tombale_mbahe=='A')
{
    count_hapus++;
    if(count_hapus==1)
    {
        if(menu_infus==1)
        {
            Serial.println(tombale_mbahe);
            simpan[tombol_BT1]=(String)tombale_mbahe;
            simpan_batas_tetesan[tombol_BT1]=simpan[tombol_BT1].toInt();
            EEPROM.write(ebatas_tetesan1[tombol_BT1]=tombol_BT1+30,
simpan_batas_tetesan[tombol_BT1]);
            EEPROM.write(etombol_BT1, tombol_BT1);
            simpan_batas_tetesan[tombol_BT1]=0;
            lcd_mbahe.setCursor(tombol_BT1+7,1);
            lcd_mbahe.print(" ");
            lcd_mbahe.setCursor(tombol_BT1,2);
            lcd_mbahe.print(" ");
            lcd_mbahe.setCursor(tombol_BT1+6,2);
            lcd_mbahe.write(2);
            lcd_mbahe.setCursor(tombol_BT1+7,2);
            lcd_mbahe.print(" ");
        }
        if(menu_infus==2)
        {
            Serial.println(tombale_mbahe);
            simpan[tombol_BT2]=(String)tombale_mbahe;
            simpan_batas_tetesan[tombol_BT2]=simpan[tombol_BT2].toInt();
            EEPROM.write(ebatas_tetesan2[tombol_BT2]=tombol_BT2+40,
simpan_batas_tetesan[tombol_BT2]);
            EEPROM.write(etombol_BT2, tombol_BT2);
            simpan_batas_tetesan[tombol_BT2]=0;
            lcd_mbahe.setCursor(tombol_BT2+7,1);
            lcd_mbahe.print(" ");
            lcd_mbahe.setCursor(tombol_BT2,2);
            lcd_mbahe.print(" ");
            lcd_mbahe.setCursor(tombol_BT2+6,2);
            lcd_mbahe.write(2);
            lcd_mbahe.setCursor(tombol_BT2+7,2);
            lcd_mbahe.print(" ");
        }
    }
}
if(count_hapus>1)
{
    if(menu_infus==1)

```

```

    {
        tombol_BT1--;
        Serial.println(tombale_mbahe);
        simpan[tombol_BT1]=(String)tombale_mbahe;
        simpan_batas_tetesan[tombol_BT1]=simpan[tombol_BT1].toInt();
        EEPROM.write(ebatas_tetesan1[tombol_BT1]=tombol_BT1+30,
simpan_batas_tetesan[tombol_BT1]);
        EEPROM.write(etombol_BT1, tombol_BT1);
        lcd_mbahe.setCursor(tombol_BT1+7,1);
        lcd_mbahe.print(" ");
        lcd_mbahe.setCursor(tombol_BT1,2);
        lcd_mbahe.print(" ");
        lcd_mbahe.setCursor(tombol_BT1+6,2);
        lcd_mbahe.write(2);
        lcd_mbahe.setCursor(tombol_BT1+7,2);
        lcd_mbahe.print(" ");
        if(tombol_BT1==1)
        {
            count_hapus=0;
            tombol_BT1=0;
        }
    }
    if(menu_infus==2)
    {
        tombol_BT2--;
        Serial.println(tombale_mbahe);
        simpan[tombol_BT2]=(String)tombale_mbahe;
        simpan_batas_tetesan[tombol_BT2]=simpan[tombol_BT2].toInt();
        EEPROM.write(ebatas_tetesan2[tombol_BT2]=tombol_BT2+40,
simpan_batas_tetesan[tombol_BT2]);
        EEPROM.write(etombol_BT2, tombol_BT2);
        lcd_mbahe.setCursor(tombol_BT2+7,1);
        lcd_mbahe.print(" ");
        lcd_mbahe.setCursor(tombol_BT2,2);
        lcd_mbahe.print(" ");
        lcd_mbahe.setCursor(tombol_BT2+6,2);
        lcd_mbahe.write(2);
        lcd_mbahe.setCursor(tombol_BT2+7,2);
        lcd_mbahe.print(" ");
        if(tombol_BT2==1)
        {
            count_hapus=0;
            tombol_BT2=0;
        }
    }
}
}
}

```



```

        simpan_batas_tetes21=(String)satu;
        simpan_batas_tetes22=(String)dua;
        simpan_batas_tetes23=(String)tiga;

simpan_batas_tetes2=simpan_batas_tetes21+simpan_batas_tetes22+simpan_batas_tetes
23;

        read_batas_tetes_fix2=simpan_batas_tetes2.toInt();
        Serial.print(satu);
        Serial.print("\t>>>\t");
        Serial.print(simpan_batas_tetes21);
        Serial.print("\t\t");
        Serial.print(dua);
        Serial.print("\t>>>\t");
        Serial.print(simpan_batas_tetes22);
        Serial.print("\t\t");
        Serial.print(tiga);
        Serial.print("\t>>>\t");
        Serial.print(simpan_batas_tetes23);
        Serial.print("\t\t\t");
        Serial.print(simpan_batas_tetes2);
        Serial.print("\t>>>\t");
        Serial.println(read_batas_tetes_fix2);
        Serial.println("");
    }
    break;
}
}

void timer_micros()
{
    waktu=sw_micros.elapsed();
    teta_waktu=waktu/1000000;
//    teta_waktu_infus1=waktu/1000000;
//    teta_waktu_infus2=waktu/1000000;
}

void timer_berjalan()
{
    timer_micros();
    sw_millis.start();
    sw_micros.start();
    sw_secs.start();
    if(teta_waktu>1)
    {
        detik++;
        sw_millis.reset();
        sw_micros.reset();
    }
}

```

```

        sw_secs.reset();
    }
    if(detik>59)
    {
        detik=0;
        menit=menit+1;
        if(menit>59)
        {
            menit=0;
            jam=jam+1;
            if(jam>23)
            {
                jam=0;
                hari=hari+1;
                if(menit>59)
                {
                    menit=0;
                }
            }
        }
    }
}

if(hari<10){lcd_mbahe.setCursor(5,3);lcd_mbahe.print("0");lcd_mbahe.print(hari);
}
else if(hari>=10){lcd_mbahe.setCursor(5,3);lcd_mbahe.print(hari);}
lcd_mbahe.setCursor(7,3);
lcd_mbahe.print(":");

