



**SIMULATOR KURSI RODA OTOMATIS DENGAN SENSOR *FLEX*  
BERBASIS MIKROKONTROLER**

**PROYEK AKHIR**

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



**Disusun Oleh:  
AFFINANNISA TIARA NIRWANI  
NIM. 15507134002**

**PROGRAM STUDI TEKNIK ELEKTRONIKA  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI YOGYAKARTA  
2018**

## LEMBAR PERSETUJUAN

### PROYEK AKHIR

#### SIMULATOR KURSI RODA OTOMATIS DENGAN SENSOR FLEX BERBASIS MIKROKONTROLER

Disusun Oleh:

Affinannisa Tiara Nirwani

NIM. 15507134002

telah memenuhi syarat dan disetujui oleh Dosen Pembimbing untuk dilaksanakan  
Ujian Proyek Akhir bagi yang bersangkutan.

Yogyakarta, Juli 2018

Mengetahui,

Ketua Program Studi

Menyetujui,

Dosen Pembimbing

**Dr. Dra. Sri Waluyanti, M.Pd**

NIP. 19581218198603 2 001

**Dr. Dra. Sri Waluyanti, M.Pd**

NIP. 195812181998603 2 001

## HALAMAN PENGESAHAN

Proyek Akhir

### SIMULATOR KURSI RODA OTOMATIS DENGAN SENSOR FLEX BERBASIS MIKROKONTROLER

Disusun Oleh:

Affinannisa Tiara Nirwani

NIM. 15507134002

Telah dipertahankan di depan Tim Penguji Proyek Akhir Program Studi Teknik Elektronika Fakultas Teknik Universitas Negeri Yogyakarta

Pada tanggal: .....

Nama	Jabatan	Tanda Tangan	Tanggal
Dr. Dra. Sri Waluyanti, M.Pd	Ketua Penguji	.....	.....
Bekti Wulandari, M.Pd	Sekretaris Penguji	.....	.....
Dessy Irmawati, M.T	Penguji Utama	.....	.....

Yogyakarta,  
Dekan Fakultas Teknik  
Universitas Negeri Yogyakarta

**Dr. Widarto, M.Pd**

**NIP. 19631230 198812 1 001**

## **SURAT PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Affinannisa Tiara Nirwani

NIM : 15507134002

Jurusan : Pendidikan Teknik Elektronika dan Informatika

Fakultas : Teknik

Judul : Simulator Kursi Roda Otomatis dengan Sensor *Flex* Berbasis  
Mikrokontroler

menyatakan bahwa dalam Proyek Akhir ini benar-benar karya saya sendiri. Sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain, kecuali sebagai acuan atau kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim.

Yogyakarta, 9 Juli 2018

Yang menyatakan,

Affinannisa Tiara Nirwani

NIM. 15507134002

## **MOTTO**

*“Semangat, Konsisten, Sukses”*

*(Affinannisa)*

*“Ketika kita meminta kepada Allah kebijakan, kita diberi permasalahan. Ketika kita meminta cinta, kita diberi orang-orang yang bermasalah. Ketika kita meminta kebaikan, kita diberi kesempatan-kesempatan. Ketika kita meminta kemurahan hati, kita diberi otak dan kaki”*

*(HR. Bukhari Muslim)*

*“Kalau kalian tidak mau repot, jangan berjuang, karena perjuangan selalu membutuhkan pengorbanan”*

*(K.H.R. As'ad Syamsul Arifin)*

*“Rendahkanlah hatimu serendah-rendahnya sehingga orang lain tidak bisa merendahkanmu. Karena pusaka yang paling ampuh untuk kehormatan kita adalah dengan merendahkan hati kita sendiri”*

*(KH. Anwar Zahid)*

## **HALAMAN PERSEMBAHAN**

Seiring rasa syukur kepada Allah SWT, karya ini saya persembahkan untuk:

1. Kedua orangtuaku Bapak Agus Krisnanto WD dan Ibu Lestari yang telah melimpahkan kasih sayang, perhatian, dukungan material maupun spiritual dan doanya yang selalu menyertai.
2. Seluruh dosen pengajar Teknik Elektronika yang sudah memberikan banyak ilmu dari awal masuk perkuliahan hingga sampai akhir perkuliahan.
3. Adikku Elmintia Us wah Fahmawati yang selalu mendukungku.
4. Paksi Wizurai Sakti, teman-teman kos Bapak Sapari, kos Pak Barjo Karangmalang, kos putri cahaya dan semua yang sudah membantu dan selalu memberi semangat tanpa henti.
5. Teman-teman kelas B Teknik Elektronika angkatan 2015.
6. Seluruh mahasiswa Jurusan Pendidikan Teknik Elektronika dan Informatika Fakultas Teknik Universitas Negeri Yogyakarta.
7. Almamater Universitas Negeri Yogyakarta

## **SIMULATOR KURSI RODA OTOMATIS DENGAN SENSOR *FLEX* BERBASIS MIKROKONTROLER**

Oleh:

Affinannisa Tiara Nirwani

NIM. 15507134002

### **ABSTRAK**

Tujuan pembuatan simulator kursi roda otomatis dengan sensor *flex* adalah membangun sebuah simulator yang berfungsi mengendalikan kursi roda dengan menggunakan sensor *flex* yang dipasang pada sarung tangan agar kursi roda bergerak sesuai intruksi pengguna. Selain itu merealisasikan sebuah kursi roda yang digunakan dengan cara yang lebih mudah, efisien, dan diharapkan dapat mempermudah pengguna dalam menggerakkan kursi roda dengan jari tangannya.

Perancangan simulator kursi roda otomatis dengan sensor *flex* berbasis mikrokontroler terdiri dari beberapa tahap dimulai dari identifikasi kebutuhan, analisis kebutuhan, perancangan sistem, perancangan perangkat keras dan perangkat lunak, pembuatan alat, pengujian alat dan pengambilan data. Komponen utama yang digunakan adalah arduino uno sebagai pusat pengolah data, sensor *flex* sebagai *input* yang mendeteksi kelengkungan jari, sensor ultrasonik HCSR-04 sebagai sensor pendeksi jarak, *buzzer* sebagai *output* dari sensor ultrasonik HCSR-04, *driver* motor L298N sebagai pengirim perintah gerakan motor DC, dan motor DC sebagai penggerak kursi roda.

Berdasarkan hasil pengujian yang telah dilakukan, sensor *flex* sudah dapat mendeteksi kelengkungan jari tangan yang akan diolah pada arduino uno. *Driver* motor L298N sudah dapat mengirimkan perintah ke motor DC. Motor DC sudah dapat bergerak sesuai dengan intruksi pengguna kursi roda yaitu bergerak dengan gerakan maju, mundur, kanan, kiri. Sensor ultrasonik HCSR-04 sudah dapat mendeteksi adanya halangan di depan sensor sampai dengan 179 cm. *Buzzer* sebagai *output* dari sensor ultrasonik HCSR-04 dapat berbunyi ketika ada halangan di depan sensor, *buzzer* akan berbunyi ketika ada halangan kurang dari 30 cm. Hasil pengujian unjuk kerja sistem dapat bekerja sesuai dengan fungsinya.

**Kata kunci:** Kursi roda otomatis, Sensor *Flex*, Motor DC, Arduino Uno.

## KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT atas berkat karunia dan rahmat-Nya, Proyek Akhir dalam rangka memenuhi sebagian persyaratan untuk mendapatkan gelar Ahli Madya dengan judul "**Simulator Kursi Roda Otomatis dengan Sensor Flex Berbasis Mikrokontroler**" dapat disusun sesuai harapan.

Pembuatan Proyek Akhir dapat diselesaikan tidak lepas dari dukungan, baik secara langsung maupun tidak langsung. Oleh karen itu, penulis menyampaikan ucapan terima kasih kepada:

1. Dr. Dra. Sri Waluyanti, M.Pd selaku Ketua Program Studi Teknik Elektronika D3, Fakultas Teknik Universitas Negeri Yogyakarta dan juga selaku dosen Pembimbing Proyek Akhir ini.
2. Dr. Fatchul Arifin, M.T selaku Ketua Jurusan Pendidikan Teknik Elektronika dan Informatika, Fakultas Teknik, Universitas Negeri Yogyakarta.
3. Dr. Widarto, M.Pd selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.
4. Seluruh Staff dan karyawan Jurusan Pendidikan Teknik Elektronika dan Informatika Fakultas Teknik Universitas Negeri Yogyakarta.
5. Semua pihak yang tidak dapat disebutkan disini atas bantuan dan perhatiannya selama penyusunan tugas akhir ini.

Akhirnya penulis menyadari bahwa dalam penyusunan laporan proyek akhir ini masih banyak kekurangan dan jauh dari kesempurnaan. Saran dan kritik yang sifatnya membangun sangat penulis harapkan. Semoga laporan proyek akhir ini bermanfaat bagi yang memerlukan. Terimakasih.

Yogyakarta, 9 Juli 2018

Affinannisa Tiara Nirwani

## DAFTAR ISI

	Halaman
HALAMAN SAMPUL .....	i
LEMBAR PERSETUJUAN .....	ii
HALAMAN PENGESAHAN .....	iii
SURAT PERNYATAAN .....	iv
MOTTO .....	v
HALAMAN PERSEMAHAN .....	vi
ABSTRAK .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI .....	x
DAFTAR TABEL .....	xii
DAFTAR GAMBAR .....	xiii
DAFTAR LAMPIRAN .....	xv
<b>BAB I PENDAHULUAN</b>	
A. Latar Belakang .....	1
B. Identifikasi Masalah .....	3
C. Batasan Masalah .....	4
D. Rumusan Masalah .....	4
E. Tujuan .....	4
F. Manfaat .....	5
G. Keaslian Gagasan .....	5
<b>BAB II PENDEKATAN PEMECAHAN MASALAH</b>	
A. Simulator .....	8
B. Disabilitas .....	8
C. Kursi Roda .....	9
D. Sensor Ultrasonik HC-SR04 .....	15
E. Arduino Uno R3 .....	20
F. Sensor <i>Flex</i> .....	25
G. Baterai <i>Li-Po</i> .....	27
H. Motor DC .....	29
I. Modul L298N <i>Driver Motor DC Dual H-Bridge</i> .....	31
J. <i>Buzzer</i> .....	35
K. <i>Software Arduino IDE</i> .....	36

BAB III KONSEP PERANCANGAN	
A. Identifikasi Kebutuhan .....	39
B. Analisis Kebutuhan .....	40
C. Perancangan Sistem .....	43
D. Langkah Pembuatan Alat .....	57
E. Spesifikasi Alat .....	61
F. Pengujian Alat .....	63
BAB IV PENGUJIAN DAN PEMBAHASAN	
A. Pengujian .....	70
B. Pembahasan .....	74
BAB V KESIMPULAN DAN SARAN	
A. Kesimpulan .....	93
B. Keterbatasan Alat .....	94
C. Saran .....	94
DAFTAR PUSTAKA .....	95
LAMPIRAN .....	98

## **DAFTAR TABEL**

Tabel 1. Pemakaian dan Sifat Motor DC .....	31
Tabel 2. Identifikasi Kebutuhan.....	42
Tabel 3. Rencana Pengukuran Tegangan Catu Daya <i>Driver</i> Motor DC.....	64
Tabel 4. Rencana Konfigurasi Sensor <i>Flex</i> .....	65
Tabel 5. Rencana Pengukuran Sensor <i>Flex</i> .....	65
Tabel 6. Rencana Pengukuran Sensor Ultrasonik .....	66
Tabel 7. Rencana Pengujian <i>Buzzer</i> .....	67
Tabel 8. Rencana Pengujian Gerak Motor DC.....	68
Tabel 9. Rencana Pengujian Gerak Motor DC dengan Sensor Ultrasonik .....	68
Tabel 10. Rencana Pengujian Secara Keseluruhan .....	69
Tabel 11. Pengukuran Tegangan Catu Daya <i>Driver</i> Motor DC.....	70
Tabel 12. Konfigurasi Sensor <i>Flex</i> .....	71
Tabel 13. Pengukuran Sensor <i>Flex</i> .....	71
Tabel 14. Pengukuran Sensor Ultrasonik .....	72
Tabel 15. Pengujian <i>Buzzer</i> .....	73
Tabel 16. Pengujian Gerak Motor DC .....	73
Tabel 17. Pengujian Gerak Motor DC dengan Sensor Ultrasonik .....	73
Tabel 18. Pengujian Secara Keseluruhan .....	74

## DAFTAR GAMBAR

Gambar 1. Kursi Roda Manual .....	10
Gambar 2. Kursi Roda Elektrik .....	11
Gambar 3. Kursi Roda Olahraga .....	12
Gambar 4. Kursi Roda Berdiri .....	13
Gambar 5. Kursi Roda <i>Bariatric</i> .....	13
Gambar 6. Kursi Roda <i>Pediatric</i> .....	14
Gambar 7. Kursi Roda <i>Travelling</i> .....	15
Gambar 8. Konfigurasi Pin dan Tampilan Sensor Ultrasonik HC-SR04 .....	18
Gambar 9. Prinsip Kerja HC-SR04 .....	19
Gambar 10. Mikrokontroler Arduino Uno R3 .....	21
Gambar 11. Konfigurasi Pin–Pin Arduino Uno .....	23
Gambar 12. Rangkaian Pembagi Tegangan Sensor <i>Flex</i> .....	26
Gambar 13. Sensor <i>Flex</i> .....	26
Gambar 14. Baterai <i>Li-Po</i> .....	28
Gambar 15. Motor DC .....	29
Gambar 16. Konfigurasi Pin pada Modul L298N .....	33
Gambar 17. Modul L298N Tampak Belakang .....	34
Gambar 18. Modul L298N Tampak Depan .....	34
Gambar 19. Bentuk Fisik <i>Buzzer</i> .....	36
Gambar 20. Tampilan <i>Software</i> Arduino IDE .....	38
Gambar 21. Blok Diagram Rangkaian .....	44
Gambar 22. Rangkaian Sensor <i>Flex</i> .....	47
Gambar 23. Rangkaian Sensor Ultrasonik .....	48
Gambar 24. Rangkaian Modul L298N dengan 2 Motor DC .....	49
Gambar 25. Inisialisasi Pin pada Program Arduino Uno .....	50
Gambar 26. <i>Flowchart</i> Kontrol Gerakan Kursi Roda .....	52
Gambar 27. <i>Flowchart</i> Gerakan Maju .....	53
Gambar 28. <i>Flowchart</i> Gerakan Mundur .....	54

Gambar 29. <i>Flowchart</i> Gerakan Kanan .....	55
Gambar 30. <i>Flowchart</i> Gerakan Kiri .....	56
Gambar 31. <i>Flowchart</i> Kursi Roda Berhenti .....	57
Gambar 32. Kerangka Kursi Roda Tampak Depan .....	59
Gambar 33. Kerangka Kursi Roda Tampak Belakang .....	60
Gambar 34. Ukuran <i>Black Box</i> .....	61
Gambar 35. Perbandingan Jarak Terukur dengan Jarak Sebenarnya .....	83
Gambar 36. Foto Alat Secara Keseluruhan.....	101
Gambar 37. Isi <i>Box</i> Kontrol .....	102

## **DAFTAR LAMPIRAN**

Lampiran 1. Skema Rangkaian Keseluruhan .....	99
Lampiran 2. Daftar Komponen .....	100
Lampiran 3. Foto Alat .....	101
Lampiran 4. Petunjuk Pengoperasian Alat .....	103
Lampiran 5. <i>Listing Program</i> .....	104
Lampiran 6. <i>Datasheet Sensor Flex</i> .....	110
Lampiran 7. <i>Datasheet Sensor Ultrasonik HC-SR04</i> .....	112
Lampiran 8. <i>Datasheet Driver Motor L298N</i> .....	115
Lampiran 9. <i>Datasheet Arduino Uno</i> .....	118

## **BAB I**

### **PENDAHULUAN**

#### **A. Latar Belakang**

Seiring perkembangan zaman terutama di bidang elektronika banyak alat elektronika yang mempermudah aktivitas manusia dan membantu pekerjaan manusia. Perkembangan tersebut dapat dilihat dari berbagai bidang, salah satunya dalam bidang kesehatan yang diantaranya meliputi kualitas layanan, ragam alat kesehatan, terapi dan lain sebagainya. Banyaknya bidang alat kesehatan dalam rumah sakit yang mulai menggunakan alat-alat modern di bidang elektronika, untuk mendapatkan hasil yang lebih cepat, akurat, dan efisien.

Kursi roda merupakan salah satu alat kesehatan yang dapat membantu penyandang cacat kaki dan orang yang fisiknya lemah, untuk berpindah dari satu tempat ke tempat lain. Ada beberapa penyebab orang yang menggunakan kursi roda, baik dikarenakan penyakit, cedera, maupun cacat. Berdasarkan (Badan Pusat Statistik (BPS) dalam Infodatin, 2014: 6) melalui data Survei Sosial Ekonomi Nasional (Susenas) pada tahun 2012 didapatkan estimasi penduduk Indonesia yang menyandang disabilitas sebesar 2,45%. Peningkatan dan penurunan persentase penyandang disabilitas dipengaruhi adanya perubahan konsep dan definisi pada Susenas 2003 dan 2009 yang masih menggunakan konsep kecacatan, sedangkan Susenas 2006 dan 2012 telah memasukkan konsep disabilitas. Walaupun demikian, perbandingan antara Susenas 2003 dan 2009 dan Susenas 2006 dan 2012 terjadi peningkatan prevalensi. Berdasarkan data Susenas tahun

2012 penyandang disabilitas adalah penyandang lebih dari satu jenis keterbatasan yaitu sebesar 39,97%, diikuti keterbatasan melihat, dan berjalan atau naik tangga.

Kesulitan fungsional adalah ketidakmampuan seseorang untuk melakukan aktivitas normal sehari-hari. Ada lima kesulitan fungsional yang dicakup dalam Sensur Penduduk (SP) 2010 yaitu kesulitan melihat sebesar 29,63%; kesulitan mendengar sebesar 7,87%; kesulitan berjalan sebesar 10,26%; kesulitan mengingat atau berkonsentrasi sebesar 6,70%; serta kesulitan mengurus diri sebesar 2,83%. Diantara penyandang disabilitas tersebut 10,26% diantaranya mengalami keterbatasan berjalan yang mana harus menggunakan kursi roda sebagai alat bantu jalan. Kemampuan mereka dalam menggerakkan beberapa anggota tubuh menjadi terbatas dan membutuhkan bantuan orang lain untuk setiap aktivitasnya.

Kursi roda yang sering digunakan oleh pasien rumah sakit maupun orang yang mempunyai kekurangan fisik, kebanyakan masih menggunakan sistem yang manual. Kursi roda manual dalam pengoperasiannya dibutuhkan seseorang untuk menggerakkan kursi roda, akan tetapi hal ini bagi pengguna yang memiliki keterbatasan untuk menggerakkan kursi roda tidak dapat melakukannya. Idealnya kursi roda yang dibutuhkan adalah pengguna merasa bebas dapat mengatur gerakan kursi roda sesuai kehendaknya. Kenyatannya hal tersebut tidak dapat dilakukan oleh pengguna yang memiliki keterbatasan pada bagian tangan, sehingga membutuhkan alat bantu untuk dapat menggerakkan kursi roda tanpa tergantung orang lain dan tetap aman. Maka diperlukan sebuah alat yang dapat

membantu mobilitas para pengguna kursi roda berupa kursi roda otomatis yang dapat dengan mudah dikendalikan oleh pengguna itu sendiri.

Berdasarkan permasalahan di atas, penulis mendesain suatu simulator kursi roda otomatis dengan menggunakan sensor *flex* yang membantu mempermudah aktivitas pengguna kursi roda dengan menggunakan gerakan jari-jari tangan, sehingga pengguna dapat leluasa bergerak sesuai dengan keinginannya. Kursi roda dirancang dengan menggunakan lima buah sensor *flex* yang akan dipasangkan dalam sarung tangan yang akan dipakai oleh pengguna. Gerakan tangan ini secara tidak langsung melatih koordinasi syaraf otak agar bekerja dengan seimbang.

Berdasarkan uraian tersebut, maka penulis memberikan gagasan berupa solusi untuk mengatasi masalah tersebut dengan merancang kursi roda otomatis dengan judul “Simulator Kursi Roda Otomatis Dengan Menggunakan Sensor *Flex* Berbasis Mikrokontroler”. Melalui pembuatan alat ini, penulis berharap pada akhirnya alat ini akan bisa membantu pengguna kursi roda agar dapat bergerak sesuai dengan instruksi gerakan jari-jari tangannya.