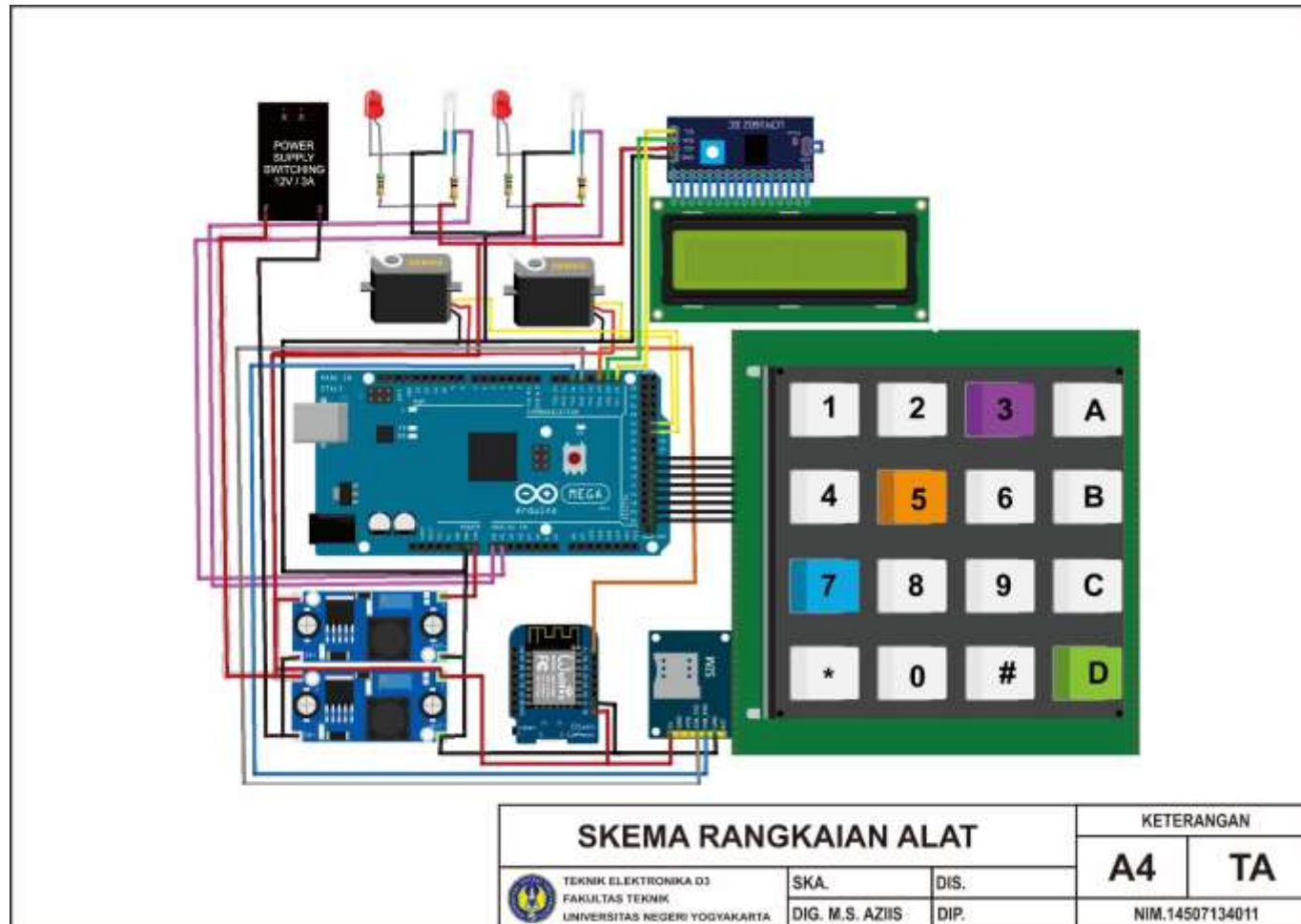
if(jam<10){lcd_mbahe.setCursor(8,3);lcd_mbahe.print("0");lcd_mbahe.print(jam);}
else if(jam>=10){lcd_mbahe.setCursor(8,3);lcd_mbahe.print(jam);}
lcd_mbahe.setCursor(10,3);
lcd_mbahe.print(":");

if(menit<10){lcd_mbahe.setCursor(11,3);lcd_mbahe.print("0");lcd_mbahe.print(menit);}
else if(menit>=10){lcd_mbahe.setCursor(11,3);lcd_mbahe.print(menit);}
lcd_mbahe.setCursor(13,3);
lcd_mbahe.print(":");

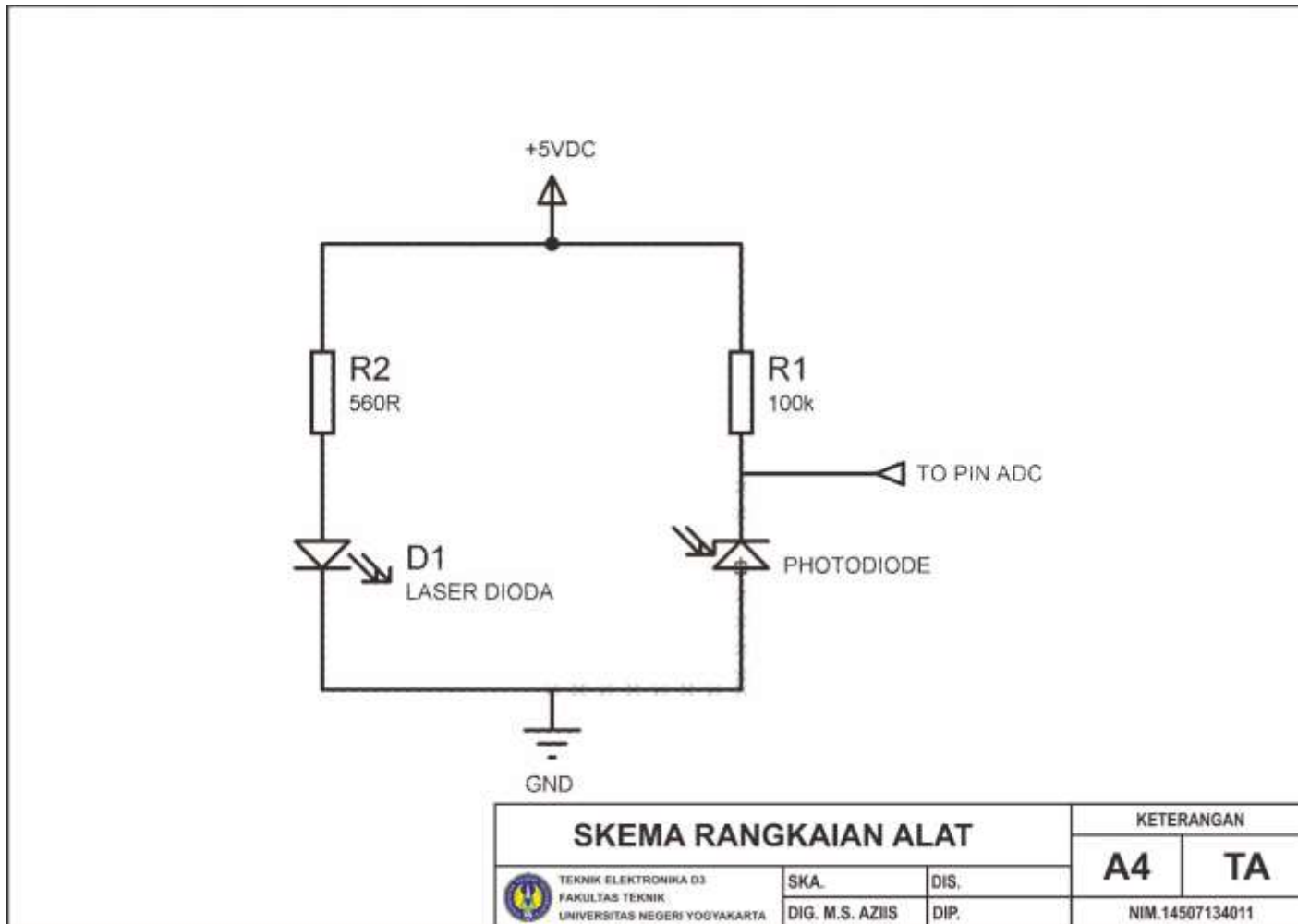
if(detik<10){lcd_mbahe.setCursor(14,3);lcd_mbahe.print("0");lcd_mbahe.print(detik);}
else if(detik>=10){lcd_mbahe.setCursor(14,3);lcd_mbahe.print(detik);}
}

```


Lampiran 3. Skema Rangkaian Alat



Lampiran 4. Skema Rangkaian Sensor Tetesan



Lampiran 5. *Layout Shield Arduino Mega*

LAYOUT SHIELD ARDUINO MEGA 2560			KETERANGAN	
			A4	TA
 TEKNIK ELEKTRONIKA D3 FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA	SKA.	DIS.	NIM.14507134011	
	DIG. M.S. AZIIS	DIP.		

Lampiran 6. *Datasheet* Arduino Mega 2560



Product 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.

Index

Technical Specifications	Page 2
How to use Arduino Programming Environment, Basic Tutorials	Page 6
Terms & Conditions	Page 7
Environmental Policies half sqm of green via Impatto Zero®	Page 7



Technical Specification

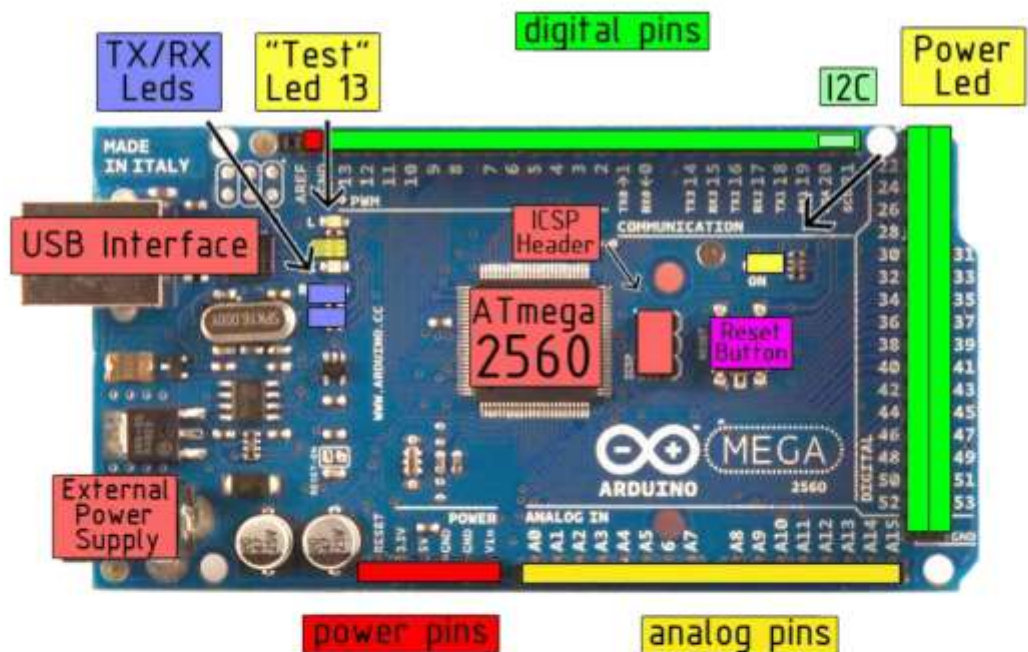


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

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

the board



radiospares **RADIONICS**



Power

The Arduino Mega2560 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.

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, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the 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.

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 is it 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.



radiospares

RADIONICS



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 can 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 MEGA

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```
Sketch - Arduino IDE
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
}
```

Press Compile button (to check for errors) Upload TX RX Flashing Blinking Led!



radiospares

RADIONICS



Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER 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

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer 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." The producer 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.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.

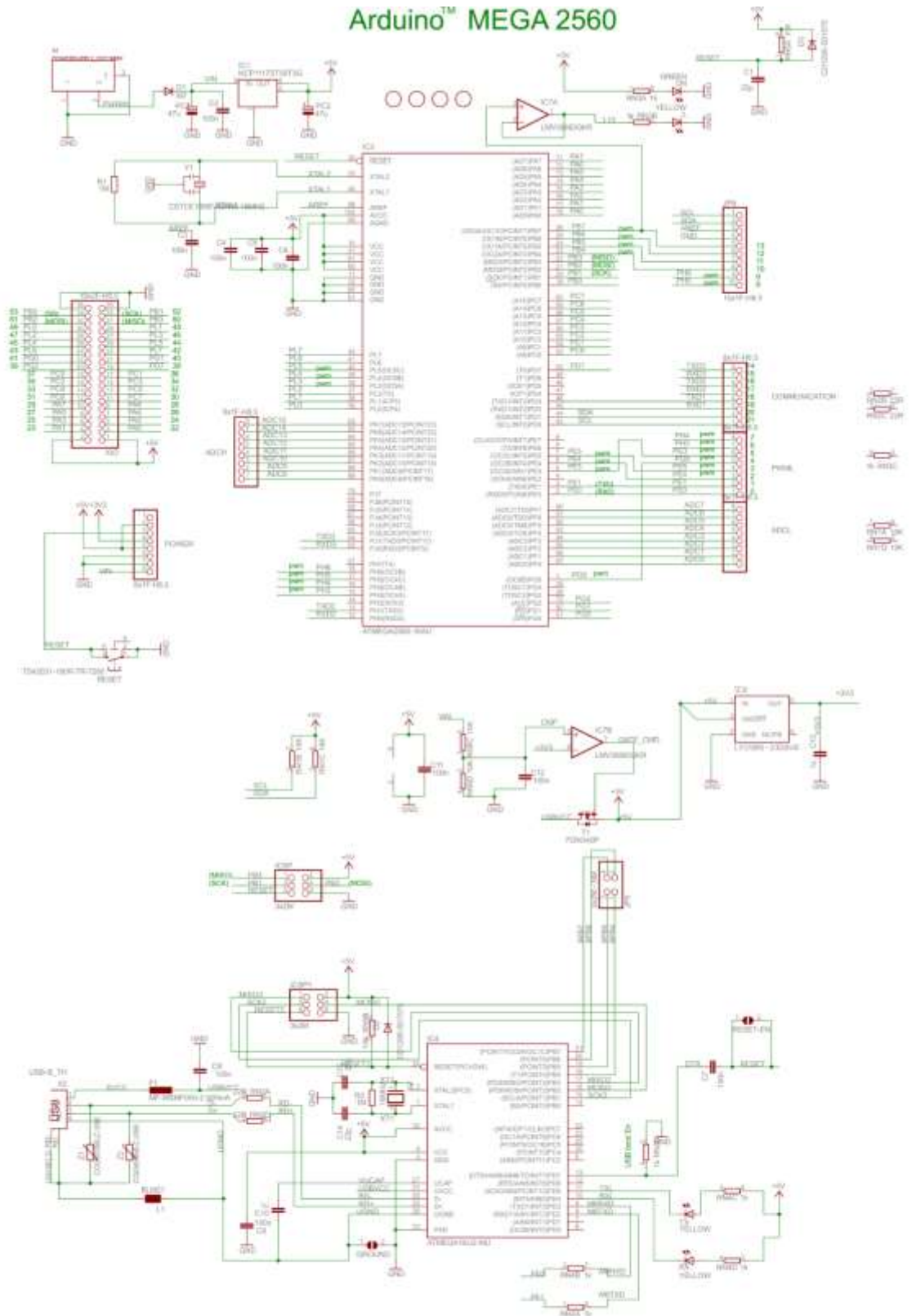


radiospares

RADIONICS

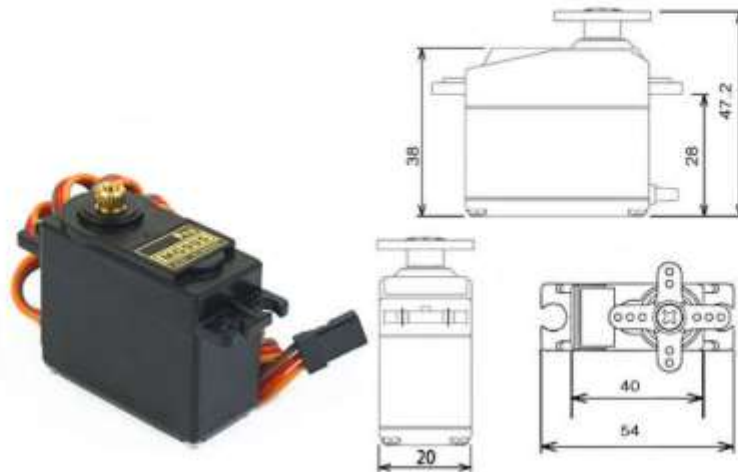


Arduino™ MEGA 2560



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.