## **B. Identifikasi Masalah**

Berdasarkan uraian latar belakang masalah di atas dapat diidentifikasi beberapa permasalahan yang berhubungan dengan simulator kursi roda otomatis dengan sensor *flex* sebagai berikut:

1. Peningkatan jumlah penyandang disabilitas yang mengalami keterbatasan berjalan dan harus menggunakan kursi roda.

2. Pengguna kursi roda yang memiliki keterbatasan di bagian tangan mengalami kesulitan untuk menjalankan kursi roda manual.
3. Belum adanya perancangan sistem pengendali kursi roda dengan menggunakan sensor *flex*.
4. Belum adannya pengukuran kinerja kursi roda otomatis dengan sensor *flex*.

### **C. Batasan Masalah**

Berdasarkan masalah yang telah dikemukakan dalam identifikasi masalah di atas, maka perlu adanya batasan masalah untuk membatasi ruang lingkup pembahasan proyek akhir. Ruang lingkup batasan proyek akhir ini adalah belum adanya kursi roda otomatis yang geraknya dikontrol menggunakan sensor *flex* yang digunakan pengguna dengan cara memakai sebuah sarung tangan khusus.

### **D. Rumusan Masalah**

Berdasarkan latar belakang di atas, maka dapat dirumuskan beberapa permasalahan sebagai berikut:

1. Bagaimana merancang simulator kursi roda otomatis dengan sensor *flex*?
2. Bagaimana merancang perangkat lunak sebagai sistem yang akan bekerja pada simulator kursi roda otomatis dengan sensor *flex*?
3. Bagaimana unjuk kerja simulator kursi roda otomatis dengan sensor *flex*?

### **E. Tujuan**

Tujuan dari penelitian ini adalah:

1. Merealisasikan rancangan perangkat keras dari kursi roda otomatis dengan sensor *flex*

2. Merealisasikan perangkat lunak yang akan dimasukkan ke dalam mikrokontroler sebagai pengendali utama sistem ini
3. Mengetahui unjuk kerja kursi roda otomatis dengan sensor *flex* yang akan menghasilkan beberapa spesifikasi yang ada pada alat ini.

## F. Manfaat

Manfaat yang diperoleh pada pembuatan alat ini adalah sebagai berikut:

1. Bagi Mahasiswa
  - a. Meningkatkan kreativitas mahasiswa dalam bidang teknologi
  - b. Meningkatkan kemampuan mahasiswa dalam merancang sebuah alat
  - c. Mengetahui cara merancang kursi roda otomatis dengan sensor *flex*
  - d. Mengetahui karakteristik dan cara kerja dari sensor *flex* pada kursi roda otomatis.
2. Bagi Pengguna Kursi Roda
  - a. Mempermudah gerakan pengguna kursi roda, karena pengguna dapat menggerakkan kursi roda sesuai dengan instruksi pengguna
  - b. Membantu pengguna kursi roda untuk terapi, karena dengan menggerakkan jari tangan dengan suatu gerakan yang membutuhkan koordinasi tubuh dan koordinasi syaraf khususnya otak. Hal ini dapat membantu proses penyembuhan pada pengguna kursi roda, karena otot jari tangan saling berkoordinasi.

## G. Keaslian Gagasan

Proyek akhir dengan judul “Simulator Kursi Roda Otomatis dengan Sensor *Flex* Berbasis Mikrokontroler” merupakan gagasan pribadi penulis yang

terinspirasi dari cara pengendalian kursi roda yang masih manual dengan harapan dapat mempermudah pengguna kursi roda dalam menggerakkan kursi rodanya. Adapun karya-karya sejenis yang berkaitan dengan proyek akhir ini adalah:

1. Tugas Akhir Rancang Bangun Kursi Roda Elektrik Menggunakan Perintah Suara Berbasis Android Oleh Liem, Yuliana Kathina Hatta, Pujiono, S.T., M.T., dan Ir. Tasripan, M.T. dari Institut Teknologi Sepuluh Nopember. Alat ini memerlukan gerakan yang harus lebih diperhalus, perlu peningkatan *filter* dalam penyaringan suara. Perbedaan dari karya yang penulis buat dengan karya milik Liem, Yuliana Kathina Hatta adalah sebagai berikut:
  - a. Menggunakan mikrokontroler AT Mega 16 sebagai pusat pengolah data sedangkan alat ini menggunakan arduino uno sebagai pusat pengolah data dan masukan dari sensor.
  - b. Menggunakan teknologi *speech recognition* untuk mengendalikan kursi roda sedangkan alat ini menggunakan sensor *flex* untuk mendeteksi kelengkungan jari yang akan mengintruksikan gerakan kursi roda sehingga gerakannya lebih halus
2. Tugas Akhir Kursi Roda Elektrik Menggunakan *Joystick* Berbasis Mikrokontroler AT Mega 8535 Oleh Okta Dwika, Riski Widodo dari Universitas Negeri Jakarta. Alat ini hanya dapat bergerak mengikuti instruksi *input joystick* yang akan menggerakkan motor DC sebagai aktuator dan mikrokontroler AT Mega 8535 sebagai pengendali.

Perbedaan dari karya yang penulis buat dengan karya milik Okta Dwika, Riski Widodo adalah sebagai berikut:

- a. Menggunakan mikrokontroler AT Mega 8535 sebagai pusat pengolah data sedangkan alat ini menggunakan arduino uno sebagai pusat pengolah data dan masukan dari sensor.
- b. Menggunakan *IC Driver Motor EMS* sedangkan alat ini menggunakan Modul L298N *driver* motor DC *Dual H-Bridge* sebagai pengatur gerakan motor DC
- c. Pengontrolan gerakan kursi roda menggunakan *joystick* sedangkan alat ini dalam pengontrolan menggunakan sensor *flex* untuk mengontrol gerakan kursi roda
- d. Menggunakan *lead-acid battery accu/ aki* sebagai *power supply* sedangkan alat ini menggunakan baterai Li-Po sebagai catu daya untuk menghidupkan *driver* motor L298N.

## **BAB II**

### **PENDEKATAN PEMECAHAN MASALAH**

#### **A. Simulator**

Simulasi berasal dari kata simulate yang artinya berpura-pura atau berbuat seakan-akan. Simulasi dapat diartikan cara penyajian pengalaman belajar dengan menggunakan situasi tiruan untuk memahami tentang konsep, prinsip, atau keterampilan tertentu (Sanjaya, 2008: 159). Simulator dalam (Depdiknas, 2005) adalah program yang berfungsi untuk menyimpulkan suatu peralatan, tetapi kerjanya agak lambat daripada keadaan yang sebenarnya.

Simulator program berfungsi untuk menyimulasikan suatu peralatan, tetapi kerjanya lebih lambat daripada keadaan yang sebenarnya; alat untuk melakukan simulasi; alat yang dapat menyimulasikan (KBBI, 2017). Secara sederhana simulator adalah sesuatu yang dibuat menyerupai bentuk nyata pada umumnya, tetapi dalam bentuk dan ukuran yang lebih sederhana.

#### **B. Disabilitas**

Disabilitas adalah seseorang yang termasuk ke dalam penyandang cacat fisik, penyandang cacat mental ataupun gabungan penyandang cacat fisik dan mental (Undang-Undang Republik Indonesia Nomor 4 Tahun 1997 tentang Penyandang Cacat). Disabilitas adalah kondisi atau fungsi dari seorang individu yang dinilai secara signifikan relatif terganggu dari standar biasa individu dari kelompok mereka. Terdapat beberapa macam disabilitas. Konsep ini sering digunakan untuk

merujuk kepada fungsi individu, termasuk di dalamnya adalah gangguan fisik, gangguan sensorik, gangguan kognitif, gangguan intelektual, penyakit mental, dan beberapa jenis penyakit kronis.

Disabilitas merupakan istilah umum, yang meliputi gangguan, keterbatasan aktivitas, dan pembatasan partisipasi. Penurunan nilai adalah masalah dalam fungsi tubuh atau struktur, pembatasan kegiatan adalah kesulitan yang dihadapi oleh individu dalam melaksanakan tugas atau tindakan; sementara pembatasan partisipasi adalah masalah yang dialami oleh seorang individu dalam keterlibatan situasi kehidupan.

### C. Kursi Roda

Kursi roda merupakan salah satu alat bantu bagi penyandang cacat kaki dan orang yang fisiknya lemah untuk dapat berpindah dari satu tempat ke tempat lain. Kursi roda digunakan untuk meningkatkan kemampuan mobilitas bagi orang yang memiliki kekurangan seperti orang yang cacat fisik, pasien rumah sakit yang tidak diperbolehkan untuk melakukan banyak aktivitas fisik, seorang yang lanjut usia, dan orang-orang yang memiliki risiko tinggi untuk terluka bila berjalan sendiri. Ada beberapa jenis kursi roda, yaitu:

#### 1. Kursi Roda Manual

Kursi roda manual merupakan kursi roda yang biasa digerakkan dengan tangan penggunanya, dapat juga dioperasikan dengan bantuan orang lain dengan cara didorong. Kursi roda manual merupakan kursi roda yang dipakai dengan tangan namun tidak dapat dioperasikan oleh pasien yang juga mengalami cacat di bagian tangannya. Kursi roda manual merupakan kursi roda yang bisa digunakan

untuk berbagai aktivitas sehari-hari. Kursi roda manual ditunjukkan pada Gambar 1.



Gambar 1. Kursi Roda Manual

(Farastya, 2017)

## 2. Kursi Roda Elektrik

Kursi roda elektrik dilengkapi dengan sistem pengendali elektronik untuk memudahkan mobilitas penggunanya, sehingga mereka tidak perlu mengendalikan kursi roda secara manual dengan menggerakkan roda menggunakan tangan. Kursi roda elektrik sebenarnya lebih diperuntukkan bagi mereka yang kemampuan motoriknya tidak memadai untuk menggunakan kursi roda manual ataupun yang mengalami masalah pada jantung, pernapasan, atau kemampuan motorik yang terbatas. Kursi roda elektrik cara penggunaannya dengan memanfaatkan tuas yang terletak pada bagian tangan. Tuas dapat digunakan untuk menjalankan kursi roda (maju atau mundur), mengubah arah kursi roda (belok kanan atau kiri), serta menghentikan kursi roda ketika sedang berjalan. Kursi roda elektrik ditunjukkan pada Gambar 2.



Gambar 2. Kursi Roda Elektrik

### 3. Kursi Roda Untuk Olahraga

Memiliki kekurangan atau ketidakmampuan berjalan bukan penghalang untuk berolah raga, hal inilah yang menginspirasi para produsen alat kesehatan untuk menciptakan kursi roda yang dapat digunakan untuk berolahraga. Kursi roda ini didesain sedemikian rupa sehingga penggunanya dapat digunakan untuk berolahraga dengan baik. Beberapa olahraga yang dapat dilakukan menggunakan jenis kursi roda ini adalah balap lintasan, basket, tenis dan lain sebagainya.

Kursi roda olahraga ditunjukkan pada Gambar 3.



Gambar 3. Kursi Roda Olahraga

#### 4. Kursi Roda Berdiri

Kursi roda berdiri didesain sedemikian rupa sehingga pengguna dapat terangkat pada posisi yang lebih tinggi. Mekanisme pengangkatan dilakukan dengan menggunakan sistem hidrolik. Kursi roda yang dapat berdiri digunakan sebagai alat latihan atau terapi berdiri. Adanya kursi roda yang dapat berdiri juga sebenarnya berguna untuk mencegah efek-efek negatif akibat duduk yang terlalu lama seperti iritasi kulit, otot kaku dan lemah akibat jarang digunakan, pegal dan ngilu, serta kebas akibat sirkulasi darah tidak lancar.

Kursi roda dapat berdiri ditunjukkan pada Gambar 4.



Gambar 4. Kursi Roda Berdiri

##### 5. Kursi Roda *Bariatric*

Kursi roda ini dirancang bagi mereka yang mengalami obesitas. Jika umumnya kursi roda biasa hanya bisa menahan beban antara 125 hingga 150 kg, maka kursi roda *bariatric* dapat menahan beban hingga 500 kg, seperti ditunjukkan pada Gambar 5.



Gambar 5. Kursi Roda *Bariatric*

## 6. Kursi Roda *Pediatric*

Kursi roda ini sengaja didesain dengan ukuran yang lebih kecil, sandaran maupun dudukan juga dapat diatur sesuai dengan kebutuhan. Kursi roda ini khusus didesain untuk digunakan oleh anak-anak, terutama yang mengidap *cerebral palsy*, seperti ditunjukkan pada Gambar 6.



Gambar 6. Kursi Roda *Pediatric*

## 7. Kursi Roda *Travelling*

Kursi roda jenis ini memang unik karena sengaja dirancang untuk membantu para penyandang disabilitas yang gemar berpergian. Kursi roda travelling dapat digunakan oleh siapa saja dan segala usia, mulai dari anak-anak, dewasa, hingga lanjut usia serta dapat digunakan untuk segala kegiatan baik *indoor* maupun *outdoor*. Untuk membawanya sangat mudah karena kursi roda ini ukurannya kecil, sangat ringan, serta dapat dilipat dengan mudah sehingga hanya perlu dimasukkan ke dalam tas khusus untuk kursi roda, seperti ditunjukkan pada Gambar 7.



Gambar 7. Kursi Roda *Travelling*

## D. Sensor Ultrasonik HC-SR04

### 1. Pengertian Sensor Ultrasonik

Sensor ultrasonik adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara dan digunakan untuk mendeteksi keberadaan suatu objek tertentu di depannya, frekuensi kerjanya pada daerah diatas gelombang suara dari 40 KHz hingga 400 KHz.

Sensor ultrasonik terdiri dari dua unit, yaitu unit pemancar dan unit penerima. Struktur unit pemancar dan penerima sangatlah sederhana, sebuah kristal *piezoelectric* dihubungkan dengan mekanik jangkar dan hanya dihubungkan dengan diafragma penggetar. Tegangan bolak-balik yang memiliki frekuensi kerja 40 KHz hingga 400 KHz diberikan pada plat logam. Struktur atom dari kristal *piezoelectric* akan berkontraksi (mengikat), mengembang atau menyusut terhadap polaritas tegangan yang diberikan, dan ini disebut dengan efek *piezoelectric*.

Sifat dari gelombang ultrasonik yang melalui medium menyebabkan getaran partikel dengan medium amplitudo sama dengan arah rambat longitudinal sehingga menghasilkan partikel medium yang membentuk suatu rapatan atau biasa disebut *strain* dan tegangan yang biasa disebut *strees*. Proses lanjut yang menyebabkan terjadinya rapatan dan regangan di dalam medium disebabkan oleh getaran partikel secara periodik selama gelombang ultrasonik lainnya. Gelombang ultrasonik merambat melalui udara dengan kecepatan 344 meter per detik, mengenai objek dan memantul kembali ke sensor ultrasonik. Gelombang ultrasonik hanya bisa didengar oleh makhluk tertentu seperti kelelawar dan ikan paus. Kelelawar menggunakan gelombang ultrasonik untuk berburu di malam hari sementara paus menggunakannya untuk berenang di kedalaman laut yang gelap (Oktarima, 2013).

## 2. Bagian-bagian dari Sensor Ultrasonik

### a. Pemancar Ultrasonik (*Transmitter*)

Pemancar Ultrasonik ini berupa rangkaian yang memancarkan sinyal sinusoidal berfrekuensi di atas 20 KHz menggunakan sebuah transduser

transmitter ultrasonik.

b. Penerima Ultrasonik (*Receiver*)

Penerima Ultrasonik ini akan menerima sinyal ultrasonik yang dipancarkan oleh pemancar ultrasonik dengan karakteristik frekuensi yang sesuai. Sinyal yang diterima tersebut akan melalui proses filterisasi frekuensi dengan menggunakan rangkaian *band pass filter* (penyaring pelewatan pita), dengan nilai frekuensi yang dilewatkan telah ditentukan. Sinyal keluarannya akan dikuatkan dan dilewatkan ke rangkaian komparator (pembanding) dengan tegangan referensi ditentukan berdasarkan tegangan keluaran penguat pada saat jarak antara sensor kendaraan mini dengan sekat/dinding pembatas mencapai jarak minimum untuk berbelok arah, dapat dianggap keluaran komparator pada kondisi ini adalah *high* (logika '1') sedangkan jarak yang lebih jauh adalah *low* (logika '0'). Logika-logika biner ini kemudian diteruskan ke rangkaian pengendali (mikrokontroler).

Jadi ketika ada sinyal ultrasonik yang masuk ke rangkaian, maka pada komparator akan mengeluarkan logika rendah (0 V) yang kemudian akan diproses oleh mikrokontroler untuk menghitung jaraknya.

### c. Prinsip Kerja Sensor Ultrasonik

HC-SR04 merupakan sensor ultrasonik yang dapat digunakan untuk mengukur jarak antara penghalang dan sensor. Konfigurasi pin dan tampilan sensor HC-SR04 diperlihatkan pada Gambar 8.

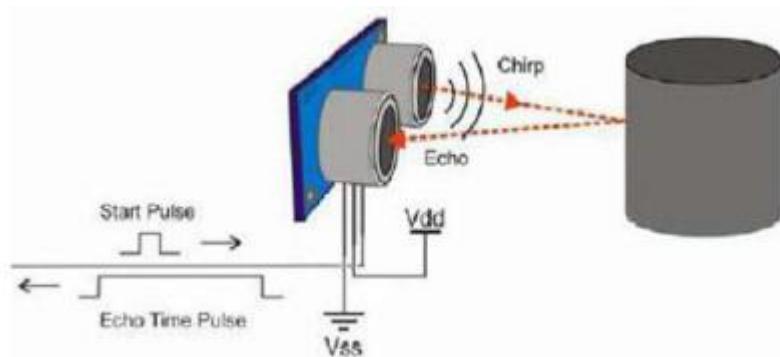


Gambar 8. Konfigurasi Pin dan Tampilan Sensor

Ultrasonik HC-SR04

(HC-SR04 Ultrasonic Range Finder, 2011)

HC-SR04 memiliki 2 komponen utama sebagai penyusunnya yaitu *ultrasonic transmitter* dan *ultrasonic receiver*. Fungsi dari *ultrasonic transmitter* adalah memancarkan gelombang ultrasonik dengan frekuensi 40 KHz kemudian *ultrasonic receiver* menangkap hasil pantulan gelombang ultrasonik yang mengenai suatu objek. Waktu tempuh gelombang ultrasonik dari pemancar hingga sampai ke penerima sebanding dengan 2 kali jarak antara sensor dan bidang pantul seperti yang diperlihatkan pada Gambar 9.



Gambar 9. Prinsip Kerja HC-SR04

Prinsip pengukuran jarak menggunakan sensor ultrasonik HC-SR04 adalah, ketika pulsa *trigger* diberikan pada sensor, *transmitter* akan mulai memancarkan gelombang ultrasonik, pada saat yang sama sensor akan menghasilkan *output* TTL transisi naik menandakan sensor mulai menghitung waktu pengukuran, setelah *receiver* menerima pantulan yang dihasilkan oleh suatu objek maka pengukuran waktu akan dihentikan dengan menghasilkan *output* TTL transisi turun. Jika waktu pengukuran adalah t dan kecepatan suara adalah 340 m/s, maka jarak antara sensor dengan objek dapat dihitung dengan menggunakan Persamaan 1.

$$s = tx \cdot 340 / 2 \quad (1)$$

Dimana :

s = Jarak antara sensor dengan objek (m)

t = Waktu tempuh gelombang ultrasonik dari *transmitter* ke *receiver* (s)

Pemilihan HC-SR04 sebagai sensor jarak yang akan digunakan pada penelitian ini karena memiliki fitur sebagai berikut; kinerja yang stabil, pengukuran jarak yang akurat dengan ketelitian 0,3 cm, pengukuran

maksimum dapat mencapai 4 meter dengan jarak minimum 2 cm, ukuran yang ringkas dan dapat beroperasi pada level tegangan TTL.

Prinsip pengoperasian sensor ultrasonik HC-SR04 adalah sebagai berikut; awali dengan memberikan pulsa *Low* (0) ketika modul mulai dioperasikan, kemudian berikan pulsa *High* (1) pada *trigger* selama 10  $\mu\text{s}$  sehingga modul mulai memancarkan 8 gelombang kotak dengan frekuensi 40 KHz, tunggu hingga transisi naik terjadi pada *output* dan mulai perhitungan waktu hingga transisi turun terjadi, setelah itu gunakan Persamaan 1 untuk mengukur jarak antara sensor dengan objek (Sarwoko, 2011).