MG995 High Speed Metal Gear Dual Ball Bearing Servo



The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

This high-speed standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG995 Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast!

Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 8.5 kgf·cm (4.8 V), 10 kgf·cm (6 V)
- Operating speed: 0.2 s/60° (4.8 V), 0.16 s/60° (6 V)
- Operating voltage: 4.8 V a 7.2 V
- Dead band width: 5 μs
- Stable and shock proof double ball bearing design
- Temperature range: 0 °C – 55 °C

Lampiran 8. *Datasheet* SIM800L V2

Description

The SIM800L module is a complete Quad-band GSM/GPRS solution in a LGA type which can be embedded in the customer, it has a set of TTL level serial interface, a set of power supply interface. Besides, there are a set of antenna interface on this module.

Features:

- Supply voltage range 3.4 ~ 4.4V and
- Current of 1A or more (the current is very important)
- It features Bluetooth, FM and Embedded AT (AT commands)
- Quad-band 850/900/1800/1900MHz
- Operation temperature: -40 ~ 85 degree Celsius

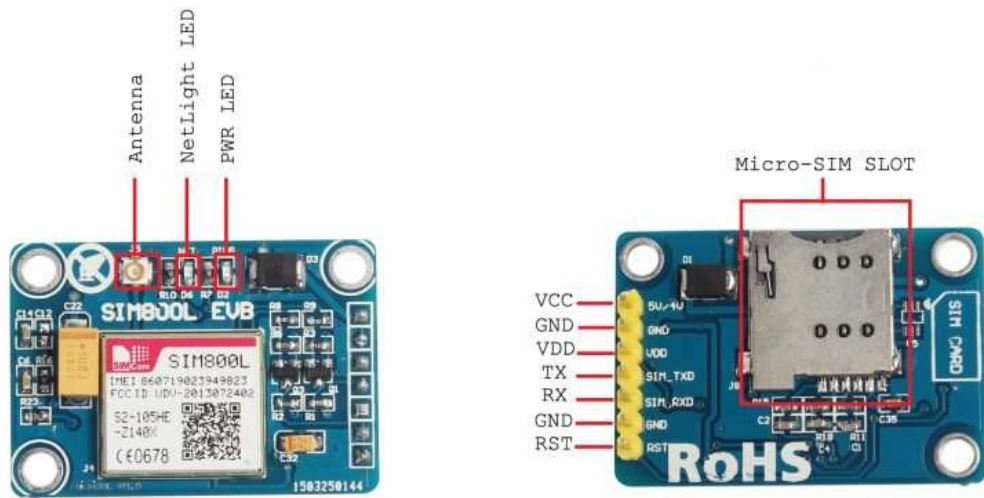
Applications:

- SIM800L support Quad-band 850/900/1800/1900MHz, it can transmit Voice, SMS and data information with low power
- Due to its size, it can fit into slim and compact demands of customer design.

How to get started with SIM800L GSM module

This SIM800L module supports quad-band GSM/GPRS network, available for GPRS and SMS message data remote transmission. SIM800L is one of the most commonly used GSM module among hobbyists and Arduino community. It is not very easy for a beginner to properly understand and use Arduino with SIM800L. Therefore, this post summarizes how a beginner could interact with SIM800 using Arduino. On the back it has a simslot on which you insert a micro-Sim. When it is receiving enough power the power LED on board must light.

Overview



Pinout Description:

There are 7 total pins on the SIM800L, which we are going to use to interface with Arduino.

VCC: External Supply Voltage input for SIM800L

GND: External Ground for SIM800L

VDD: MicroController Supply voltage input for SIM800L

RST: Reset pin for SIM800L

RXD: Serial communication (Receiver Pin)

TXD: Serial communication (Transfer Pin)

Description:

The WeMos D1 min PRO is a miniature wireless 802.11 (Wifi) microcontroller development board. It turns the very popular ESP8266 wireless microcontroller module into a fully fledged development board. Programming the D1 mini pro is as simple as programming any other Arduino based microcontroller as the module includes a built in microUSB interface allowing the module to be programmed directly from the Arduino IDE (requires the ESP8266 support to be added via board manager) with no additional hardware.

The D1 mini Pro is also designed to allow Wemos compatible shields to be plugged into the board in a similar way to the Arduino development board platform which greatly expands its capabilities. There is already a large range of compatible shields available and can also be purchased via our website. Included with the module is a set of headers (requires soldering) that allow thses shield to be easily added or removed from the D1 mini PRO.

Other features of the D1 Mini Pro include 11 digital input/output pins, 1 analogue input pin (3.2V Max), 16MB (128M bit) Flash, an external antenna connector, built in ceramic antenna and houses the new CP2104 US to UART IC. The D1 Pro is the same size as the D1 mini, but is lighter.



Specification

Operating Voltage: 3.3V
Digital I/O Pins: 11
Analogue Pins: 1 (Max input 3.2V)
Clock Speed: 80MHz/160MHz
Flash: 16MB

Dimensions

Length: 34.2mm
Width: 25.6mm
Weight: 2.5g

Pin	Function	ESP-8266 Pin
TX	TXD	TXD
RX	RXD	RXD
A0	Analog input, max 3.3V input	A0
D0	IO	GPIO16
D1	IO, SCL	GPIO5
D2	IO, SDA	GPIO4
D3	IO, 10k Pull-up	GPIO0
D4	IO, 10k Pull-up, BUILTIN_LED	GPIO2
D5	IO, SCK	GPIO14
D6	IO, MISO	GPIO12
D7	IO, MOSI	GPIO13
D8	IO, 10k Pull-down, SS	GPIO15
G	Ground	GND
5V	5V	-
3V3	3.3V	3.3V
RST	Reset	RST

Lampiran 10. *Datasheet Keypad 4x4*



Web Site: www.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com

Office: (916) 624-6333
Fax: (916) 624-6003
Sales: (888) 512-1024
Tech Support: (888) 997-8267

4x4 Matrix Membrane Keypad (#27899)

This 16-button keypad provides a useful human interface component for microcontroller projects. Convenient adhesive backing provides a simple way to mount the keypad in a variety of applications.

Features

- Ultra-thin design
- Adhesive backing
- Excellent price/performance ratio
- Easy interface to any microcontroller
- Example programs provided for the BASIC Stamp 2 and Propeller P8X32A microcontrollers

Key Specifications

- Maximum Rating: 24 VDC, 30 mA
- Interface: 8-pin access to 4x4 matrix
- Operating temperature: 32 to 122 °F (0 to 50°C)
- Dimensions:
Keypad, 2.7 x 3.0 in (6.9 x 7.6 cm)
Cable, 0.78 x 3.5 in (2.0 x 8.8 cm)

Application Ideas

- Security systems
- Menu selection
- Data entry for embedded systems



How it Works

Matrix keypads use a combination of four rows and four columns to provide button states to the host device, typically a microcontroller. Underneath each key is a pushbutton, with one end connected to one row, and the other end connected to one column. These connections are shown in Figure 1.

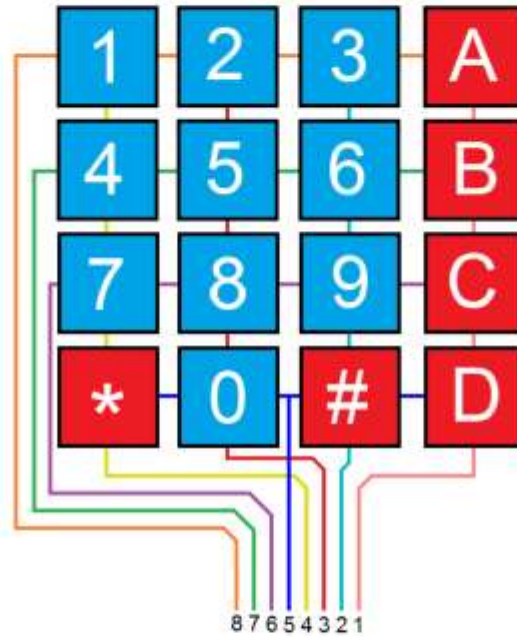


Figure 1: Matrix Keypad Connections

In order for the microcontroller to determine which button is pressed, it first needs to pull each of the four columns (pins 1-4) either low or high one at a time, and then poll the states of the four rows (pins 5-8). Depending on the states of the columns, the microcontroller can tell which button is pressed.

For example, say your program pulls all four columns low and then pulls the first row high. It then reads the input states of each column, and reads pin 1 high. This means that a contact has been made between column 4 and row 1, so button 'A' has been pressed.

3.0A, 150kHz, Step-Down Switching Regulator

LM2596

FEATURES

- 3.3V, 5.0V, 12V, and Adjustable Output Versions
- Adjustable Version Output Voltage Range, 1.2 to 37V
+/- 4%, Maximum Over Line and Load Conditions
- Guaranteed 3.0A Output Current
- Wide Input Voltage Range
- Requires Only 4 External Components
- 150kHz Fixed Frequency Internal Oscillator
- TTL Shutdown Capability, Low Power Standby Mode
- High Efficiency
- Uses Readily Available Standard Inductors
- Thermal Shutdown and Current Limit Protection
- Moisture Sensitivity Level 3

Applications

- Simple High-Efficiency Step-Down(Buck) Regulator
- Efficient Pre-Regulator for Linear Regulators
- On-Card Switching Regulators
- Positive to Negative Converter(Buck-Boost)
- Negative Step-Up Converters
- Power Supply for Battery Chargers

DESCRIPTION

The LM2596 series of regulators are monolithic integrated circuits ideally suited for easy and convenient design of a step-down switching regulator(buck converter).

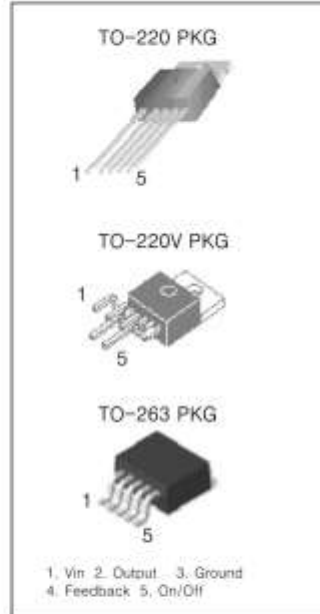
All circuits of this series are capable of driving a 3.0A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3V, 5.0V,12V, and an adjustable output version.

These regulators were designed to minimize the number of external components to simplify the power supply design. Standard series of inductors optimized for use with the LM2596 are offered by several different inductor manufacturers.

Since the LM2596 converter is a switch-mode power supply, its efficiency is significantly higher in comparison with popular three-terminal linear regulators, especially with higher input voltages.

In many cases, the power dissipated is so low that no heatsink is required or its size could be reduced dramatically. A standard series of inductors optimized for use with the TJ2596 are available from several different manufacturers. This feature greatly simplifies the design of switch-mode power supplies. The LM2596 features include a guaranteed +/- 4% tolerance on output voltage within specified input voltages and output load conditions, and +/-15% on the oscillator frequency (+/- 2% over 0°C to 125°C).

External shutdown is included, featuring 80µA (typical) standby current. Self protection features include a two stage frequency reducing current limit for output switch and an over temperature shutdown for complete protection under fault conditions.



ORDERING INFORMATION

Device	Marking	Package
LM2596T-X.X	LM2596-X.X	TO-220
LM2596TV-X.X	LM2596-X.X	TO-220V
LM2596R	LM2596-X.X	TO-263

Typical Application (Fixed Output Voltage Versions)

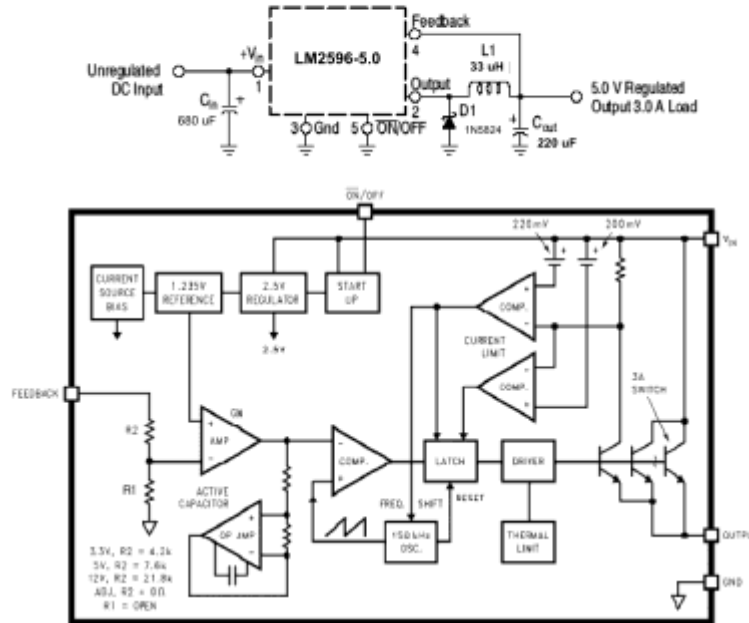


Figure 1. Block Diagram and Typical Application

ABSOLUTE MAXIMUM RATINGS

(Absolute Maximum Ratings indicate limits beyond which damage to the device may occur.)