## E. Arduino Uno R3

### 1. Mikrokontroler Arduino Uno R3

Arduino uno adalah *board* mikrokontroler berbasis ATmega328 (*datasheet*). Memiliki 14 *pin input* dari *output* digital dimana 6 *pin input* tersebut dapat digunakan sebagai *output* PWM dan 6 *pin input* analog, 16 MHz osilator kristal, koneksi USB, *jack power*, ICSP *header*, dan tombol *reset*. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan *board* arduino uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang ke adaptor DC atau baterai untuk menjalankannya.

Arduino uno R3 berbeda dengan semua board sebelumnya karena arduino uno R3 ini tidak menggunakan *chip driver* FTDI *USB-to-serial*. Melainkan menggunakan fitur dari ATMega 16U2 yang diprogram

sebagai konverter *USB-to-serial* (Arduino, 2009).

## 2. *Board* Arduino Uno

*Board* arduino uno memiliki fitur–fitur baru sebagai berikut :

- a. *Pin out* : menambahkan SDA dan SCL *pin* yang dekat dengan *pin* aref dan dua *pin* baru lainnya ditempatkan dekat dengan *pin* RESET, dengan I/O REF yang memungkinkan sebagai *buffer* untuk beradaptasi dengan tegangan yang disediakan dari *board* sistem. Pengembangannya, sistem akan lebih kompatibel dengan prosesor yang menggunakan AVR, yang beroperasi dengan 5 V dan dengan arduino karena beroperasi dengan 3,3 V. Yang kedua adalah *pin* yang tidak terhubung, yang disediakan untuk tujuan pengembangannya. *Board* arduino ditunjukkan seperti Gambar 10.



Gambar 10. Mikrokontroler Arduino Uno R3

- b. Sirkuit *reset*
- c. ATMega 16U2 ganti 8U yang digunakan sebagai konverter *USB-to-serial*.

Ringkasan spesifikasi arduino uno:

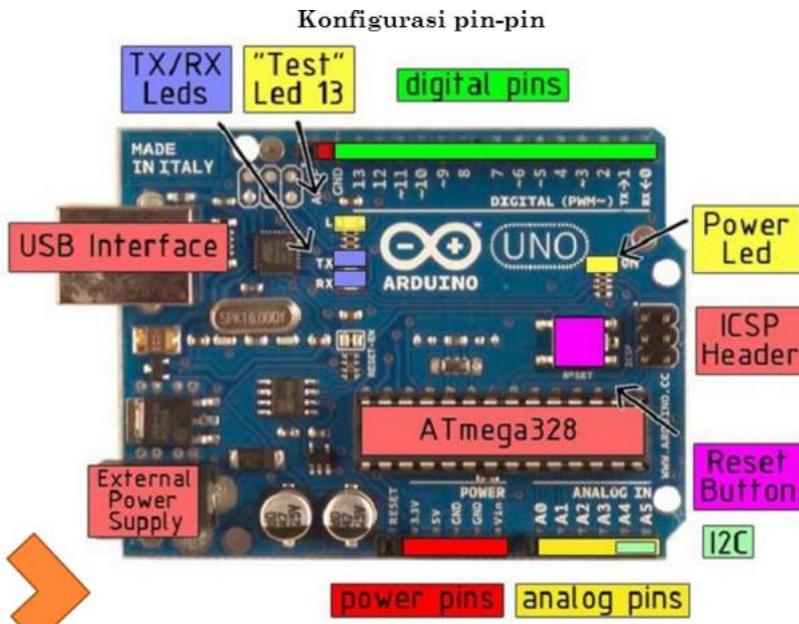
- 1) Mikrokontroler : ATmega328
- 2) Operasi tegangan : 5 Volt
- 3) Input tegangan : disarankan 7-11 Volt

- 4) *Input* tegangan batas : 6-20 Volt
  - 5) *Pin* I/O digital : 14 (6 bisa untuk PWM)
  - 6) *Pin* Analog : 6
  - 7) Arus DC tiap *pin* I/O : 50 mA
  - 8) Arus DC ketika 3.3 V : 50 mA
  - 9) Memori *flash* : 32 KB (ATmega328) dan 0,5 KB  
digunakan oleh *bootloader*
  - 10) SRAM : 2 KB (ATmega328)
  - 11) EEPROM : 1 KB (ATmega328)
  - 12) Kecepatan *clock* : 16 MHz
2. Catu Daya

Arduino uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber listrik dipilih secara otomatis. Eksternal (*non-USB*) daya dapat berupa baik AC-DC adaptor atau baterai. Adaptor ini dapat dihubungkan dengan cara menghubungkan *plug* pusat-positif 2.1 mm ke dalam *board* colokan listrik. Sedangkan untuk baterai dapat dihubungkan kedalam *header pin* GND dan Vin dari konektor *Power*. Berikut pin *power supply* pada arduino uno:

- a.  $V_{IN}$ : Tegangan *input board* arduino ketika menggunakan sumber daya (5 V dari sambungan USB atau dari sumber regulator lain). Anda dapat mensuplai tegangan pada pin ini, jika suplai tegangan lewat *power jack*, dapat mengakses melalui pin ini.
- b. 5 V: keluaran pin ini telah diatur sebesar 5 V dari regulator pada board. Board dapat disuplai melalui DC jack power (7-12 V). menyuplai tegangan melalui

- pin 5 V atau 3.3 V bypasses regulator, dapat merusak *board*.
- 3.V3: Suplai 3,3 Volt dihasilkan oleh regulator pada *board*, menarik arus maksimum 50 mA.
  - GND: pin *ground*



Gambar 11. Konfigurasi Pin–Pin Arduino Uno

*Board* dapat beroperasi pada pasokan daya dari 6 – 20 Volt. Jika diberikan dengan kurang dari 7 V, bagaimanapun, pin 5 V untuk supply kurang dari 5 Volt dan board mungkin tidak stabil. Jika menggunakan lebih dari 12 V, regulator bisa panas dan merusak *board*. Rentang yang dianjurkan adalah 7 V – 12 V. Selain itu, beberapa pin memiliki fungsi khusus :

- 1) Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirim (TX) data TTL serial. Pin ini terhubung ke pin yang sesuai dari chip ATMega8U2 *USB-to-Serial* TTL
- 2) Eksternal Interupsi: 2 dan 3. Pin ini dapat dapat dikonfigurasi untuk memicu

interupsi pada nilai yang rendah, tepi naik atau jatuh, atau perubahan nilai.

Lihat *attachInterrupt()* fungsi untuk rincian

- 3) PWM: 3, 5, 6, 9, 10, dan 11. Menyediakan 8-bit *output* PWM dengan *analogWrite()* fungsi.
- 4) SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). *Pin* ini mendukung komunikasi SPI menggunakan *library* SPI
- 5) LED: 13. Ada *built-in* LED terhubung ke *pin* digital 13. Ketika *pin* adalah nilai TINGGI. LED menyala, ketika *pin* adalah RENDAH, itu *off*.

Arduino Uno R3 memiliki 6 *input analog* diberi label A0 sampai A5, masing–masing menyediakan 10-bit resolusi (yaitu 1024 nilai yang berbeda). Secara *default* sistem mengukur dari *ground* sampai 5 Volt, meskipun mungkin untuk mengubah ujung atas rentang menggunakan *pin* AREF dan fungsi *analogReference()*. Selain itu, beberapa *pin* memiliki fungsi khusus:

- 1) TWI: A4 atau SDA *pin* dan A5 atau SCL *pin*. Mendukung komunikasi TWI
- 2) AREF: Referensi tegangan untuk *input* analog. Digunakan dengan *analogReference()*
- 3) RESET

### 3. Komunikasi

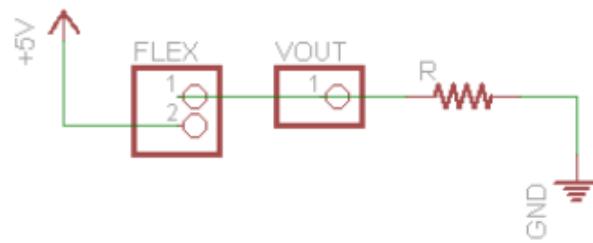
Arduino Uno R3 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, arduino lain, atau mikrokontroler lainnya. ATmega328 menyediakan UART TTL (5 V) komunikasi serial, yang tersedia di *pin* digital 0 (RX) dan 1 (TX). Sebuah ATmega16U2 pada *board* ini komunikasi serial melalui USB dan muncul sebagai *com port virtual* untuk perangkat lunak pada komputer. *Firmware*

'16U2 menggunakan USB *driver* standar COM, dan tidak ada *driver* eksternal yang diperlukan. Namun, pada Windows, file. Inf diperlukan. Perangkat lunak arduino termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke *board* arduino. RX dan TX di *board* LED akan berkedip ketika data sedang dikirim melalui *chip USB-to-serial* dan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada *pin* 0 dan 1). Fungsi ini digunakan untuk melakukan komunikasi *interface* pada sistem. ATmega328 juga mendukung komunikasi I2C (TWI) dan SPI.

#### F. Sensor *Flex*

Sensor *flex* adalah sensor yang memiliki perubahan resistansi akibat adanya perubahan lekukan pada bagian sensor, untuk mendeteksi suatu kelengkungan. Prinsip kerjanya sama seperti potensimeter. Sensor *flex* memiliki 2 kaki pin, dengan bentuk fisik tipis memanjang dan lentur. Sensor ini memiliki *output* berupa resistansi. Dua pin kaki tersebut, jika salah satu pin diberikan tegangan sebesar +5 V maka pin yang lainnya sebagai *output* serta tegangan 0 Volt. Prinsip kerja sensor *flex* ini mirip dengan variabel resistor. Sensor *flex* memberikan resistansi kepada mikrokontroler melalui rangkaian pembagi tegangan. *Output* resistansi ini akan diberikan tegangan yang nantinya akan dibaca oleh mikrokontroler (Ikram, Aisuwarya, Hersyah. 2016).

Mikrokontroler mengkonversi data menggunakan ADC (*analog to digital converter*), dimana data masukkannya didapat dari tegangan yang sudah terkena resistansi. Rangkaianya dapat dilihat pada Gambar 12 (Taufik, 2014: 12).



Gambar 12. Rangkaian Pembagi Tegangan Sensor *Flex*

Hambatan sensor fleksibel ini berubah ketika bantalan logam berada diluar tekukan. Spesifikasi:

Cakupan suhu : -35°C sampai +80°C

Hambatan datar : 10K Ohm

Toleransi hambatan :  $\pm 30\%$

Cakupan hambatan tekukan : 60K Ohm

Nilai *power* : 0,5 Watt dst. 1 Watt sampai batas maksimal

Resistansi *flat* 10Kohm  $\pm 30\%$

Rentang lengkungan kontur antara 45 KOhm – 125 Kohm

Gambar sensor *flex* ditunjukkan seperti Gambar 13.



Gambar 13. Sensor *Flex*

(Sparkfun datasheet, 2014)

## G. Baterai *Li-Po*

### 1. Pengertian Baterai *Li-Po*

Baterai *Lithium Polimer* atau biasa disebut dengan *LiPo* merupakan salah satu jenis baterai yang sering digunakan dalam dunia RC. Bentuk fisik baterai *Li-Po* dapat dilihat pada Gambar 14. Ada tiga kelebihan baterai berjenis *LiPo* daripada baterai jenis lain yaitu:

- a. Baterai *LiPo* memiliki bobot yang ringan dan tersedia dalam berbagai macam bentuk dan ukuran
  - b. Baterai *LiPo* memiliki kapasitas penyimpanan energi listrik yang besar
  - c. Baterai *LiPo* memiliki tingkat *discharge rate* energi yang tinggi, dimana hal ini sangat berguna sekali dalam bidang RC.
2. Kelemahan baterai *LiPo* yaitu:
    - a. Harga baterai *LiPo* masih tergolong mahal
    - b. Baterai *Lithium Polymer* memiliki *life time* yang lebih pendek daripada baterai *NiMH*. Mereka rata-rata hanya 300-400 siklus jika dirawat dengan baik (Afif dan Pratiwi, 2015)
    - c. Alasan keamanan. Baterai *LiPo* menggunakan bahan elektrolit yang mudah terbakar
    - d. Baterai *LiPo* membutuhkan penanganan khusus agar dapat bertahan lama.

*Charging, discharging*, maupun penyimpanan dapat mempengaruhi usia dari baterai jenis ini.



Gambar 14. Baterai *Li-Po*

### 3. Tegangan

Pada baterai *LiPo* memiliki rating 3,7 Volt per sel. Keuntungannya adalah tegangan baterai yang tinggi dapat dicapai dengan menggunakan jumlah sel yang lebih sedikit. Pada setiap paket baterai *LiPo* selain tegangan ada label yang disimbolkan dengan "S". Disini "S" berarti sel yang dimiliki sebuah paket baterai (*battery pack*).

### 4. Kapasitas

Kapasitas baterai menunjukkan seberapa banyak energi yang dapat disimpan oleh sebuah baterai dan diindikasikan dalam *miliampere hours* (mAh).

### 5. *Discharge Rate*

*Discharge rate* biasa disimbolkan dengan "C" merupakan notasi yang menyatakan seberapa cepat sebuah baterai untuk dapat dikosongkan secara aman. Energi listrik pada baterai *LiPo* berasal dari pertukaran ion dari anoda ke katoda. Semakin cepat pertukaran ion yang dapat terjadi maka berarti semakin nilai dari "C". Sebuah baterai dengan discharge rate 10C berarti baterai tersebut dapat di discharge 10 kali dari kapasitas baterai sebenarnya (Musbikhin, 2014).

## 6. Hambatan Dalam

Hambatan dalam adalah bilangan yang menyatakan nilaih tahanan yang ada didalam komponen baterai. Hambatan ini akan menentukan kecepatan pertuakan ion dari anoda ke katoda.

## H. Motor DC

Motor DC (*Direct Current*) adalah peralatan elektromekanik dasar yang berfungsi untuk mengubah tenaga listrik menjadi tenaga mekanik. Motor DC merupakan jenis motor yang menggunakan tegangan searah sebagai sumber tenaganya. Dengan memberikan beda tegangan pada kedua terminal tersebut, motor akan berputar pada satu arah, dan bila polaritas dari tegangan tersebut dibalik maka arah putaran motor akan terbalik pula. Polaritas dari tegangan yang diberikan pada dua terminal menentukan arah putaran motor sedangkan besar dari beda tegangan pada kedua terminal menentukan kecepatan motor (D. Petruzella, 2001). Diperlihatkan pada Gambar 15.



Gambar 15. Motor DC

Motor DC memiliki 3 bagian atau komponen utama untuk dapat berputar sebagai berikut:

1. Kutub medan

Secara sederhana bahwa interaksi dua kutub magnet akan menyebabkan putaran pada motor DC. Motor DC memiliki kutub medan yang stasioner dan dinamo yang menggerakkan *bearing* pada ruang diantara kutub medan. Motor DC sederhana memiliki dua kutub medan yaitu kutub utara dan kutub selatan.

2. Rotor

Bila arus masuk menuju rotor (bagian motor yang bergerak), maka arus ini akan menjadi elektromagnet. Rotor yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, rotor berputar dalam medan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan dinamo.

3. Komutator

Komponen ini terutama ditemukan di dalam motor DC. Kegunaannya adalah untuk membalikkan arah arus listrik dalam dinamo. Komutator juga membantu dalam transmisi arus antara dinamo dan sumber daya (Hamdani, 2010: 9-10).

Keuntungan utama motor DC adalah sebagai pengendali kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur:

- a. Tegangan dinamo, meningkatkan tegangan dinamo akan meningkatkan kecepatan

- b. Arus medan, menurunkan arus medan akan meningkatkan kecepatan.

Prinsip kerja motor DC didasarkan pada prinsip bahwa jika sebuah konduktor yang dialiri arus listrik diletakkan dalam medan magnet, maka tercipta gaya pada konduktor tersebut yang cenderung membuat konduktor berotasi. Arah medan magnet ditentukan oleh arah aliran arus pada konduktor. Berikut ini adalah tabel pemakaian dan sifat motor DC, ditunjukkan dengan Tabel 1.

Tabel 1. Pemakaian dan Sifat Motor DC

Jenis motor	Sifat	Pemakaian
Motor <i>shunt</i>	1. Putaran tetap (30% dari putaran nominal) 2. Torsi awal tidak terlalu tinggi	1. <i>Fan, blower</i> 2. Mesin penggeraan logam ( <i>mesffris</i> ) 3. Penggerak <i>wiper</i> mobil 4. Mesin slep
Motor Seri	1. Putaran bervariasi (mudah diatur) 2. Torsi awal tinggi	1. Traksi (derek) 2. Krane 3. Trem, kereta listrik 4. Kereta bawah tanah
Motor Kompon	Hampir sama dengan motor <i>shunt</i>	Hampir sama dengan motor <i>shunt</i>

## I. Modul L298N Driver Motor DC Dual H-Bridge

*Driver* motor L298N merupakan modul driver motor DC yang paling banyak digunakan untuk mengontrol kecepatan serta arah perputaran motor DC. L298

adalah *driver* motor berbasis H-Bridge, mampu menangani beban hingga 4A pada tegangan 6 V–46 V. Dalam *chip* terdapat dua rangkaian H-Bridge. IC L298 merupakan sebuah IC tipe *H-bridge* yang mampu mengendalikan beban-beban induktif seperti relay, solenoid, motor DC dan motor stepper. Mampu mengeluarkan *output* tegangan untuk motor DC sebesar 50 Volt dan dapat mengendalikan 2 untuk motor DC.

Pengaturan kecepatan motor digunakan teknik PWM (*Pulse with Modulation*) yang diinputkan dari mikrokontroler melalui pin *enable*. PWM untuk kecepatan rotasi yang bervariasi level *high*-nya (Ardiansyah, Hidayatama. 2013).

Spesifikasi:

1. Tipe: *Dual H-Bridge*
2. *Chip* kontrol: ST L298N
3. *Logic voltage*: 5 V DC
4. *Drive voltage*: 5-35 V DC
5. *Logical current*: 0 mA-36 mA
6. *Driving current*: 2 A (*Max. single bridge*)
7. Temperatur: -20 C – 135 C
8. *Power maksimum*: 25 W
9. Berat: 30 g
10. Ukuran: 43 x 43 x 27 mm

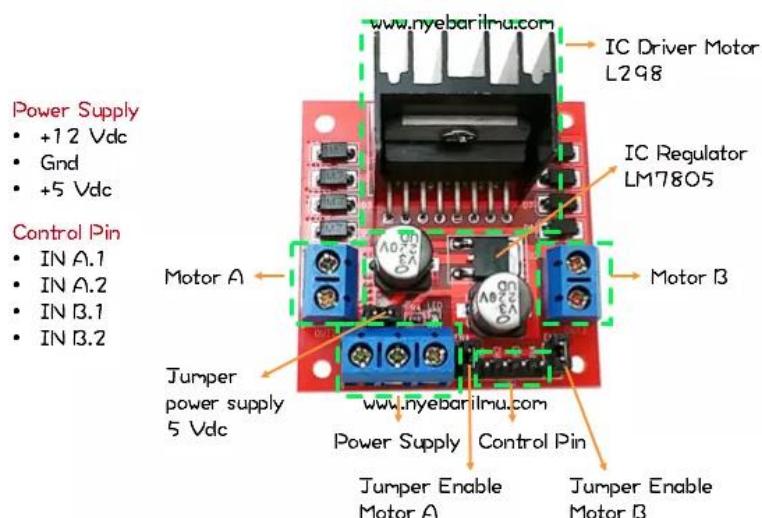
Kelebihan L298N *motor driver* adalah:

1. Lebih presisi dalam mengontrol motor
2. Dapat mengendalikan motor yang besar (maksimal 2 A)
3. Dilengkapi dengan *heatsink* sehingga lebih tahan panas
4. Mudah untuk pemasangannya

Untuk pemasangan *driver* L298N ini dibutuhkan 6 buah pin mikrokontroler.

Dua buah untuk pin *enable* (masing-masing satu untuk tiap motor DC). Empat buah untuk mengatur kecepatan motor DC tersebut. Pada prinsipnya rangkaian *driver* motor L298N ini dapat mengatur tegangan dan arus sehingga kecepatan dan arah motor dapat diatur. Konfigurasi pin pada modul L298N ditunjukkan pada

Gambar 16 :



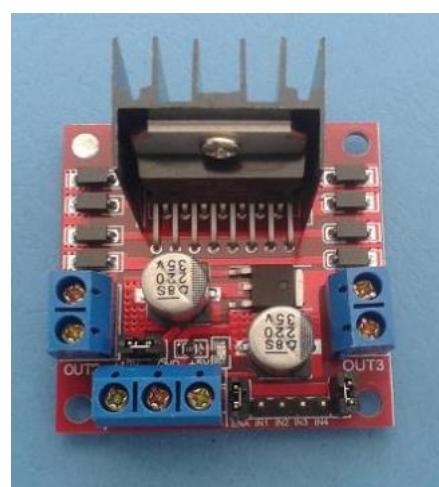
Gambar 16. Konfigurasi Pin pada Modul L298N

(Nyebarilmu, 2017)

*Driver* motor ini pada prinsipnya sama dengan L293D. Cara pengontrolannya pun sama. Perbedaannya mendasar hanya terletak pada karakteristik elektronikanya, yaitu kemampuan L298N dalam melewatkkan arus untuk motor DC lebih besar yaitu sebesar 3 A. Agar mampu bekerja dengan baik maka diperlukan beberapa komponen dioda pendukung. Bentuk modul L298N dapat dilihat pada Gambar 17 dan Gambar 18 (Herdianto, 2017).



Gambar 17. Modul L298N Tampak Belakang



Gambar 18. Modul L298N Tampak Depan

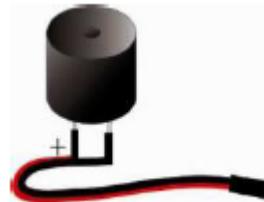
Fitur Modul L298N *Driver Motor DC Dual H-Bridge*:

1. Tegangan operasi 0-46 V
2. Tegangan logic 4,5-7 V
3. Arus 4 A
4. Heatsink untuk membuang panas
5. Regulator 7805 dengan keluaran 5 V
6. Dioda proteksi
7. Mampu mengontrol 2 motor DC

#### **J. *Buzzer***

*Buzzer* adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan *loud speaker*, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. *Buzzer* ini digunakan sebagai indikator (Sulistyowati dan Febriantoro, 2012: 26-27).

Bentuk fisik *buzzer* dapat dilihat pada Gambar 19.



Gambar 19. Bentuk Fisik *Buzzer*

(Hardiyanto, Rochim, dan Windasari, 2015: 186)

## K. Software Arduino IDE

### 1. Pengertian Arduino Software (IDE)

IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman arduino (*sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* arduino dengan mikrokontroler.

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut *Wiring* yang membuat operasi *input* dan *output* menjadi lebih mudah. Arduino IDE ini dikembangkan

dari *software processing* yang dirombak menjadi arduino IDE khusus untuk pemrograman dengan arduino.

## 2. Menulis *Sketch*

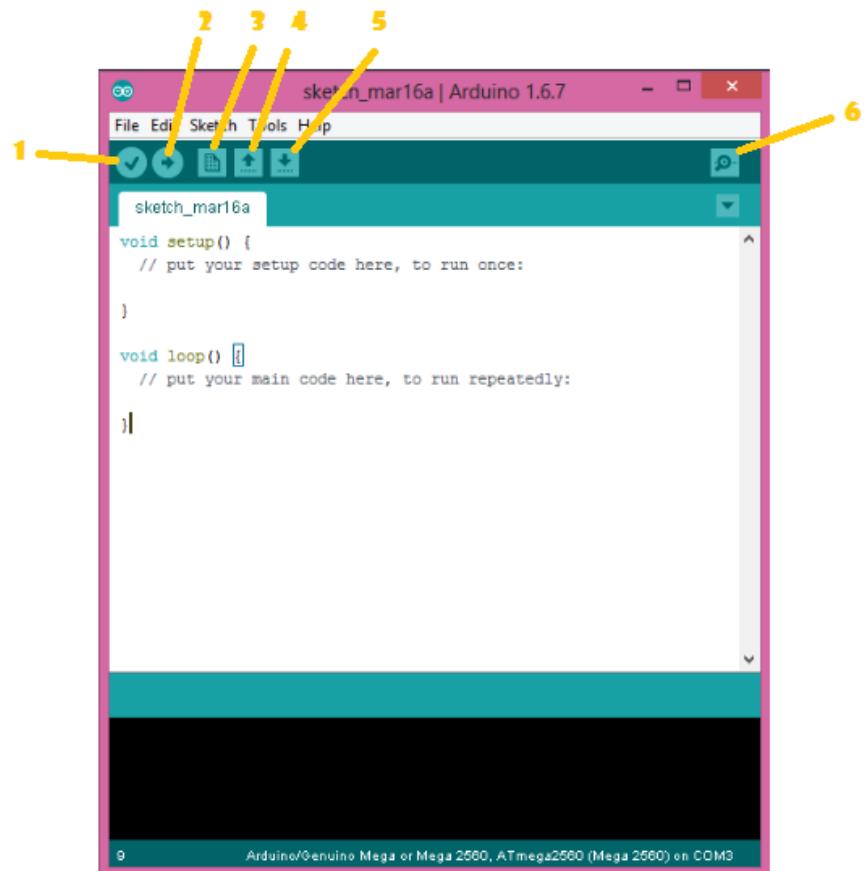
Program yang ditulis dengan menggunakan arduino *software* (IDE) disebut sebagai *sketch*. *Sketch* ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi *.ino*. Teks editor pada arduino *software* memiliki fitur seperti *cutting/paste* dan *searching/replacing* sehingga memudahkan kamu dalam menulis kode program.

Pada *software* arduino IDE, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload program*. Di bagian bawah paling kanan *software* arduino IDE, menunjukan *board* yang terkonfigurasi beserta COM *ports* yang digunakan.

Beberapa fungsi dari arduino IDE yang sering digunakan terletak pada *toolbar*. Terdapat 6 buah tombol pada *toolbar* tersebut yang memiliki fungsi sesuai dengan deskripsi pada Gambar 20 adalah:

- a. *Verify*: berfungsi untuk melakukan *checking* kode yang kamu buat apakah sudah sesuai dengan kaidah pemrograman yang ada atau belum
- b. *Upload*: berfungsi untuk melakukan kompilasi program atau kode yang kamu buat menjadi bahasa yang dapat dipahami arduino
- c. *New*: berfungsi untuk membuat *sketch* baru

- d. *Open*: berfungsi untuk membuka *sketch* yang pernah kamu buat dan membuka kembali untuk dilakukan *editing* atau sekedar *upload* ulang ke arduino.
- e. *Save*: berfungsi untuk menyimpan *sketch* yang telah dibuat.
- f. *Serial Monitor*: *Serial monitor* merupakan jendela yang menampilkan data apa saja yang dikirimkan atau dipertukarkan antara arduino dengan *sketch* pada *port serial*-nya.



Gambar 20. Tampilan *Software Arduino IDE*

## BAB III

### KONSEP PERANCANGAN

#### A. Identifikasi Kebutuhan

Dalam pembuatan proyek akhir “**simulator kursi roda otomatis dengan sensor *flex* berbasis mikrokontroler**” ini dibutuhkan komponen-komponen berikut:

1. Sensor adalah suatu peralatan yang berfungsi untuk mendeteksi gejala-gejala atau sinyal-sinyal yang berasal dari perubahan energi, dibutuhkan sensor yang dapat mendeteksi kelengkungan jari.
2. Sensor jarak untuk mendeteksi adanya jarak antara sensor dan penghalang di depan sensor
3. Pengontrol rangkaian elektronik yang dapat menyimpan program berupa sebuah *chip* mikrokontroler.
4. *Driver* motor DC dan motor DC
5. Sumber tegangan yang dapat menghasilkan beda potensial
6. Kabel penghubung berfungsi untuk menghubungkan mikrokontroler dengan komponen-komponen lainnya.
7. Pipa paralon, T dan siku pipa paralon

## B. Analisis Kebutuhan

Berdasarkan identifikasi kebutuhan di atas, maka diperoleh beberapa analisis kebutuhan terhadap sistem yang akan dirancang adalah sebagai berikut:

1. Sensor yang digunakan untuk mendeteksi kelengkungan jari tangan menggunakan sensor *flex*. Terdapat dua macam sensor *flex* menurut ukuran panjangnya, yaitu sensor *flex* yang memiliki panjang 2,2 inchi dan panjang 4,5 inchi. Pada alat ini dibutuhkan sensor *flex* dengan panjang 2,2 inchi karena ukuran ini lebih ideal untuk panjang jari tangan manusia.
2. Sensor jarak digunakan untuk mengukur jarak antara penghalang dengan sensor. Sensor jarak yang digunakan adalah sensor ultrasonik, karena sensor ultrasonik karena ukuran yang ringkas, memiliki kinerja yang lebih stabil dengan ketelitian pengukuran jarak sebesar 0,3 cm, jarak maksimum jangkauan sensor ini mencapai 4 m dan jarak minimum 2 cm. Terdapat dua jenis sensor ultrasonik, yaitu sensor ultrasonik ping (*parallax*) dan sensor ultrasonik *defantech* (*SRF 04 and SRF05 ranger*). Pada alat ini menggunakan sensor ultrasonik HC-SR04 karena tipe ini lebih efisien. Perbedaan antara SRF05 dan HC-SR04 hanya terletak pada jumlah kaki sensor. SRF05 memiliki satu tambahan kaki yang berfungsi sebagai *pin out*. Namun, umumnya baik SRF05 dan HC-SR04 kaki yang digunakan hanyalah 4 pin saja, yakni *vcc*, *trig*, *echo*, *gnd*.
3. Mikrokontroler sebagai pusat pengolah data dan keputusan pengendali sistem dari data yang dikirimkan oleh *inputan* yang ditangkap oleh sensor *flex*. Terdapat beberapa jenis arduino diantaranya Arduino Uno R3,

Arduino Nano, Arduino Pro Mini dan Arduino Mega, secara spesifikasi teknis tidak terlalu banyak perbedaan antara Arduino UNO R3, Arduino Nano, dan Arduino Pro Mini. Ketiganya memiliki chip mikrokontroler yang sama (ATmega328), jumlah pin yang sama, clock speed yang sama, dan jumlah memori yang sama. Mikrokontroler yang dipilih pada alat ini menggunakan arduino uno R3 karena arduino uno dipilih sebagai pengendali utama karena memiliki 14 buah pin digital, 6 pin analog yang dapat digunakan untuk jalur *input* maupun *output* yang sifatnya dapat di program ulang (*programable*).

4. *Driver* motor DC adalah bagian yang berfungsi untuk menggerakkan motor DC. *Driver* motor L298N pada prinsipnya sama dengan L293D. Cara pengontrolannya pun sama. Perbedaannya mendasar hanya terletak pada karakteristik elektronikanya, yaitu kemampuan L298N dalam melewatkkan arus untuk motor DC lebih besar yaitu sebesar 3 A. Hanya saja, untuk menggunakan IC ini sebagai driver motor, tidak semudah menggunakan IC L293D. Agar mampu bekerja dengan baik maka diperlukan beberapa komponen dioda pendukung. *Driver* motor yang dipilih adalah modul L298N *driver* motor DC *dual H-Bridge* sebagai pengatur dan pengirim perintah pada gerakan motor DC. *Driver* motor DC tipe ini mudah pemasangannya, dapat digunakan untuk mengendalikan dua buah motor DC dengan dua arah putaran dan kecepatan putar motor.

5. Motor DC sebagai *output* berupa gerakan yang dihasilkan oleh Modul L298N *driver* motor DC *dual H-Bridge*.
6. *Power supply* dengan baterai *Li-Po* 12 V 1450 mAh sebagai pemberi tegangan serta arus listrik pada komponen, baterai 9V sebagai sumber tegangan untuk Arduino.
7. Untuk menghubungkan komponen dengan mikrokontroler Arduino Uno membutuhkan kabel penghubung.
8. Pipa paralon, T pipa paralon, siku pipa paralon sebagai kerangka utama simulator kursi roda. Ada berbagai macam ukuran pipa paralon mulai dari 1/2", 3/4", 1", 1 1/4", 1 1/2", 2", 2 1/2", 3", 4", sampai 5". Ukuran yang akan digunakan dalam pembuatan alat ini adalah 5,8" karena ukuran ini paling pas dengan alat yang akan dibuat.

Kebutuhan komponen baik komponen utama maupun komponen penunjang dapat dilihat pada Tabel 2:

Tabel 2. Identifikasi Kebutuhan

No	Rangkaian	Komponen	Spesifikasi
1	<i>Input</i>	Sensor <i>flex</i>	<ul style="list-style-type: none"> <li>- Hambatan datar : 10 KΩ</li> <li>- Tegangan <i>output</i> sensor: 5 V</li> </ul>
		<i>Driver</i> motor DC	Tipe Modul L298N <i>driver</i> motor DC <i>dual H-Bridge</i>
		Sensor Ultrasonik	<ul style="list-style-type: none"> <li>- Tipe HC-SR04</li> <li>- Pengukuran maksimum dapat mencapai 50 cm</li> <li>- Jarak minimum 2 cm</li> </ul>
2	<i>Output</i>	Motor DC	Tegangan 12 V dengan torsi 1000rpm

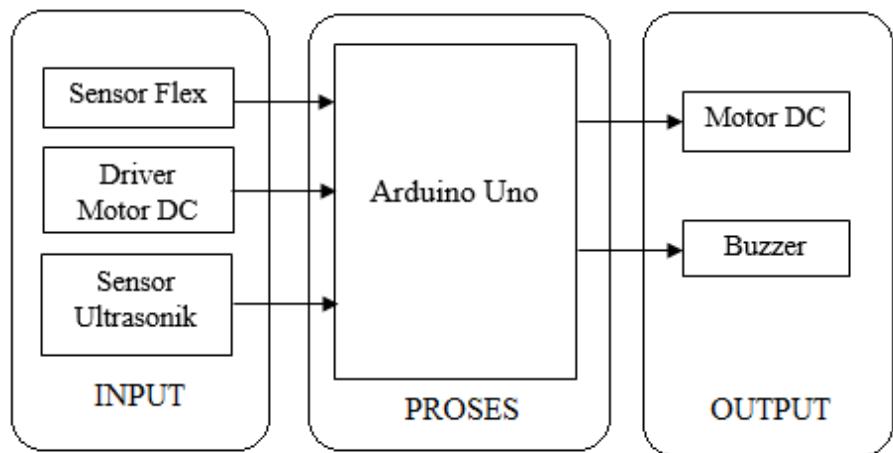
Lanjutan Tabel 2:

No	Rangkaian	Komponen	Spesifikasi
		<i>Buzzer</i>	5 V
3	<i>System Minimum</i>	Mikrokontroler	Arduino Uno R3
4	Catu daya	Baterai	Baterai <i>Li-Po</i> 12 V 1450 mAh
			Daya 9 V
5	Lain-lain	Kabel jumper	Secukupnya
		LED warna merah	Satu buah
		Pipa paralon	5/8 inchi dua buah
		T pipa paralon	5/8 inchi 10 buah
		Siku pipa paralon	5/8 inchi 6 buah
		Gabus asturo	Satu buah
		Resistor	10 KΩ tiga buah
		<i>Pylox</i> warna <i>silver</i>	Satu buah

### C. Perancangan Sistem

Pada tahap ini dilakukan persiapan tentang segala keperluan, baik peralatan maupun bahan yang digunakan. Desain dari kursi roda otomatis berbasis sensor *flex*. Sensor *flex* yang merupakan sensor yang digunakan untuk mendeteksi gerakan jari-jari tangan yang dipasangkan pada jari tangan dilakukan pergerakan sehingga terjadi perubahan tegangan pada sensor *flex* yang dikirimkan menuju arduino dan diproses oleh arduino untuk menggerakan motor DC sesuai dengan besaran dari sensor *flex*. Sedangkan sensor ultrasonik berfungsi untuk menghindari tabrakan pada kursi roda. Sehingga apabila ada rintangan di depan dan di belakang kursi roda ketika berjalan dan jaraknya membahayakan maka kursi roda akan berhenti. Adapun skema rangkaian secara keseluruhan dapat

dilihat pada Lampiran 1, dan secara garis besar akan digambarkan pada blok diagram rangkaian pada Gambar 21.



Gambar 21. Blok Diagram Rangkaian

Dapat dilihat pada Gambar 21, yaitu proses kinerja yang dilakukan pada simulator kursi roda otomatis dengan sensor *flex* berbasis mikrokontroler yaitu:

1. Sensor *flex* sebagai pendekripsi kelengkungan jari tangan
2. Arduino uno sebagai pusat pengolah data dan keputusan pengendali sistem dari data yang dikirimkan oleh *input* yang ditangkap oleh sensor *flex*
3. Modul L298N *driver* motor DC *dual H-Bridge* sebagai pengatur motor DC sehingga motor DC dapat bergerak dengan gerakan belok kanan, belok kiri, maju, mundur
4. Motor DC sebagai penerima sinyal perintah yang dihasilkan oleh Modul L298N *driver* motor DC *dual H-Bridge*. Motor DC akan menerima perintah sesuai dengan kelengkungan jari tangan pengguna kursi roda.

5. Sensor ultrasonik HC-SR04 berfungsi ketika pengguna kursi roda kesulitan untuk mengontrol gerakan kursi roda atau kursi roda yang digunakan akan menabrak medan tertentu
6. *Buzzer* sebagai *output* dari sensor ultrasonik ketika ada penghalang di depan atau di belakang kursi roda.

Berdasarkan Gambar 21 maka dapat dijelaskan bagaimana cara kerja simulator kursi roda otomatis dengan sensor *flex* berbasis mikrokontroler, yaitu lima buah sensor *flex* yang dipasangkan pada sarung tangan yang akan dikenakan oleh pengguna kursi roda. Prinsip kerja sensor membutuhkan rangkaian pembagi tegangan yang akan menjadi masukan dan diolah mikrokontroler arduino uno. *Driver* motor DC digunakan untuk mengontrol kecepatan serta arah putaran motor DC. Sensor ultrasonik sebagai pendekripsi jarak dari alat yang dijalankan dengan halangan yang ada di depan dan di belakangnya. Motor DC akan bekerja ketika sensor *flex* mendekripsi adanya perubahan resistansi pada kelengkungan jari, sesuai dengan jari mana yang akan digerakkan oleh pengguna kursi roda.

## 1. *Hardware*

Perancangan simulator kursi roda otomatis dengan sensor *flex* berbasis mikrokontroler yaitu dengan membuat rangkaian catu daya, rancangan sensor *flex*, rancangan sensor ultrasonik, rancangan modul L298N dengan 2 buah motor DC.

### a. Rangkaian catu daya

Sebuah rangkaian elektronika semestinya harus di *supply* oleh tegangan *Direct Current* (DC) yang stabil agar dapat bekerja dengan baik. Diperlukan

sumber tegangan dari baterai *Li-Po* sebagai catu daya yang dibutuhkan untuk mengaktifkan *driver* motor DC sehingga motor DC dapat menggerakkan kursi roda. Untuk mengaktifkan arduino uno dibutuhkan sumber tegangan dari baterai 9 V. Arduino uno sebagai pusat pengolah data dari semua komponen simulator kursi roda otomatis dengan sensor *flex*.

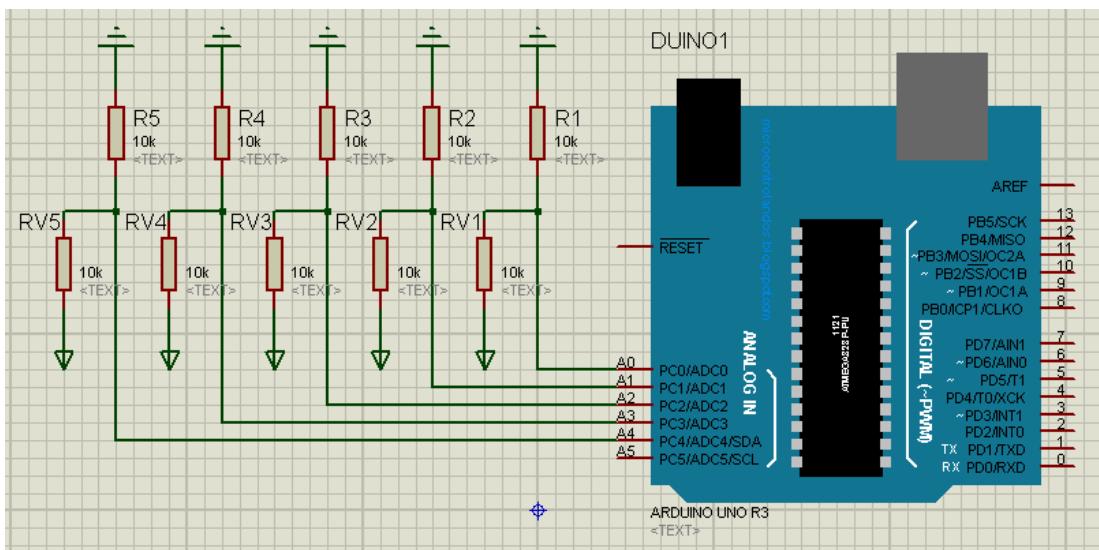
b. Rancangan Sensor *Flex*

Rangkaian simulator kursi roda otomatis dengan sensor *flex* ini mempunyai dua kaki. Salah satu kaki pin diberikan tegangan +5 Volt, sedangkan kaki pin yang lain secara urut terhubung sebagai *output* data pada pin A0, A1, A2, A3, A4; resistor  $1K\Omega$ , serta tegangan 0 Volt. Sensor *flex* akan membeberikan resistansi kepada mikrokontroler melalui rangkaian pembagi tegangan.