Rating	Symbol	Value	Unit
Maximum Supply Voltage	V _{in}	45	V
On/Off Pin Input Voltage	-	-0.3V ≤ V ≤ +V _{in}	V
Output Voltage to Ground (Steady-State)	-	- 1.0	V
Power Dissipation			
TO-220 5Lead	P _D	Internally Limited	W
Thermal Resistance, Junction to Ambient	P _{θJA}	65	°C/W
Thermal Resistance, Junction to Case	P _{θJC}	5.0	°C/W
TO-263 5Lead	P _D	Internally Limited	W
Thermal Resistance, Junction to Ambient	P _{θJA}	70	°C/W
Thermal Resistance, Junction to Case	P _{θJC}	5.0	°C/W
Storage Temperature Range	T _{stg}	-60 to +150	°C
Minimum ESD Rating(Human Body Model : C=100 pF, R=1.5kΩ)	-	2.0	kV
Lead Temperature (Soldering,10seconds)	-	260	°C
Maximum Junction Temperature	T _J	150	°C

HTC

PIN FUNCTION DESCRIPTION

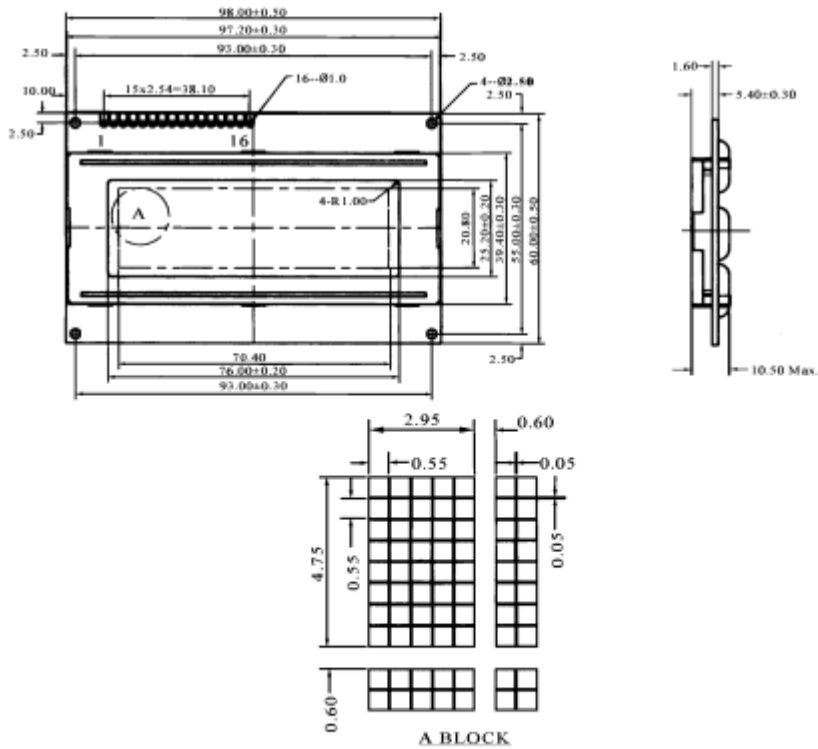
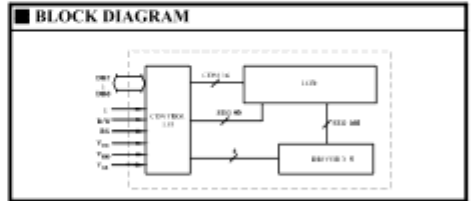
	Symbol	Description
1	Vin	This pin is the positive input supply for the LM2596 step-down switching regulator. In order to minimize voltage transients and to supply the switching currents needed by the regulator, a suitable input bypass capacitor must be present. (Cin in Figure 1).
2	Output	This is the emitter of the internal switch. The saturation voltage Vsat of this output switch is typically 1.5 V. It should be kept in mind that the PCB area connected to this pin should be kept to a minimum in order to minimize coupling to sensitive circuitry.
3	Gnd	Circuit ground pin. See the information about the printed circuit board layout.
4	Feedback	This pin senses regulated output voltage to complete the feedback loop. The signal is divided by the internal resistor divider network R2, R1 and applied to the non-inverting input of the internal error amplifier. In the Adjustable version of the LM2596 switching regulator this pin is the direct input of the error amplifier and the resistor network R2, R1 is connected externally to allow programming of the output voltage.
5	ON/OFF	It allows the switching regulator circuit to be shut down using logic level signals, thus dropping the total input supply current to approximately 80mA. The threshold voltage is typically 1.4V. Applying a voltage above this value (up to +Vin) shuts the regulator off. If the voltage applied to this pin is lower than 1.4V or if this pin is left open, the regulator will be in the "on" condition.

Lampiran 12. Datasheet LCD 20x4

■ ABSOLUTE MAXIMUM RATINGS					
Item	Symbol	Standard Value			Unit
		Min.	Typ.	Max.	
Supply Voltage for Logic	V_{DD}	0	—	7.0	V
Supply Voltage for LCD Driver	$V_{DD}-V_{EE}$	—	—	13.5	V
Input Voltage	V_i	V_{DD}	—	V_{DD}	V
Operate Temp.	T_{op}	0	—	50	°C
Storage Temp.	T_{stg}	-20	—	70	°C

■ ELECTRICAL CHARACTERISTICS (REFLECTIVE TYPE)						
Item	Symbol	Test Condition	Standard Value			Unit
			Min.	Typ.	Max.	
Input "High" Voltage	V_{IH}	—	2.2	—	V_{EE}	V
Input "Low" Voltage	V_{IL}	—	—	—	0.6	V
Output "High" Voltage	V_{OH}	$I_{OH}=0.2mA$	2.2	—	—	V
Output "Low" Voltage	V_{OL}	$I_{OL}=1.2mA$	—	—	0.4	V
Supply Current	I_{DD}	$V_{DD}=5.0V$	—	2.5	4.0	mA

■ PIN FUNCTIONS					
No.	Symbol	Function	No.	Symbol	Function
1	V_{SS}	GND, 0V	10	DB3	Data Bus
2	V_{RH}	-5V	11	DB4	—
3	V_{EE}	for LCD Drive	12	DB5	—
4	RS	Function Select	13	DB6	—
5	R/W	Read/Write	14	DB7	—
6	E	Enable Signal	15	LEDA	LED Power Supply
7-9	DB0-DB2	Data Bus Line	16	LEDB	



HD44780U

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)

ROM Code	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00000000	0	@	P	\	P				-	9	3	α	p			
00000001 (2)	!	1	A	Q	a	q			■	ア	チ	4	ä	q		
00000010 (3)	"	2	B	R	b	r			「	イ	ツ	×	β	θ		
00000011 (4)	#	3	C	S	c	s			」	ウ	テ	ε	ε	ε		
00000100 (5)	\$	4	D	T	d	t			\	エ	ト	†	μ	Ω		
00000101 (6)	%	5	E	U	e	u			・	オ	ナ	1	σ	ü		
00000110 (7)	&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ		
00000111 (8)	'	7	G	W	g	w			フ	キ	ヌ	ラ	q	π		
00001000 (1)	(8	H	X	h	x			イ	ク	ネ	リ	∫	×		
00001001 (2))	9	I	Y	i	y			ウ	ケ	ル	ル	∫	∫		
00001010 (3)	*	:	J	Z	j	z			エ	コ	ハ	レ	j	∫		
00001011 (4)	+	;	K	L	k	l			オ	サ	ヒ	ロ	*	∫		
00001100 (5)	,	<	L	¥	1	l			ハ	シ	フ	ワ	φ	∫		
00001101 (6)	-	=	M	J	m	>			ユ	ス	ヘ	ン	∫	∫		
00001110 (7)	.	>	N	^	n	→			ヨ	セ	ホ	ッ	∫	∫		
00001111 (8)	/	?	O	_	o	←			ツ	ソ	マ	□	ö	■		

Note: The user can specify any pattern for character-generator RAM.