Tegangan keluaran arduino yang digunakan sebesar 5 V akan melewati rangkaian pembagi tegangan sehingga tegangan keluaran arduino akan dibagi menjadi tegangan *output* yang sebanding dengan resistansi yang dihasilkan sensor *flex*. Fungsinya adalah untuk membagi tegangan keluaran arduino dengan tegangan *output* sensor *flex* ke data (pin analog *input* arduino). Tegangan *output* sensor *flex* ke data (pin analog *input* arduino) dapat bervariasi dengan menggunakan prinsip pembagi tegangan yaitu besarnya resistansi sensor *flex* yang terukur dibagi dengan resistor pembagi tegangan yang digunakan ditambah dengan tegangan *output* dari sensor *flex* dikalikan tegangan keluaran arduino.

Pada Gambar 22 ditunjukkan pembagi tegangan adalah resistor  $10 K\Omega$  karena menyesuaikan dengan hambatan datar sensor *flex* yang tertera di *datasheet* pada

Lampiran 6. Sensor *flex* memiliki dua kaki pin yaitu resistor pembagi tegangan yang terhubung dengan data masukan pin analog arduino dan diberikan tegangan vcc, sedangkan kaki pin yang lain sebagai tegangan *output* yang terhubung dengan *ground*. Rangkaian masing-masing sensor *flex* dijadikan satu dan dihubungkan dengan masing-masing kabel yang sudah diselubungi sebuah kabel telepon. Bagian pangkal kabel telepon akan terbagi menjadi beberapa kabel, masing-masing kabel akan dihubungkan pada pin analog arduino uno, *ground* dan vcc. Mikrokontroler mengkonversi data menggunakan ADC, dimana data masukannya didapat dari tegangan yang sudah terkena resistansi.

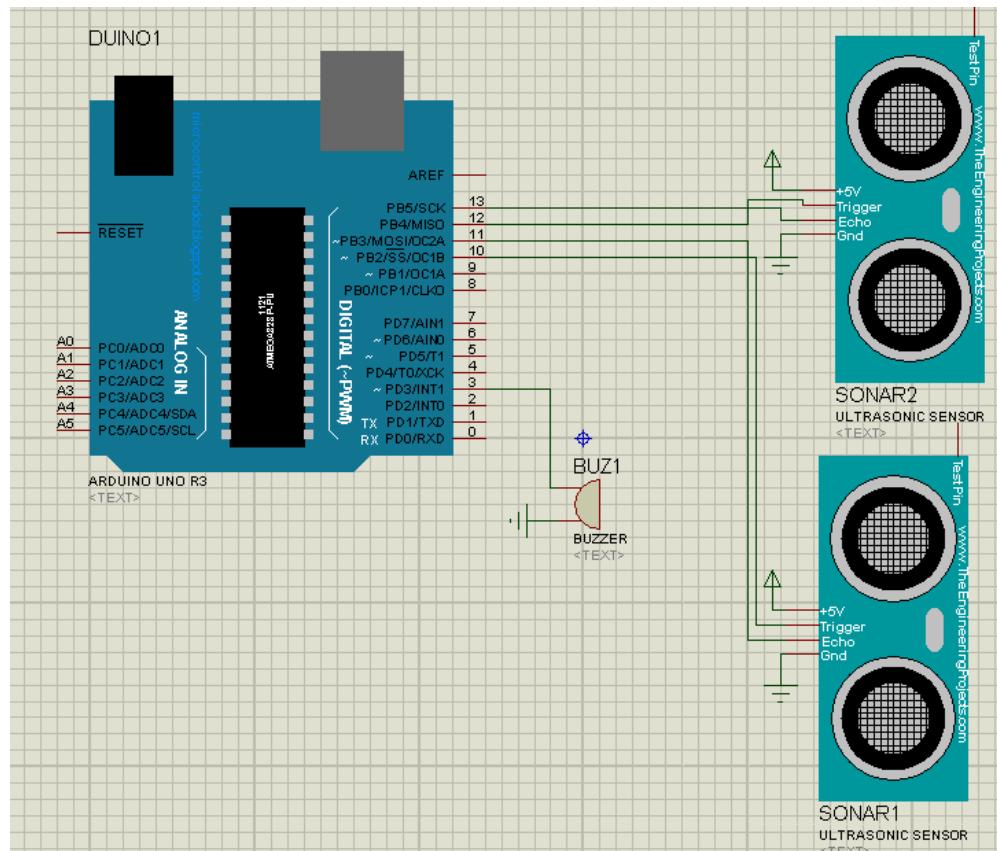


Gambar 22. Rangkaian Sensor *Flex*

### c. Rancangan Sensor Ultrasonik

Sensor ini hanya memerlukan 2 pin I/O untuk berkomunikasi dengan mikrokontroler, yaitu *trigger* dan *echo*. Pin *trigger* dan *echo* dihubungkan ke mikrokontroler. Mikrokontroler akan mengeluarkan *output high* pada pin *trigger* selama minimal  $10 \mu\text{s}$  saat memancarkan gelombang ultrasonik ke penghalang.

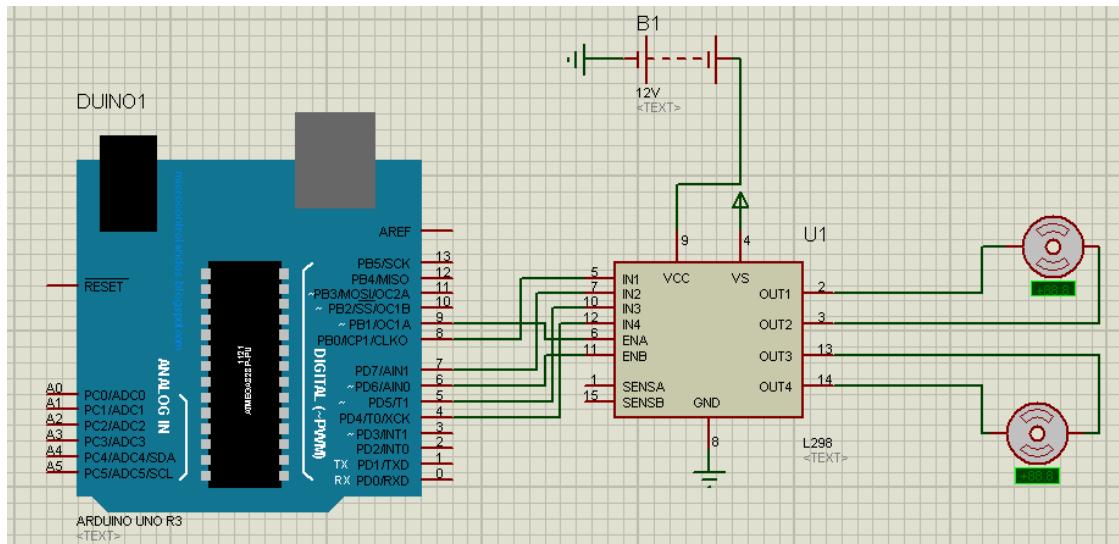
Sinyal *high* yang masuk akan membuat SRF04 mengeluarkan suara ultrasonik. Bunyi yang dipantulkan kembali ke sensor SRF04, bunyi tadi akan diterima dan membuat keluaran sinyal *high* pada pin *echo* yang kemudian menjadi *input* pada mikrokontroler. Lamanya sinyal *high* dari *echo* inilah yang digunakan untuk menghitung jarak antara sensor SRF04 dengan benda yang memantulkan bunyi yang berada di depan sensor. Pin *trigger* dan *echo* pada sensor ultrasonik bagian depan dihubungkan dengan pin 10 dan 11 pada pin digital mikrokontroler, sedangkan pin *trigger* dan *echo* pada sensor ultrasonik bagian belakang dihubungkan dengan pin 12 dan 13 pada pin digital mikrokontroler. Rancangan sensor ultrasonik dapat dilihat pada Gambar 23.



Gambar 23. Rangkaian Sensor Ultrasonik

d. Rancangan Modul L298N dengan 2 motor DC

Motor DC 12 Volt memerlukan adanya pengontrol motor DC agar putaran dan kecepatannya bisa dikendalikan dengan teratur, membutuhkan dua buah motor DC. Pengontrolan motor DC pada simulator kursi roda otomatis dengan sensor *flex* menggunakan modul L298N *driver* motorDC *dual H-Bridge* yang memiliki kemampuan menggerakkan motor DC sampai arus 4A dan tegangan maksimum 46 Volt DC untuk satu kanalnya. Melalui 4 pin yaitu pada IN0-IN3 sebagai *output*, sedangkan 2 pin untuk Vcc dan *ground* yang disuplai langsung dari modul mikrokontroler sebesar 5 Volt dan dari baterai Li-Po 1450 mAh 12 Volt. Pada Gambar 24 dapat dilihat skema rangaian modul L298N *driver* motor DC *dual H-Bridge*.



Gambar 24. Rangkaian Modul L298N dengan 2 Motor DC

## 2. Software

Perangkat lunak yang digunakan dalam pembuatan alat ini yaitu:

### a. Software Arduino IDE

Dalam pemrograman arduino ini sendiri menggunakan bahasa pemrograman C. *Listing* program arduino ini dikenal dengan nama *sketch*. Dalam setiap *sketch* memiliki dua buah fungsi penting yaitu “*void setup() {}*” dan “*void loop() {}*”. Pembuatan program arduino ini sendiri dimulai dengan menginisialisasi pin-pin mana saja yang akan digunakan oleh sistem. Adapun *listing* program secara keseluruhan dapat dilihat pada Lampiran 5, pada Gambar 25 merupakan potongan dari program yang digunakan:

```
#include <L298N.h>
const int FLEX_PIN0 = A0;
const int FLEX_PIN1 = A1;// Pin connected to voltage divider output
const int FLEX_PIN2 = A2;
const int FLEX_PIN3 = A3;// Pin connected to voltage divider output
const int FLEX_PIN4 = A4;
const float VCC = 4.98; // Measured voltage of Arduinio 5V line
const float R_DIV = 10000.0; // Measured resistance of 10k resistor

//pin definition
#define EN 9
#define EN2 6
#define IN1 8
#define IN2 7
#define IN3 5
#define IN4 4

//create a motor instance
L298N motor(EN, IN1, IN2);
L298N motor2(EN2, IN3, IN4);

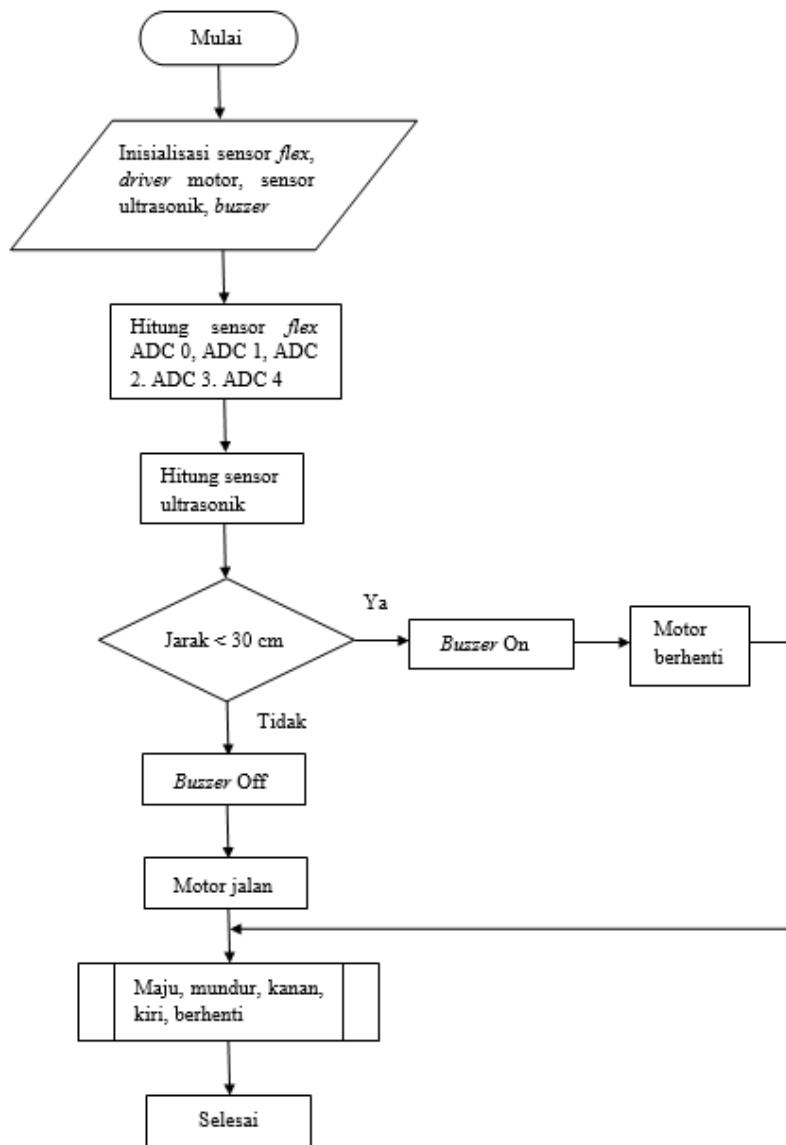
#define trigPin1 10
#define echoPin1 11
#define trigPin2 12
#define echoPin2 13
#define Buzzer 3
long duration1, distance1, duration2, distance2;
```

Gambar 25. Inisialisasi Pin pada Program Arduino Uno

b. Diagram alur (*flowchart*)

Berikut adalah diagram alur (*flowchart*) dari alat yang akan dibuat:

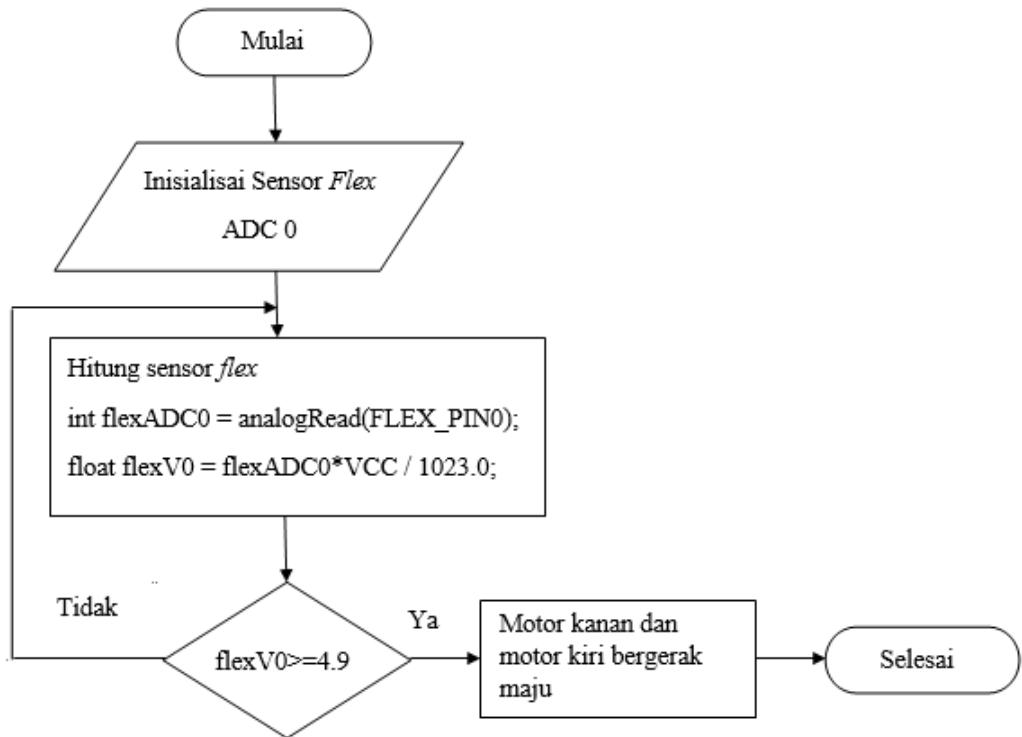
- 1) Alur *flowchart* kontrol gerakan kursi roda
  - a) Mulai
  - b) Menginisialisasi pin-pin yang digunakan oleh sistem
  - c) Memasukkan rumus dalam perhitungan sensor *flex*
  - d) Memasukkan rumus dalam perhitungan sensor ultrasonik
  - e) Apakah jarak<30 cm, jika ya maka *buzzer on* motor berhenti, jika tidak maka *buzzer off* motor berjalan
  - f) Subrutin gerakan kursi roda
  - g) Selesai



Gambar 26. *Flowchart* Kontrol Gerakan Kursi Roda

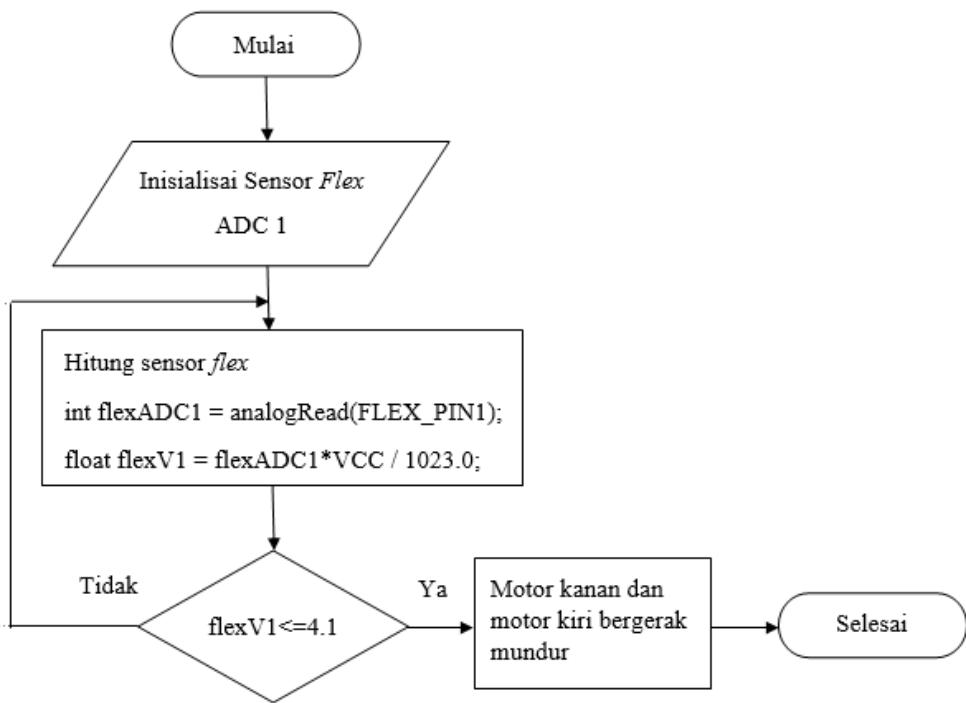
- 2) Alur *flowchart* gerakan maju
  - a) Mulai
  - b) Inisialisasi pin ADC 0 pada sensor *flex*
  - c) Memasukkan rumus dalam perhitungan sensor *flex*
  - d) Apakah  $\text{flexV0} \geq 4.9$ , jika ya motor kanan dan kiri bergerak maju, jika tidak *go to step 3*

e) Selesai



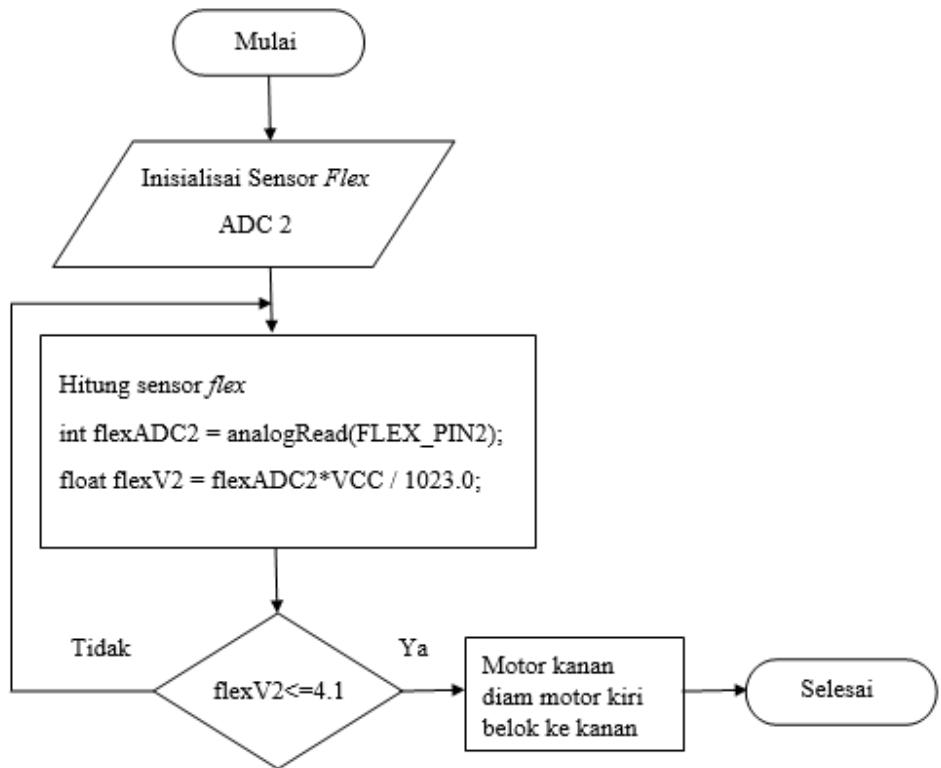
Gambar 27. Flowchart Gerakan Maju

- 3) Alur *flowchart* gerakan mundur
- Mulai
  - Inisialisasi pin ADC 1 pada sensor *flex*
  - Memasukkan rumus dalam perhitungan sensor *flex*
  - Apakah  $\text{flexV1} \leq 4.1$ , jika ya motor kanan dan kiri bergerak mundur, jika tidak *go to step 3*
  - Selesai



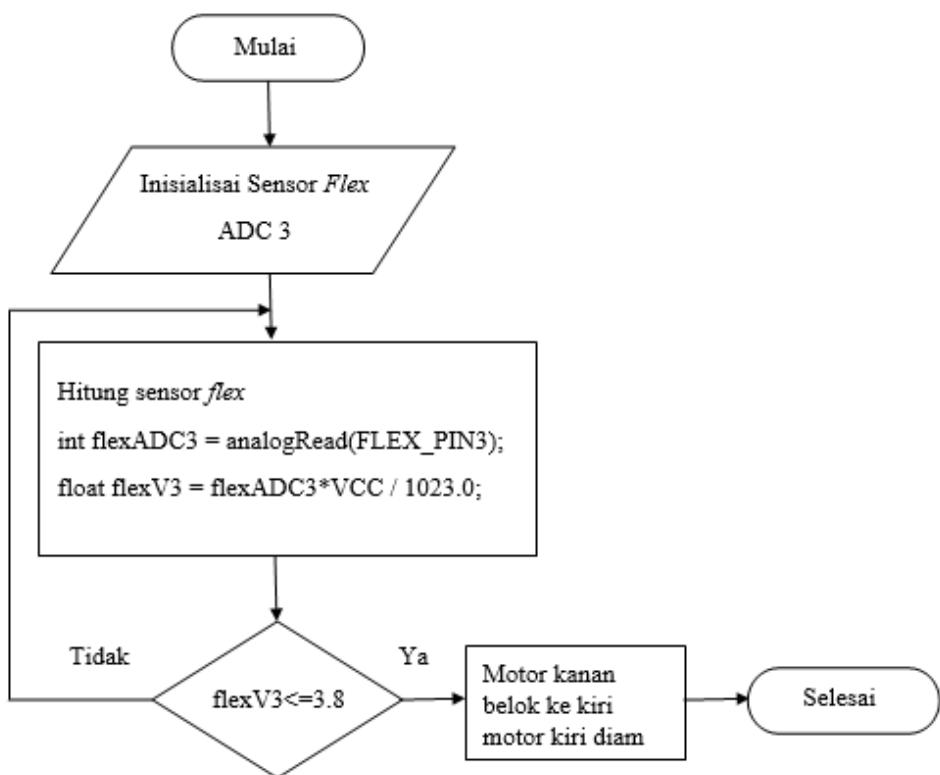
Gambar 28. *Flowchart* Gerakan Mundur

- 4) Alur *flowchart* gerakan kanan
- Mulai
  - Inisialisasi pin ADC 2 pada sensor *flex*
  - Memasukkan rumus dalam perhitungan sensor *flex*
  - Apakah  $\text{flexV2} \leq 4.1$ , jika ya motor kanan diam dan motor kiri belok ke kanan, jika tidak *go to step 3*
  - Selesai



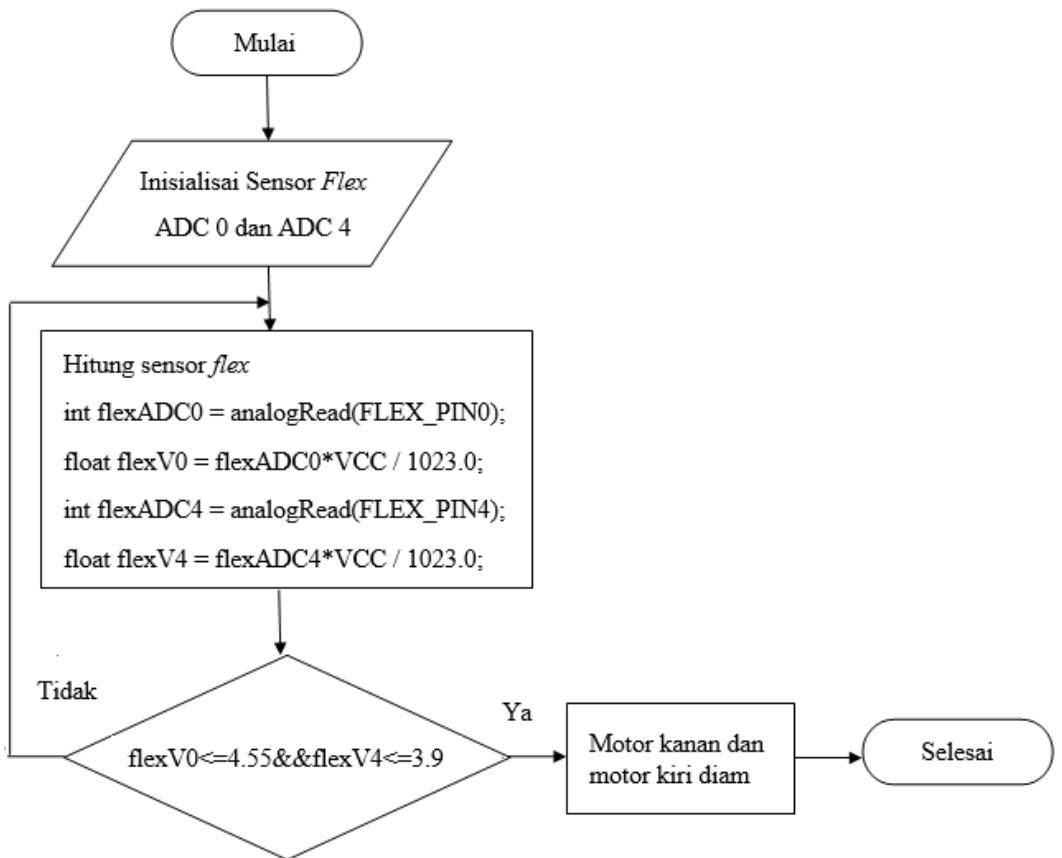
Gambar 29. *Flowchart* Gerakan Kanan

- 5) Alur *flowchart* gerakan kiri
  - a) Mulai
  - b) Inisialisasi pin ADC 3 pada sensor *flex*
  - c) Memasukkan rumus dalam perhitungan sensor *flex*
  - d) Apakah  $\text{flexV3} \leq 3.8$ , jika ya motor kanan belok ke kiri dan motor kiri diam, jika tidak *go to step 3*
  - e) Selesai



Gambar 30. Flowchart Gerakan Kiri

- 6) Alur *flowchart* kursi roda berhenti
  - a) Mulai
  - b) Inisialisasi pin ADC 0 dan ADC 4 pada sensor *flex*
  - c) Memasukkan rumus dalam perhitungan sensor *flex*
  - d) Apakah  $\text{flexV0} \leq 4.55 \& \& \text{flexV4} \leq 3.9$ , jika ya motor kanan dan kiri berhenti, jika tidak go to step 3
  - e) Selesai



Gambar 34. Flowchart Kursi Roda Berhenti

#### D. Langkah Pembuatan Alat

Tugas akhir ini memerlukan beberapa tahapan dalam pembuatannya adapun tahapannya sebagai berikut :

##### 1. Langkah Pembuatan Alat

Pembuatan simulator kursi roda otomatis dengan sensor *flex* berbasis mikrokontroler ini diperlukan beberapa tahapan dalam pembuatannya adapun tahapannya sebagai berikut:

- Perakitan komponen rangkaian simulator kursi roda otomatis dengan sensor *flex* berbasis mikrokontroler, yaitu sebagai berikut:

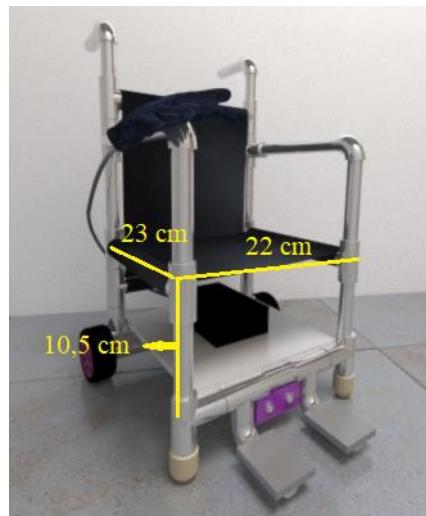
- 1) Menyiapkan alat yang akan digunakan berupa satu set obeng, gunting, korek gas, solder, tenol
- 2) Menyiapkan komponen yang akan digunakan berupa arduino uno, modul motor L298N, motor DC, sensor *flex*, sensor ultrasonik, *buzzer*, catu daya berupa baterai 9 Volt dan baterai Li-Po
- 3) Menyambung dan menyusun kabel pada modul motor yang dihubungkan dengan arduino uno dan motor DC
- 4) Menguji gerakan motor DC
- 5) Menyusun kabel sensor ultrasonik yang dihubungkan dengan arduino uno
- 6) Menambahkan *buzzer* sebagai tanda adanya halangan pada sensor ultrasonik
- 7) Menguji jarak yang terukur pada sensor ultrasonik
- 8) Merakit sekaligus melakukan pengujian koneksi antara masing-masing sensor *flex* dan *driver* motor DC dengan gerakan yang dihasilkan motor DC
- 9) Menggabungkan dan menguji gerakan sensor *flex* dan motor DC
- 10) Memasang semua rangkaian pada simulator kursi roda

b. Membuat kerangka simulator kursi roda, langkah pembuatannya sebagai berikut:

- 1) Menyiapkan alat yang akan digunakan berupa gergaji, lem pipa paralon, lem G, lem bakar

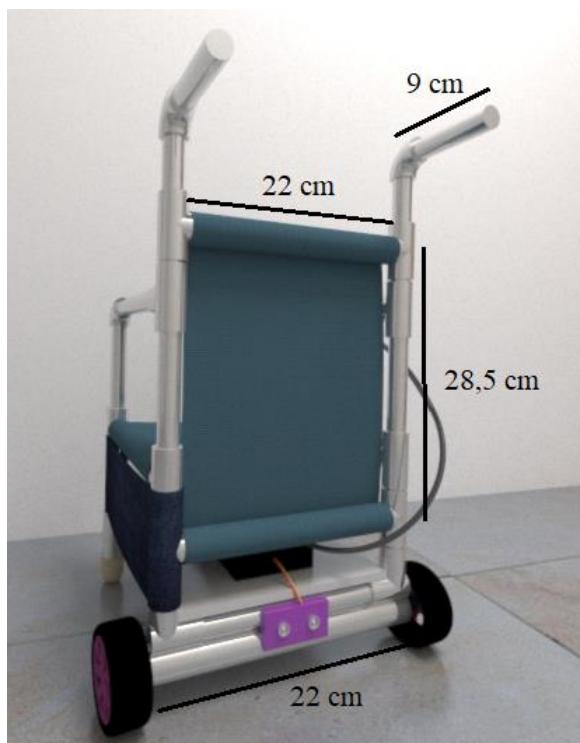
- 2) Menyiapkan alat yang akan digunakan berupa pipa paralon ukuran 5/8”, T pipa paralon, dan siku pipa paralon
- 3) Membuat desain simulator kursi roda dengan menggunakan pipa paralon
- 4) Memotong pipa paralon menjadi bagian-bagian dengan ukuran yang sudah ditentukan
- 5) Menyambung potongan paralon menjadi kerangka simulator kursi roda seperti pada desain yang sudah dibuat
- 6) Melakukan penyambungan kabel instalasi dan komponen yang telah siap untuk dipasang dalam sebuah *box*
- 7) Melakukan uji coba simulator kursi roda

Pada Gambar 32 dapat dilihat konsep kerangka simulator kursi roda tampak depan.



Gambar 32. Kerangka Kursi Roda Tampak Depan

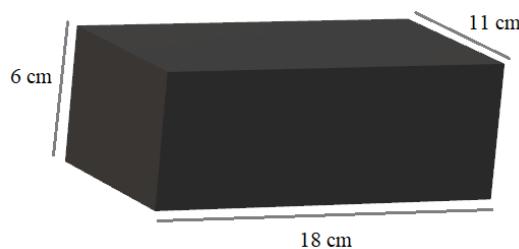
Gambar 32 menunjukkan desain simulator kursi roda tampak dari depan yang didesain dengan menggunakan pipa paralon berukuran 5/8 inchi, kemudian dipotong sebagai kerangka alat. Pada bagian dudukan kursi roda dibuat dengan ukuran lebar pada sisi kanan dan kiri 23 cm, sedangkan lebar pada sisi depan 22 cm, dan tinggi antara dudukan kursi roda dengan bagian bawah yang akan digunakan sebagai wadah *blac box* dengan ukuran 35 cm.



Gambar 33. Kerangka Kursi Roda Tampak Belakang

Gambar 33 menunjukkan desain simulator kursi roda tampak dari belakang. Bagian sandaran punggung di desain dengan ukuran lebar 22 cm dan tinggi sandaran 28,5 cm. Panjang pipa paralon bagian bawah 22 cm, paralon yang berukuran 1 inchi ini digunakan sebagai penghubung kabel dari roda ke *driver* motor yang berada di dalam *black box*. Pegangan tangan yang digunakan untuk

mendorong kursi roda dibuat dengan panjang 9 cm. Sandaran punggung, dudukan dan penutup *black box* pada kursi roda dibuat dengan menggunakan kain yang sudah dijahit sesuai dengan ukuran yang telah ditentukan. Bagian dudukan diberi gabus asturo sebagai penyangga *black box*. Adapun foto alat secara keseluruhan dapat dilihat pada Gambar 36 pada Lampiran 3.



Gambar 34. Ukuran *Black Box*

Gambar 34 menunjukkan ukuran *black box* yang digunakan sebagai wadah rangkaian komponen mulai dari arduino, baterai, saklar, kabel jumper yang sudah dirangkai komponen pendukungnya. Adapun isi *box* kontrol dapat dilihat pada Gambar 37 pada Lampiran 3.

#### E. Spesifikasi Alat

Simulator kursi roda otomatis dengan sensor *flex* berbasis mikrokontroler memiliki spesifikasi sebagai berikut.

1. Bahan pembuatan simulator kursi roda otomatis dengan sensor *flex* ini menggunakan bahan pipa paralon
2. Unit masukan

- a. Sensor *flex* berupa sensor yang memiliki perubahan resistansi akibat adanya perubahan lekukan pada bagian sensor, untuk mendeteksi suatu kelengkungan.
  - b. Sensor ultrasonik berfungsi untuk pendekripsi jarak ada atau tidaknya halangan di depan dan di belakang alat ini.
  - c. Modul L298N untuk mengontrol kecepatan serta arah perputaran motor DC.
3. Sistem pengendali yang digunakan adalah mikrokontroler arduino uno.
  4. Unit Keluaran
    - a. Motor DC 12 V yang berfungsi untuk penerima sinyal perintah dari modul L298N sekaligus penggerak kursi roda.
    - b. *Buzzer* sebagai isyarat suara ketika ada halangan di depan maupun di belakang kursi roda.
  5. Tegangan rangkaian yang digunakan adalah 9 V DC dan 12 V DC.
  6. Simulator kursi roda otomatis dengan sensor *flex* ini bekerja dengan prinsip otomatis dalam hal simulasi pergerakan kursi roda. Prinsip kerja dari simulator kursi roda otomatis dengan sensor *flex* adalah ketika sensor *flex* mendekksi perubahan tegangan akibat perubahan lekukan pada bagian sensor. Dengan memanfaatkan perubahan resistansi tersebut dengan prinsip pembagi tegangan. Perubahan tegangan akibat dari perubahan resistansi karena perubahan lekukan pada bagian sensor ini akan diolah oleh mikrokontroler arduino uno. Perubahan lekukan inilah yang digunakan sebagai dasar pendeksi kelengkungan jari pengguna kursi roda dan motor

DC akan bergerak sesuai intruksi jari pengguna. Pengguna kursi roda dapat leluasa mengatur gerakan simulator kursi roda dengan menekuk salah satu jari tangannya sesuai *setting* program gerakan sensor *flex* yang telah dibuat, dan simulator kursi roda akan berjalan.

## F. Pengujian Alat

### 1. Uji Fungsional

Pada pengujian ini dilakukan dengan cara mengukur setiap bagian dari masing-masing alat. Tujuan dari pengujian ini adalah untuk mengetahui setiap bagian dari perangkat telah bekerja sesuai dengan fungsinya. Dalam tugas akhir ini diperlukan beberapa pengujian yaitu :

#### a. Pengujian *Hardware*

##### 1) Pengukuran tegangan catu daya *driver* motor DC

Pengukuran catu daya ini dilakukan untuk mengukur catu daya untuk *driver* motor DC. *Driver* motor L298N memiliki rangkaian driver L298N yang berfungsi sebagai sistem kendali motor DC, salah satunya mengatur tegangan keluaran motor yang nantinya menjadi referensi untuk kecepatan motor. Pengukuran bertujuan untuk mengetahui kinerja dan respon dari rangkaian driver pengendali motor DC L298N dengan membandingkan dan menguji sinyal keluaran dari *driver* motor terhadap sinyal masukan yang diberikan oleh mikrokontroler.

Motor DC dikontrol oleh modul L298N *driver* motor DC *dual H-Bridge*. L298N yang berfungsi sebagai sistem kemudi motor DC salah satunya mengatur tegangan keluaran motor yang nantinya menjadi referensi untuk kecepatan motor. Pin EN1 dan EN2 pada *driver* motor L298N dihubungkan dengan pin 9 dan 6

pada pin digital mikrokontroler. Sedangkan pin IN1, IN2, IN3, IN4 secara berurutan terhubung pada pin 8, 7, 5, 4 pada pin digital mikrokontroler.

Tabel 3. Rencana Pengukuran Tegangan Catu Daya *Driver Motor DC*

No	Pengukuran	Vin	Vout		Selisih
			Motor Kanan	Motor Kiri	
1.	Tanpa Beban				
2.	Dengan Beban				

## 2) Pengukuran Sensor *Flex*

Rangkaian sensor *flex* berfungsi sebagai pendekripsi gerakan pada sarung tangan yang dipakai oleh pengguna untuk menggerakkan kursi roda. Sensor ini akan menghasilkan *output* berupa resistansi ketika mendekripsi adanya gerakan berupa lengkungan jari pada pengontrol sarung tangan. Letak masing-masing sensor *flex* dihubungkan pada analog pin arduino mulai dari A0, A1, A2, A3, dan A4 ditunjukkan pada Tabel 12 mengenai konfigurasi sensor *flex*

Tabel 4. Rencana Konfigurasi Sensor *Flex*

Pin Analog	<i>Output</i> Gerakan	Letak Sensor	Posisi Jari

Tabel 5. Rencana Pengukuran Sensor *Flex*

Jari	Posisi Jari	R multimeter (KΩ)	ADC	Vout (V)

### 3) Pengukuran Sensor Ultrasonik

Pengukuran terhadap sensor ultrasonik dilakukan dengan mengukur jarak sensor dengan menggunakan penggaris dan mencatat ukurannya saat suatu penghalang yang diletakkan dihadapan sensor terukur, pembacaan sensor dibatasi pada jarak pengujian 5 cm-100 cm.