RED LASER DIODE

DL-3149-057



Ver.1 Jun. 2000

Features

- Short wavelength : 670 nm (Typ.)
- Low threshold current : $I_{th} = 25$ mA (Typ.)
- High operating temperature : 5 mW at 60°C
- Small package : $\phi 5.6$ mm

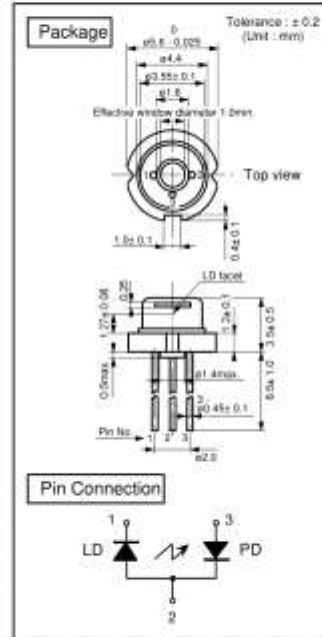
Applications

- Laser beam printer
- Bar-code scanner

Absolute Maximum Ratings

($T_c=25^\circ\text{C}$)

Parameter	Symbol	Rating	Unit
Light Output	CW	P_o	7
Reverse Voltage	Laser	V_R	2
	PD		30
Operating Temperature	T_{opr}	-10 ~ +60	$^\circ\text{C}$
Storage Temperature	T_{stg}	-40 ~ +85	$^\circ\text{C}$



Electrical and Optical Characteristics ^{1) 2)}

($T_c=25^\circ\text{C}$)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit	
Threshold Current	I_{th}	CW	-	25	35	mA	
Operating Current	I_{op}	$P_o=5\text{mW}$	-	40	45	mA	
Operating Voltage	V_{op}	$P_o=5\text{mW}$	-	2.3	2.6	V	
Lasing Wavelength	λ_p	$P_o=5\text{mW}$	660	670	678	nm	
Beam ³⁾ Divergence	Perpendicular	Q_v	$P_o=5\text{mW}$	25	30	35	$^\circ$
	Parallel	Q_h	$P_o=5\text{mW}$	6.5	8	10	$^\circ$
Off Axis Angle	Perpendicular	dQ_v	-	-	± 3	$^\circ$	
	Parallel	dQ_h	-	-	± 3	$^\circ$	
Differential Efficiency	dP_o/dI_{op}	-	0.2	0.4	0.6	mW/mA	
Monitoring Output Current	I_m	$P_o=5\text{mW}$	0.5	1.5	2	mA	
Astigmatism	A_s	$P_o=5\text{mW}$	-	8	-	μm	

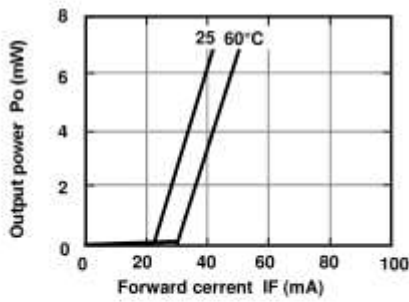
1) Initial values 2) All the above values are evaluated with Tottori Sanyo's measuring apparatus

3) Full angle at half maximum Note : The above product specification are subject to change without notice.

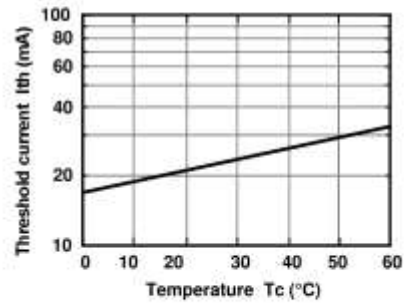
Tottori SANYO Electric Co., Ltd. Electronic Device Business Headquarters
 LED Division
 5-318, Tachikawa, Tottori 680-8634 Japan TEL : +81-857-21-2137 FAX : +81-857-21-2161

Characteristics

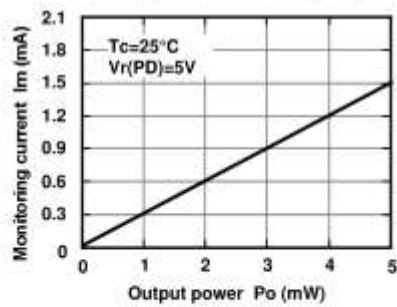
Output power vs. Forward current



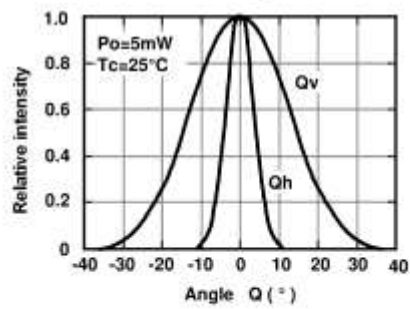
Threshold current vs. Temperature



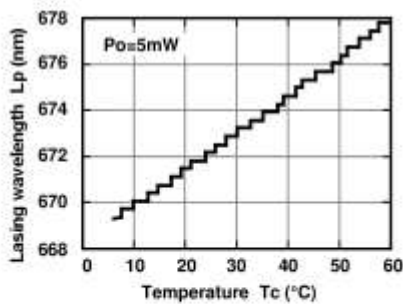
Monitoring current vs. Output power



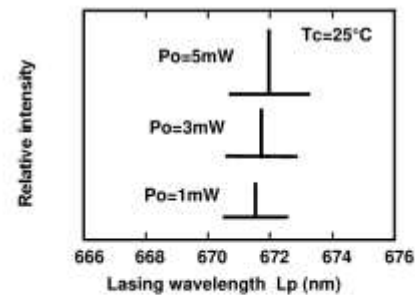
Beam divergence



Lasing wavelength vs. Temperature



Lasing wavelength vs. Output power



This is typical data and it may not represent all products.

Lampiran 14. *Datasheet* Photodioda



5mm photodiode PD333-3C/H0/L2

Features

- Fast response time
- High photo sensitivity
- Small junction capacitance
- Pb free
- The product itself will remain within RoHS compliant version
- Compliance with EU REACH



Description

- PD333-3C/H0/L2 is a high speed and high sensitive PIN photodiode in a standard 5Φ plastic package. Due to its water clear epoxy the device is sensitive to infrared radiation

Applications

- High speed photo detector
- Security system
- Camera

Device Selection Guide

Chip Materials	Lens Color
Silicon	Water clear

Absolute Maximum Ratings (Ta=25°C)

Parameter	Symbol	Rating	Unit
Reverse Voltage	V_R	32	V
Operating Temperature	T_{opr}	-25 ~ +85	°C
Storage Temperature	T_{stg}	-40 ~ +100	°C
Soldering Temperature	T_{sol}	260	°C
Power Dissipation at (or below) 25°C Free Air Temperature	P_T	150	mW