Tabel 6. Rencana Pengukuran Sensor Ultrasonik

Pengujian ke-	Jarak Pengujian (cm)	Jarak Terukur (cm)			
		Sensor Depan	Selisih	Sensor Belakang	Selisih
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

### b. Pengujian *Software*

Pengujian ini dilakukan untuk mengetahui kinerja alat dari program yang telah dibuat dengan menggunakan arduino. Program yang dimasukkan ke arduino dibuat dengan menggunakan *tools* pemrograman arduino IDE. Fungsi pemrograman disini bertujuan untuk menginisialisasi pin-pin mana saja yang akan menjadi *output* dan *input*, menggunakan *library* program sensor *flex* dan *driver* motor.

### 2. Uji Unjuk Kerja

Pada pengujian unjuk kerja fungsional alat ini dilakukan dengan cara menggerakkan jari tangan yang sudah dipasang sarung tangan untuk menggerakkan kursi roda. Kemudian uji coba gerakan motor DC dan sensor ultrasonik saat diberi halangan, dan untuk mengetahui kondisi *buzzer*. Pengujian ini dilakukan pada semua bagian alat untuk mengetahui keterikatan antara alat ini.

### a. Pengujian *Buzzer*

Pengujian *buzzer* dilakukan untuk mengetahui jarak yang terbaca atau tidak terbaca pada *serial monitor* dan kondisi *buzzer* pada sensor ultrasonik, dengan membagi dalam beberapa jarak yang telah ditentukan dan memberikan halangan di depan sensor ultrasonik. Mikrokontroler akan menerima data *logic* dari sensor ultrasonik dilakukan dengan pengaktifan port dari *buzzer* yang *active high* atau aktif saat menerima logika “1” saat sensor mengirim hasil pendektsian jarak dan *buzzer* akan berbunyi. Kaki *buzzer* positif dihubungkan dengan pin 3 pada pin digital mikrokontroler, sedangkan kaki *buzzer* negatif dihubungkan pada GND di mikrokontroler.

Tabel 7. Rencana Pengujian *Buzzer*

Jarak	Banyak Percobaan (kali)	Jarak Respon <i>Buzzer</i>		Kondisi <i>Buzzer</i>	
		Sensor Depan	Sensor Belakang	Sensor Depan	Sensor Belakang

### b. Pengujian Gerak Motor DC

Pengujian motor DC dilakukan untuk mengetahui putaran roda pada masing-masing motor DC, juga digunakan sensor ultrasonik dengan melibatkan parameter jarak dari benda yang dijadikan penghalang dan mengetahui kondisi *buzzer*.

Tabel 8. Rencana Pengujian Gerak Motor DC

Gerakan yang diinginkan	Putaran Roda	
	Motor Kanan	Motor Kiri
Maju		
Mundur		
Kanan		
Kiri		
Berhenti		

Tabel 9. Rencana Pengujian Gerak Motor DC dengan Sensor Ultrasonik

Gerakan Kursi Roda	Sensor Depan		Sensor Belakang	
	Diberi Halangan	Tanpa Halangan	Diberi Halangan	Tanpa Halangan
Maju				
Mundur				
Kanan				
Kiri				

### 3. Pengujian Secara Keseluruhan

Pada pengujian ini seluruh rangkaian yang terpasang pada alat baik *hardware* maupun *software* dioperasikan. Seluruh komponen yang akan diujikan dirangkai menjadi satu setelah itu dilakukan pengujian dengan menjalankan kursi roda. Pengujian dilakukan dengan dua cara yaitu dengan ada penghalang dan tanpa penghalang di depan kursi roda. Tujuan dari pengujian ini adalah untuk mengetahui kinerja alat baik dan benar sesuai dengan yang diharapkan. Rencana pengujian alat terdapat pada Tabel 10.

Tabel 10. Rencana Pengujian Secara Keseluruhan

Posisi Jari	Gerakan Kursi Roda	Penghalang	Jarak Penghalang	Kondisi <i>Buzzer</i>
Ibu jari dilengkungkan 60° ke atas		Ada		
		Tidak Ada		
Telunjuk dilengkungkan 50° ke bawah		Ada		
		Tidak Ada		
Jari tengah dilengkungkan 50° ke bawah		Ada		
		Tidak Ada		
Jari manis dilengkungkan 50° ke bawah		Ada		
		Tidak Ada		
Kelingking dan ibu jari dilengkungkan 50° ke bawah		Ada		
		Tidak Ada		

## **BAB IV**

### **PENGUJIAN DAN PEMBAHASAN**

#### **A. Pengujian**

Pengujian sistem ini memiliki tujuan untuk menguji kinerja serta hubungan antara perangkat keras dengan perangkat lunak sebagai program aplikasi sistem. Dengan pengujian ini dapat diketahui apakah alat dan aplikasi yang telah dirancang dapat bekerja sesuai dengan yang diinginkan atau tidak.

Pengujian ini dilakukan pada perangkat keras dan perangkat lunak.

##### **1. Uji Fungsional**

###### **a. Pengujian *Hardware***

###### **1) Pengukuran tegangan catu daya *driver* motor DC**

Tabel 11. Pengukuran Tegangan Catu Daya *Driver* Motor DC

No	Pengukuran	Vin	Vout		Selisih
			Motor Kanan	Motor Kiri	
1.	Tanpa Beban	12,5	12	12	0
		12,5	12	12	0
		12,5	12	12	0
2.	Dengan Beban	12,10	10,76	10,75	0,01
		12,10	10,75	10,71	0,04
		12,08	10,74	10,68	0,06

## 2) Pengukuran Sensor *Flex*

Tabel 12. Konfigurasi Sensor *Flex*

Pin Analog	<i>Output</i> Gerakan	Letak Sensor	Posisi Jari
A0	Maju	Ibu jari	Dilengkungkan 60° ke atas
A1	Mundur	Telunjuk	Dilengkungkan 50° ke bawah
A2	Kanan	Tengah	Dilengkungkan 50° ke bawah
A3	Kiri	Manis	Dilengkungkan 50° ke bawah
A4 dan A1	Berhenti	Kelingking dan ibu jari	Keduanya dilengkungkan 50° ke bawah

Tabel 13. Pengukuran Sensor *Flex*

Jari	Posisi Jari	R multimeter (KΩ)	ADC	Vout (V)
Ibu jari dilengkungkan ke atas	Lurus (0°)	9,0	4,5	4,4
	Lengkung 60° ke atas	7,0	4,65	4,6
Ibu jari dilengkungkan ke bawah	Lurus (0°)	8,0	4,5	4,4
	Lengkung 50° ke bawah	7,9	3,67	3,65
Telunjuk	Lurus (0°)	9,0	4,56	4,48
	Lengkung 50° ke bawah	8,9	3,87	3,75
Tengah	Lurus (0°)	8,0	4,29	4,1
	Lengkung 50° ke bawah	7,9	3,91	3,34
Manis	Lurus (0°)	9,0	4,00	3,9
	Lengkung 50° ke bawah	8,9	3,54	3,28
Kelingking	Lurus (0°)	9,0	4,19	4,19
	Lengkung 50° ke bawah	8,9	3,4	3,1

### 3) Pengukuran Sensor Ultrasonik

Tabel 14. Pengukuran Sensor Ultrasonik

Pengujian ke-	Jarak Pengujian (cm)	Jarak Terukur (cm)			
		Sensor Depan	Selisih	Sensor Belakang	Selisih
1	5	5	0	4	1
2	10	10	0	9	1
3	15	18	3	14	1
4	20	20	0	20	0
5	30	30	0	28	2
6	45	44	1	42	3
7	55	55	0	53	2
8	60	59	1	58	2
9	85	84	1	83	2
10	100	99	1	98	2

#### b. Pengujian *Software*

Pengujian program disini bertujuan untuk menginisialisasi pin-pin mana saja yang akan menjadi *output* dan *input*, menggunakan *library* program sensor *flex* dan *driver* motor.

2. Uji Unjuk Kerja

a. Pengujian *Buzzer*

Tabel 15. Pengujian *Buzzer*

Jarak (cm)	Banyak Percobaan (kali)	Kondisi <i>Buzzer</i>		Hasil
		Sensor Depan	Sensor Belakang	
5 - 30	4	Berbunyi	Berbunyi	Sesuai
31 – 179	3	Tidak Berbunyi	Tidak Berbunyi	Sesuai
180 - 185	3	Tidak Berbunyi	Tidak Berbunyi	Sesuai

b. Pengujian Gerak Motor DC

Tabel 16. Pengujian Gerak Motor DC

Gerakan yang diinginkan	Putaran Roda	
	Motor Kanan	Motor Kiri
Maju	Berputar ke depan	Berputar ke depan
Mundur	Berputar ke belakang	Berputar ke belakang
Kanan	Diam	Berputar belok kanan
Kiri	Berputar belok kiri	Diam
Berhenti	Diam	Diam

Tabel 17. Pengujian Gerak Motor DC dengan Sensor Ultrasonik

Gerakan Kursi Roda	Sensor Depan		Sensor Belakang	
	Diberi Halangan	Tanpa Halangan	Diberi Halangan	Tanpa Halangan
Maju	Berhenti	Berjalan	Berhenti	Berjalan
Mundur	Berhenti	Berjalan	Berhenti	Berjalan
Kanan	Berhenti	Berjalan	Berhenti	Berjalan
Kiri	Berhenti	Berjalan	Berhenti	Berjalan

### 3. Pengujian Secara Keseluruhan

Tabel 18. Pengujian Secara Keseluruhan

Posisi Jari	Gerakan Kursi Roda	Penghalang	Jarak Penghalang	Kondisi Buzzer
Kelingking dan ibu jari dilengkungkan 50° ke bawah	Berhenti	Ada	5 cm	Nyala
		Tidak Ada	-	Mati
Ibu jari dilengkungkan 60° ke atas	Maju	Ada	25 cm	Nyala
		Tidak Ada	-	Mati
Telunjuk dilengkungkan 50° ke bawah	Mundur	Ada	30 cm	Nyala
		Tidak Ada	-	Mati
Jari tengah dilengkungkan 50° ke bawah	Kanan	Ada	50 cm	Mati
		Tidak Ada	-	Mati
Jari manis dilengkungkan 50° ke bawah	Kiri	Ada	181 cm	Mati
		Tidak Ada	-	Mati
		Tidak Ada	-	Mati

## B. Pembahasan

Berdasarkan hasil pengamatan dan pengujian beberapa rangkaian dan komponen pada proyek akhir ini, maka dapat disimpulkan bahwa seluruh rangkaian dapat bekerja dengan baik dan sesuai dengan fungsi dari masing-masing komponen. Pada pengujian pengukuran rangkaian sistem terdapat beberapa perbedaan karena adanya selisih dari hasil pengukuran dengan apa yang diperoleh dari teori *datasheet* komponen. Perbedaan tersebut dipengaruhi oleh banyak faktor seperti alat ukur yang digunakan, nilai komponen yang tidak sesuai

dengan labelnya, toleransi nilai komponen dari pabrik yang memproduksi komponen, dan kurang telitinya dalam pengukuran.

1. Uji Fungsional

a. Pengujian *Hardware*

1) Analisis pengukuran tegangan catu daya *driver* motor DC

Catu daya yang digunakan dalam rangkaian sistem ini yaitu menggunakan sumber daya dari baterai Li-Po 12 V 1450 mAh, menggunakan catu daya ini karena memiliki tegangan *output* yang stabil, sebagai sumber untuk memberikan *supply* daya motor DC. Tegangan *input* untuk motor DC dihasilkan dengan menghubungkan baterai Li-Po dengan *power supply driver* motor pada GND dan 12 V. L298 adalah *driver* motor dengan spesifikasi yang mampu menangani beban hingga 4A pada tegangan kerja 5-35 V. Tegangan *input* yang dihasilkan pada pengukuran tanpa beban maupun dengan beban menghasilkan rata-rata sebesar 12,3 V.

Pengukuran dilakukan dengan mengukur tanpa diberi beban pada *output* sebanyak tiga kali percobaan, dan dengan memberi beban pada *output* berupa motor DC, dengan hasil pengukuran berikut:

a) Tanpa beban

Pengukuran daya tanpa beban dilakukan sebanyak tiga kali, hal ini dilakukan agar data yang didapatkan *valid*. Dapat dilihat pada tabel 10 hasil pengukuran *input* yang didapatkan 12,5 Volt, dengan pengukuran tanpa beban rata-rata pada motor kanan maupun motor kiri menghasilkan tegangan *output* sebesar 12 Volt

jadi tidak murni sebesar 12,5 Volt dengan selisih tegangan *output* motor kanan dan motor kiri 0 Volt. Pengukuran tanpa beban ini dilakukan tanpa menyambungkan kabel motor DC pada masing-masing port motor DC yang berada pada *driver* motor L298N, karena beban yang dipakai adalah dua motor DC.

b) Dengan beban

Pengukuran daya dengan beban juga dilakukan sebanyak tiga kali pengukuran. Hasil yang didapatkan dapat dilihat pada tabel 10 hasil pengukuran tegangan *input* yang didapatkan pada percobaan pertama sebesar 12,10 Volt dengan diberi beban pada motor kanan menghasilkan *output* sebesar 10,76 Volt, sedangkan pada motor kiri menghasilkan *output* sebesar 10,75 Volt. Pada percobaan kedua tegangan *input* yang didapatkan sebesar 12,10 Volt dengan diberi beban pada motor kanan menghasilkan *output* sebesar 10,75 Volt, sedangkan pada motor kiri menghasilkan *output* sebesar 10,71 Volt. Pada percobaan terakhir tegangan *input* yang didapatkan sebesar 12,08 Volt dengan diberi beban pada motor kanan menghasilkan *output* sebesar 10,74 Volt, sedangkan pada motor kiri menghasilkan *output* sebesar 10,68 Volt. Pengukuran dengan memberikan beban, rata-rata tegangan *input* yang dihasilkan 12,09 Volt, rata-rata tegangan *output* yang dihasilkan motor kanan sebesar 10,75 Volt, rata-rata tegangan *output* yang dihasilkan motor kiri sebesar 10,71 Volt,dengan selisih rata-rata pada motor kanan dan motor kiri sebesar 0,36 Volt.

Pengukuran tegangan pada motor DC tidak penuh 12 Volt dikarenakan tegangan yang diperoleh motor DC melewati *driver* motor yang menyebabkan *supply* tegangan yang diperoleh berkurang, selain itu regulasi dan *filter* pada *driver* motor juga berpengaruh terhadap *supply* tegangan pada motor DC. Dari hasil pengujian dapat disimpulkan bahwa tegangan yang diperoleh motor DC tidak penuh karena melewati *driver* motor sehingga daya yang diperoleh sedikit dan menyebabkan putaran motor DC menjadi lambat.

## 2) Analisis pengujian sensor *flex*

Sensor *flex* berfungsi sebagai pendekripsi gerakan pada sarung tangan yang dipakai oleh pengendali. Sensor ini akan menghasilkan *output* berupa resistansi ketika mendekripsi adanya gerakan berupa lengkungan jari pada pengontrol sarung tangan. Dibutuhkan sarung tangan yang berserat halus atau tidak berserat dan berbahan yang tipis, karena untuk memudahkan sensor *flex* masuk pada selubung yang telah dibuat pada bagian atas sarung tangan.

Jari telunjuk digerakkan dengan cara dilengkungkan ke bawah atau berada pada sudut  $50^\circ$  ke bawah, maka akan mendapatkan *output* gerakan mundur. Jari tengah digunakan untuk mendapatkan *output* gerakan belok ke kanan dengan cara dilengkungkan ke bawah atau berada pada sudut  $50^\circ$  ke bawah. Jari manis digunakan untuk mendapatkan *output* gerakan belok ke kiri dengan cara dilengkungkan ke bawah atau berada pada sudut  $50^\circ$  ke bawah. Sensor *flex* yang terdapat pada kelingking digunakan untuk mendapatkan *output* gerakan berhenti. Posisi jari pada saat menggerakkan sensor adalah dilengkungkan ke bawah atau berada pada sudut  $50^\circ$  ke bawah, tetapi terdapat kesulitan pada saat

melengkungkan jari ke depan. Maka dari itu untuk mendapatkan *output* gerakan berhenti tanpa kesulitan menggerakkan jari kelingking, dilakukan dengan menambahkan gerakan pada ibu jari dengan kedua posisi jari dilengkungkan ke bawah atau berada pada sudut  $50^\circ$  ke bawah. Selain itu, sensor *flex* yang terdapat pada ibu jari juga digunakan untuk mendapatkan *output* gerakan maju. Posisi jari pada saat menggerakkan sensor adalah dilengkungkan sedikit ke atas atau berada pada sudut  $60^\circ$  ke atas. Penggunaan ibu jari dengan posisi tersebut dapat membantu saat ingin mendapatkan *output* gerakan maju dengan posisi jari kelingking dan ibu jari dilengkungkan ke bawah, agar tidak terjadi gerakan yang bertumbukan pada ibu jari.

Pengukuran sensor *flex* bertujuan untuk melihat perubahan nilai resistansi yang terjadi pada sensor. Masing-masing sensor diuji dengan posisi jari pada keadaan lurus dan ditekuk (dilengkungkan). Posisi sensor *flex* pada jari berdasarkan pada dua kondisi yaitu saat jari dalam keadaan lurus atau berada pada sudut  $0^\circ$  dan saat jari dalam keadaan dilengkungkan ( $50^\circ$  -  $60^\circ$ ). Pada tabel 12 ditunjukkan pengujian sensor *flex* ini dilakukan dengan mengamati berapa nilai resistansi yang dihasilkan ketika terdapat kelengkungan pada saat sensor dilengkungkan. Ketika sensor *flex* dilengkungkan maka akan menghasilkan nilai resistansi yang semakin kecil. Ketika sensor *flex* dilengkungkan ke bawah maka perubahan tegangan yang semakin kecil, sedangkan jika sensor *flex* dilengkungkan ke atas maka perubahan tegangan semakin besar. Pada saat ibu jari dilengkungkan ke atas, resistansi yang didapatkan saat posisi jari lurus sebesar  $9,0\text{ K}\Omega$  sedangkan resistansi yang didapatkan saat posisi jari dilengkungkan ke atas sebesar  $8,9\text{ K}\Omega$ . Tegangan

keluaran yang dihasilkan saat posisi jari lurus sebesar 4,4 V sedangkan resistansi yang didapatkan saat posisi jari dilengkungkan ke atas sebesar 4,6 V. Perubahan tegangan yang semakin besar dikarenakan posisi jari yang dilengkungkan semakin ke atas, maka akan mempengaruhi besar ADC yang semakin bertambah. Begitu juga sebaliknya pada saat jari yang lain semakin dilengkungkan ke bawah, maka akan mempengaruhi terhadap besar resistansi dan tegangan keluarannya.