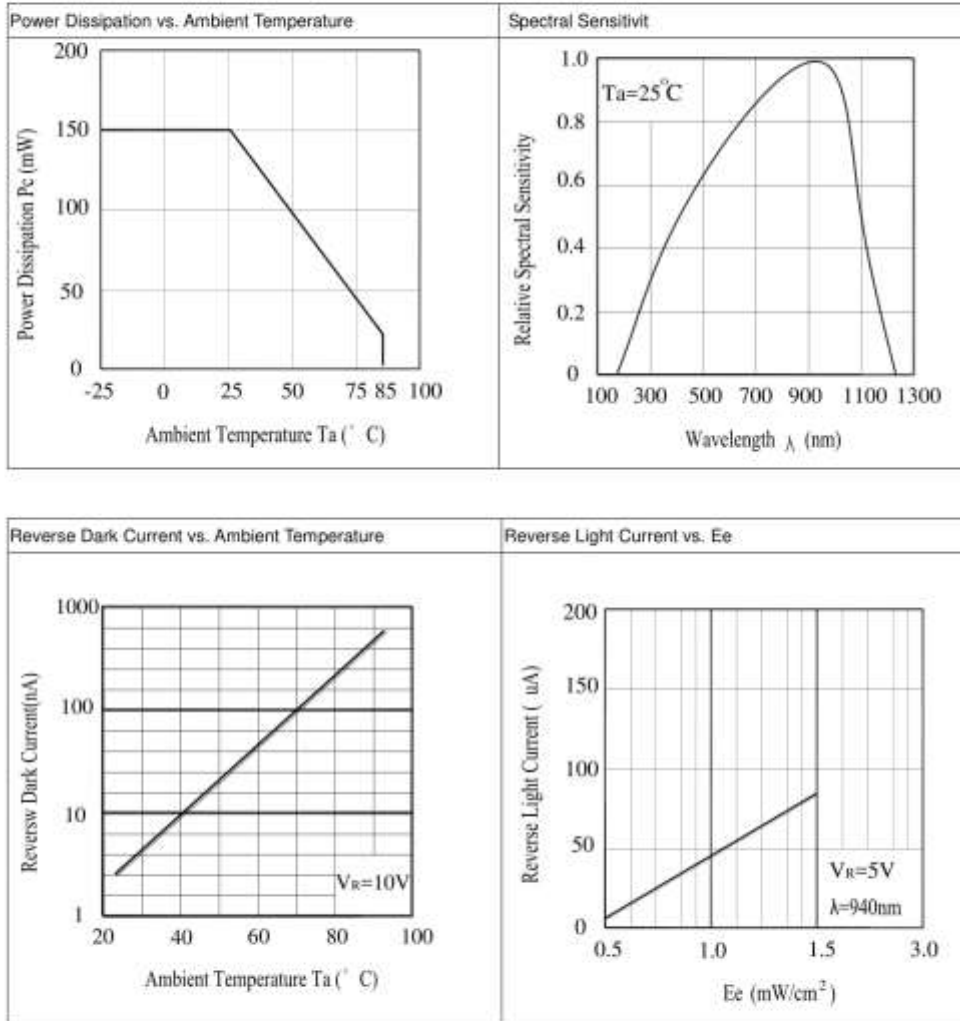
Electro-Optical Characteristics (Ta=25°C)

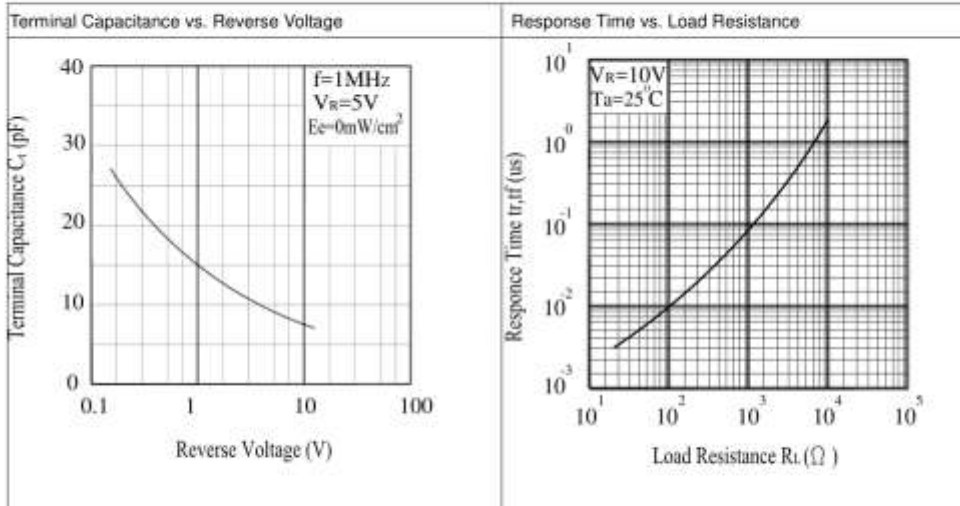
Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Range Of Spectral Bandwidth	$\lambda_{0.5}$	400	-----	1100	nm	-----
Wavelength Of Peak Sensitivity	λ_p	-----	940	-----	nm	-----
Open-Circuit Voltage	V_{oc}	-----	0.39	-----	V	$E_e=1mW/cm^2$ $\lambda_p=940nm$
Short-Circuit Current	I_{sc}	-----	40	-----	μA	$E_e=1mW/cm^2$ $\lambda_p=940nm$
Reverse Light Current	I_L	36	40	-----	μA	$E_e=1mW/cm^2$ $\lambda_p=940nm$ $V_R=5V$
Reverse Dark Current	I_D	-----	5	30	nA	$E_e=0mW/cm^2$ $V_R=10V$
Reverse Breakdown Voltage	V_{BR}	32	170	-----	V	$E_e=0mW/cm^2$ $I_R=100\mu A$
Total Capacitance	C_T	-----	18	-----	pF	$E_e=0mW/cm^2$ $V_R=5V$ $f=1MHz$
Rise Time/ Fall Time	t_r / t_f	-----	45/45	-----	ns	$V_R=10V$ $R_L=100\Omega$
View Angle	2 $\theta_{1/2}$	-----	80	-----	deg	$I_F=20mA$

Note:

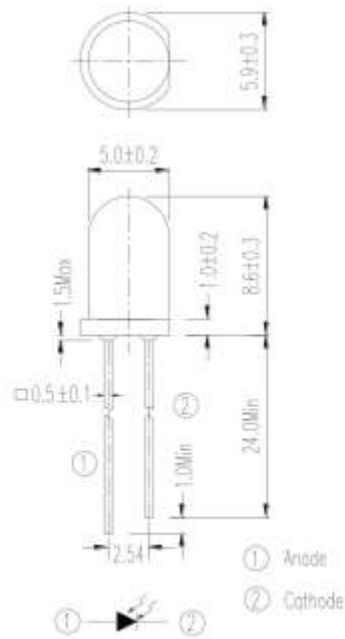
Tolerance of Luminous Intensity: $\pm 10\%$
Tolerance of Dominant Wavelength: $\pm 1nm$
Tolerance of Forward Voltage: $\pm 0.1V$

Typical Electro-Optical Characteristics Curves





Package Dimension



Note: Tolerances unless dimensions ± 0.25 mm

Label Form Specification



- CPN: Customer's Product Number
- P/N: Product Number
- QTY: Packing Quantity
- CAT: Luminous Intensity Rank
- HUE: Dom. Wavelength Rank
- REF: Forward Voltage Rank
- LOT No: Lot Number
- X: Month
- Reference: Identify Label Number

Packing Quantity Specification

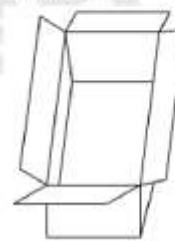
■ Anti-electrostatic bag



■ Inner Carton



■ Outside Carton



1,200-500PCS/1Bag · 5Bags/1 Inner Carton
2,10 Inner Cartons/1 Outside Carton