Cara menentukan besar resistor untuk pembagi tegangan adalah harus mengetahui besar resistansi maksimum dan minimum dari sensor *flex*. Pada pengujian yang didapatkan pada tabel 12, dapat diketahui bahwa resistansi minimum yang dihasilkan ketika sensor *flex* dilengkungkan adalah 7 K $\Omega$  sedangkan resistansi maksimumnya adalah 10 K $\Omega$  yang merupakan hambatan datar sensor *flex*. Pin analog (ADC) dapat mengukur nilai tegangan yang masuk dengan kondisi normal *range* 0-5 Vdc yang akan dibandingkan dengan tegangan Vref. Hal ini berguna saat mengukur *output* tegangan dari suatu sensor yang terhubung dari serta dapat dimanfaatkan untuk keperluan pengondisian program. Suatu tegangan 5V yang dihubungkan dengan 2 resistor dengan nilai yang sama, maka hasilnya akah separuhnya. Tegangan *output* pada data adalah setengah dari 5 V, sebesar 2,5 V. Dengan menggunakan rumus pembagi tegangan yaitu :

$$V_{out} = \frac{R_1}{R_1+R_2} \times V_{cc}$$

Vout = Tegangan keluaran pada data

R1 = Resistansi tetap pada sensor *flex*

R2 = Resistansi yang diukur pada sensor *flex*

Vcc = tegangan *input* arduino

Perhitungan untuk mendapatkan Vout:

$$V_{out} = \frac{R_1}{R_1+R_2} \times V_{cc}$$

$$V_{out} = \frac{10\text{ K}\Omega}{10\text{ K}\Omega+10\text{ K}\Omega} \times 5\text{ V}$$

$$V_{out} = \frac{10\text{ K}\Omega}{20\text{ K}\Omega} \times 5\text{ V}$$

$$V_{out} = 2,5\text{ V}$$

Selanjutnya adalah menentukan nilai Vref dengan cara mengambil nilai tengah ADC 8 bit yaitu  $255/2 = 127$ . Hasil nilai tengah Vref dikonversi menjadi tegangan 2,5 V pada Arduino Uno, sehingga bisa membaca data di bawah dan di atasnya. Pada ADC 8 bit tegangan 5 V direpresentasikan menjadi angka 255 dalam desimal, sedangkan 0 V direpresentasikan menjadi angka 0 dalam desimal. Kemudian cara menghitung besar resistor yang digunakan adalah dengan menggunakan rumus pembagi tegangan, dengan perhitungan sebagai berikut:

$$V_{out} = \frac{R_1}{R_1+R_2} \times V_{cc}$$

$$2,5\text{ V} = \frac{R_1}{R_1+10\text{ K}\Omega} \times 5\text{ V}$$

$$\frac{2,5\text{ V}}{5\text{ V}} = \frac{R_1}{R_1 + 10\text{ K}\Omega}$$

$$5R_1 = 2,5\text{ V} (R_1 + 10\text{ K}\Omega)$$

$$5R_1 = 2,5\text{ V} R_1 + 25\text{ K}\Omega$$

$$5R_1 - 2,5R_1 = 25\text{ K}\Omega$$

$$2,5R_1 = 25\text{ K}\Omega$$

$$R_1 = 10\text{ K}\Omega$$

Dari hasil perhitungan maka dapat didapatkan resistor tetap pada sensor *flex* sebesar 10 KΩ. Berdasarkan pengujian sensor *flex* yang telah dilakukan

menunjukkan bahwa nilai resistansi yang didapat berbanding terbalik dengan posisi jari saat melengkungkan sensor. Semakin besar sudut kelengkungan sensor tersebut, maka semakin rendah nilai resistansi sensor. Keluaran sensor *flex* berupa resistansi yang memiliki nilai hambatan mulai dari  $10 \text{ K}\Omega$  -  $40 \text{ K}\Omega$ , nilai hambatan tersebut berubah secara linear. Sensor *flex* bekerja menggunakan prinsip *voltage divider* atau pembagi tegangan sehingga dapat meruah nilai tegangan yang akan dibaca oleh mikrokontroler arduino uno. Nilai tegangan yang diubah oleh sensor *flex* memiliki rnetang nilai 0 V hingga 5 V. Rangkaian pembagi tegangan digunakan karena keluaran sensor *flex* berupa hambatan dan resistansi. Keluaran dari rangkaian pembagi tegangan tersebut dihubungkan pada *port analog input* arduino uno sehingga mikrokontroler dapat membaca nilai tegangan yang keluar dari rangkaian sensor tersebut. Nilai ADC didapatkan dari nilai yang terbaca pada *serial monitor* arduino yang menunjukkan niai tegangan dari port analog *input*. Sedangkan nilai Vout sensor dapat diukur dengan menggunakan multimeter.

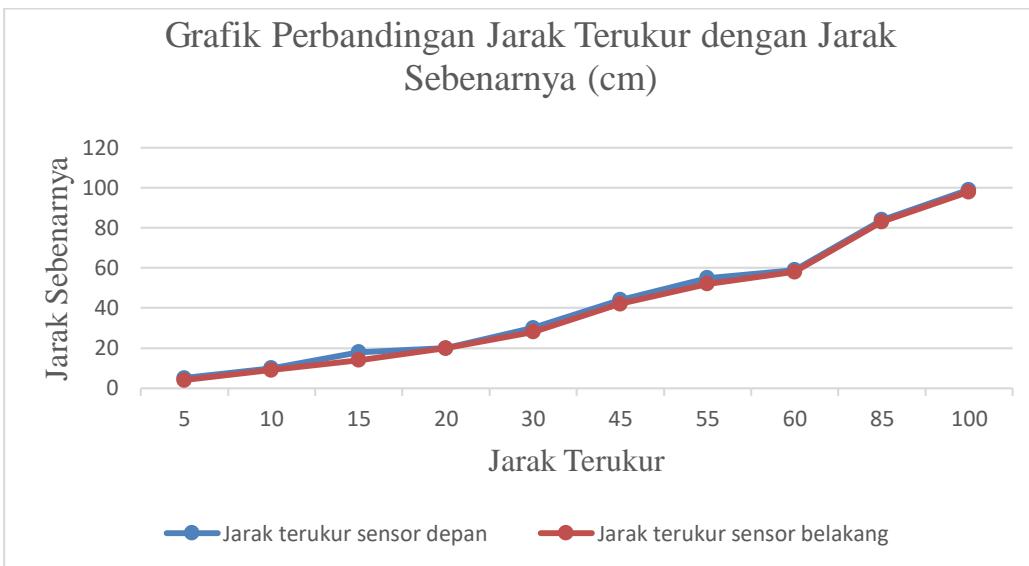
### 3) Analisis pengukuran sensor ultrasonik

Penggunaan sensor ultrasonik HC-SR04 karena pengukuran jarak yang akurat dengan ketelitian sampai dengan 0,3 cm dan pengukuran maksimum dapat mencapai 4 meter dengan jarak minimum 2 cm. Dari hasil pengukuran menggunakan sensor ultrasonik, pengujian jarak dilakukan dari jarak minimum 5 cm sampai jarak maksimum 100 cm dan dapat diketahui bahwa pengukuran yang dilakukan sensor hampir mendekati jarak sesungguhnya. Namun hasil jarak yang terdeteksi oleh sensor ultrasonik dapat berbeda-beda, tergantung bahan suatu permukaan atau benda yang terdeteksi oleh sensor ultrasonik. Tidak semua bahan

dapat memantulkan bunyi ultrasonik, ada bahan yang permukaannya lunak sehingga dapat menyerap bunyi atau disebut kedap terhadap bunyi. Bahan kedap bunyi seperti karpet, karet, busa, wol, karton, dan gabus.

Jarak terukur (jarak yang terbaca oleh sensor) pada sensor ultrasonik bagian depan mempunyai kisaran selisih 0-3 cm, sedangkan jarak terukur (jarak yang terbaca oleh sensor) pada sensor ultrasonik bagian belakang mempunyai kisaran selisih 0-3 cm yang telah ditunjukkan pada Tabel 13. Sensor ultrasonik bekerja berdasarkan perbandingan dari berapa waktu yang ditangkap setelah gelombang itu dipancarkan. Semakin jauh benda maka waktu pantulan akan semakin lama sedangkan jika semakin dekat benda maka pantulan akan semakin cepat. Perbedaan pembacaan sensor pada sensor bagian belakang kurang akurat karena selisih pembacaan yang lebih besar dibandingkan dengan pembacaan sensor pada sensor bagian depan. Perbedaan jarak hasil pengukuran dengan hasil yang terbaca pada *serial monitor* dapat pula disebabkan oleh adanya *noise*, sensor ultrasonik bekerja berdasarkan prinsip pemantulan gelombang ultrasonik, terkadang pantulan gelombang ultrasonik menjadi tidak periodik dan menyebabkan hasil pengukuran tidak akurat.

Setelah dilakukan beberapa percobaan maka dapat dibuat grafik presentasi antara jarak terukur (jarak yang terbaca oleh sensor) dengan jarak sebenarnya, dari Gambar 35 dapat dilihat bahwa perbandingan antara jarak terukur (jarak yang terbaca oleh sensor depan maupun belakang) dengan jarak sebenarnya adalah berbanding lurus.



Gambar 35. Perbandingan Jarak Terukur dengan Jarak Sebenarnya

Hal ini karena pada setiap pembacaan jarak pada sensor depan maupun sensor belakang hampir mendekati nilai data pada pengukuran jarak sebenarnya.

### b. Pengujian *Software*

#### 1) Inisialisasi pin *input* arduino

```

#include <L298N.h>
const int FLEX_PIN0 = A0;
const int FLEX_PIN1 = A1;// Pin connected to voltage divider output
const int FLEX_PIN2 = A2;
const int FLEX_PIN3 = A3;// Pin connected to voltage divider output
const int FLEX_PIN4 = A4;
const float VCC = 4.98; // Measured voltage of Arduunio 5V line
const float R_DIV = 10000.0; // Measured resistance of 10k resistor

//pin definition
#define EN 9
#define EN2 6
#define IN1 8
#define IN2 7
#define IN3 5
#define IN4 4

//create a motor instance
L298N motor(EN, IN1, IN2);
L298N motor2(EN2, IN3, IN4);

#define trigPin1 10
#define echoPin1 11
#define trigPin2 12
#define echoPin2 13
#define Buzzer 3
long duration1, distance1, duration2, distance2;

```

Pada potongan program di atas merupakan pin-pin yang digunakan sebagai *input* arduino. Semua pin yang digunakan harus diinisialisasi untuk mengenalkan pin mana saja yang akan digunakan dan diaktifkan pada arduino.

## 2) Inisialisasi pin output arduino

Potongan program berikut merupakan inisialisasi pin yang akan digunakan untuk menentukan *input* dan *output* yang akan mendapatkan loika *high* atau *low*.

```
void setup() {
    Serial.begin(9600);
    //Set pins as outputs
    pinMode(FLEX_PIN0, INPUT);
    pinMode(FLEX_PIN1, INPUT);
    pinMode(FLEX_PIN2, INPUT);
    pinMode(FLEX_PIN3, INPUT);
    pinMode(FLEX_PIN4, INPUT);

    pinMode(trigPin1, OUTPUT);
    pinMode(echoPin1, INPUT);
    pinMode(trigPin2, OUTPUT);
    pinMode(echoPin2, INPUT);
    pinMode(Buzzer, OUTPUT);
```

## 3) Kalkulasi perhitungan sensor *flex*

Potongan program berikut merupakan rumus yang digunakan untuk perhitungan sensor *flex*.

```
void loop() {
    int flexADC0 = analogRead(FLEX_PIN0);
    float flexV0 = flexADC0 * VCC / 1023.0;
    float flexR0 = R_DIV * (VCC / flexV0 - 1.0);
    Serial.print("Volt : " + String(flexV0) + " volt ");
    //Serial.println("Resistance: " + String(flexR) + " ohm ");
    delay(200);
```

#### 4) Kalkulasi perhitungan jarak sensor ultrasonik

Potongan program ini merupakan penentu dari logika yang akan diberikan logika *high* atau *low* pada sensor ultrasonik dan perintah gerakan motor dengan menggunakan logika *high* atau *low* pada *buzzer*.

```
digitalWrite(trigPin1, LOW);
delayMicroseconds(2);
digitalWrite(trigPin1, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin1, LOW);
delayMicroseconds(10);

duration1 = pulseIn(echoPin1, HIGH);
distance1 = (duration1/2) / 29.1;

if (distance1 < 30){
    digitalWrite(Buzzer,HIGH); //less than 30cm then buzzer will produce beep sound
    motor.setSpeed(0);
    motor2.setSpeed(0);
    //tell the motor to go back (may depend by your wiring)
    motor.forward();
    motor2.forward();
}
else {
    digitalWrite(Buzzer,LOW);
}
if (distance1 >= 180 || distance1 <= 0)
{
    Serial.println("Out of range1");
}
else {
    Serial.print(distance1);
    Serial.println(" cm");
    delay(1000);
}
```

Hasil pengujian yang dilakukan dengan memberikan jarak di depan sensor depan dan belakang. Pada *serial monitor* akan membaca jarak penghalang yang ada di depan sensor depan sejauh 2 cm, sedangkan jarak penghalang yang ada di depan sensor bagian belakang sejauh 96 cm.

```
Volt : 4.71 volt Volt1 : 4.60 volt Volt2 : 4.41 volt Volt3 : 4.18 volt Volt4 : 4.51 volt
2 cm
96 cm
```

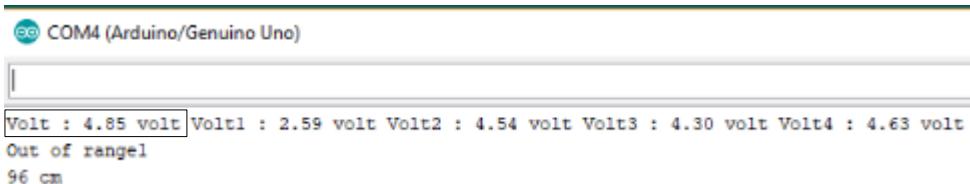
## 5) Identifikasi logika perintah

Potongan program berikut ini merupakan salah satu dari logika yang diberikan *low* atau *high* dari gerakan jari tangan. Pada Tabel 16 hasil pengujian untuk mengetahui apakah kursi roda sudah dapat bergerak sesuai dengan gerakan yang diinginkan. Pada pengujian alat ini dilakukan dengan menggerakkan jari tangan yang sudah dipasangkan sarung tangan khusus untuk dapat menggerakkan kursi roda mengikuti intruksi pengguna kursi roda. Respon dari setiap perintah sudah cukup baik dan bekerja sesuai dengan yang diharapkan.

```
void maju (){
    int flexADC0 = analogRead(FLEX_PIN0);
    float flexV0 = flexADC0 * VCC / 1023.0;

    if(flexV0>=4.9){// POSISI NOL ATAU TIDAK JALAN Maju dan mundur
        motor.setSpeed(255);
        motor2.setSpeed(255);
        //tell the motor to go back (may depend by your wiring)
        motor.forward();
        motor2.forward();
        delay(3000);
    }
}
```

Pengujian gerakan maju didapatkan dari tampilan *serial monitor* pada program arduino IDE. Ketika ibu jari digerakkan ke atas, maka tampilan hasil pengujian pada *serial monitor* akan seperti tampilan berikut:



Pada program arduino akan membaca data ADC lebih besar dan sama dengan 4,9 V untuk mendapatkan gerakan maju, sedangkan pada *serial monitor* menunjukkan Volt0 = 4.85 V maka kursi roda akan bergerak maju.

```
void mundur (){
    int flexADC1 = analogRead(FLEX_PIN1);
    float flexV1 = flexADC1 * VCC / 1023.0;

    if(flexV1<=4.3){// POSISI NOL ATAU TIDAK JALAN Maju dan mundur
        motor.setSpeed(255);
        motor2.setSpeed(255);
        //tell the motor to go back (may depend by your wiring)
        motor.backward();
        motor2.backward();
        delay(3000);
    }
}
```

Pengujian gerakan mundur didapatkan dari tampilan *serial monitor* pada program arduino IDE. Ketika jari telunjuk digerakkan ke bawah, maka hasil pengujian tampilan pada *serial monitor* akan seperti tampilan berikut:

```
Volt : 4.75 volt Volt1 : 4.12 volt Volt2 : 4.37 volt Volt3 : 4.19 volt Volt4 : 4.52 volt
Out of range!
95 cm
```

Pada program arduino akan membaca data ADC lebih kecil dan sama dengan 4,3 V untuk mendapatkan gerakan mundur, sedangkan pada *serial monitor* menunjukkan Volt1 = 4.12 V maka kursi roda akan bergerak mundur.

```

void kanan (){
    int flexADC2 = analogRead(FLEX_PIN2);
    float flexV2 = flexADC2 * VCC / 1023.0;

    if(flexV2<=4.1){// POSISI NOL ATAU TIDAK JALAN Maju dan mundur
        motor.setSpeed(0);
        motor2.setSpeed(255);
        //tell the motor to go back (may depend by your wiring)
        motor.forward();
        motor2.forward();
        delay(3000);
    }
}

```

Pengujian gerakan kanan didapatkan dari tampilan *serial monitor* pada program arduino IDE. Ketika jari tengah digerakkan ke bawah, maka hasil pengujian tampilan pada *serial monitor* akan seperti tampilan berikut:

```

Volt : 4.74 volt Volt1 : 2.59 volt Volt2 : 4.07 volt Volt3 : 3.71 volt Volt4 : 4.52 volt
Out of range!
95 cm

```

Pada program arduino akan membaca data ADC lebih kecil dan sama dengan 4,1 V untuk mendapatkan gerakan ke kanan, sedangkan pada *serial monitor* menunjukkan Volt2 = 4.07 V maka kursi roda akan bergerak ke kanan.

```

void kiri (){
    int flexADC3 = analogRead(FLEX_PIN3);
    float flexV3 = flexADC3 * VCC / 1023.0;

    if(flexV3<=3.8){// POSISI NOL ATAU TIDAK JALAN Maju dan mundur
        motor.setSpeed(255);
        motor2.setSpeed(0);
        //tell the motor to go back (may depend by your wiring)
        motor.forward();
        motor2.forward();
        delay(3000);
    }
}

```

Pengujian gerakan kiri didapatkan dari tampilan *serial monitor* pada program arduino IDE. Ketika jari manis digerakkan ke bawah, maka hasil pengujian tampilan pada *serial monitor* akan seperti tampilan berikut:

```
Volt : 4.74 volt Volt1 : 2.59 volt Volt2 : 4.07 volt Volt3 : 3.71 volt Volt4 : 4.52 volt
Out of range!
95 cm
```

Pada program arduino akan membaca data ADC lebih kecil dan sama dengan 3,8 V untuk mendapatkan gerakan ke kiri, sedangkan pada *serial monitor* menunjukkan Volt3 = 3,71 V maka kursi roda akan bergerak ke kiri.

```
void berhenti (){
    int flexADC0 = analogRead(FLEX_PIN0);
    float flexV0 = flexADC0 * VCC / 1023.0;
    int flexADC4 = analogRead(FLEX_PIN4);
    float flexV4 = flexADC4 * VCC / 1023.0;

    if(flexV0<=4.85&&flexV4<=4.1){ // POSISI NOL ATAU TIDAK JALAN Maju dan mundur
        motor.setSpeed(0);
        motor2.setSpeed(0);
        //tell the motor to go back (may depend by your wiring)
        motor.forward();
        motor2.forward();
        delay(3000);
    }
}
```

Pengujian agar kursi roda berhenti didapatkan dari tampilan *serial monitor* pada program arduino IDE. Ketika ibu jari dan kelingking digerakkan ke bawah, maka hasil pengujian tampilan pada *serial monitor* akan seperti tampilan berikut:

```
Volt : 4.74 volt Volt1 : 2.71 volt Volt2 : 4.42 volt Volt3 : 4.15 volt Volt4 : 4.06 volt
Out of range!
95 cm
```

Pada program arduino akan membaca data ADC lebih kecil dan sama dengan 4,85 V saat ibu jari digerakkan ke bawah, dan akan membaca data ADC sebesar kurang dari sama dengan 4,1 V saat kelingking digerakkan ke bawah untuk

mendapatkan gerakan ke kiri, sedangkan pada *serial monitor* menunjukkan saat ibu jari digerakkan ke bawah mendapatkan nilai data ADC sebesar Volt = 4,74 V, dan saat kelingking digerakkan ke bawah mendapatkan nilai data ADC sebesar Volt4 = 4,06 V maka kursi roda akan bergerak ke kiri.

## 2. Uji Unjuk Kerja

### a. Analisis pengujian *buzzer*

Pengujian *buzzer* bertujuan untuk memberi *output* sensor ultrasonik melalui suara yang keluar dari *buzzer*. *Buzzer* ini berfungsi untuk mengetahui apakah di depan kursi roda tersebut terdapat halangan atau tidak. Pada pengujian ini menggunakan fungsi *delay* pada program untuk mengontrol *buzzer* dalam keadaan berbunyi dan tidak berbunyi secara bergantian dengan sela waktu yang ditentukan dengan menggunakan program. Jarak pengujian *buzzer* dilakukan dengan membagi dalam tiga rentang jarak, hal ini bertujuan untuk memudahkan dalam pengujian jarak pada *buzzer* dengan memberikan halangan di depan sensor ultrasonik.

Pada setiap rentang jarak, didapatkan kondisi *buzzer* “berbunyi” jika sensor ultrasonik mendeteksi suatu halangan, atau kondisi “tidak berbunyi” jika sensor ultrasonik tidak mendeteksi adanya jarak yang diberikan pada saat pengujian dilakukan. Hasil pengujian yang ditunjukkan pada Tabel 14 menunjukkan pengujian dengan memberikan halangan di depan sensor ultrasonik bagian depan dengan rentang jarak 5 cm sampai 30 cm *buzzer* dalam kondisi berbunyi. Pengujian kedua dengan rentang jarak 31 cm sampai 179 cm *buzzer* dalam kondisi

tidak berbunyi. Pengujian ketiga dengan rentang jarak 180 cm sampai 185 cm *buzzer* dalam kondisi tidak berbunyi.

Data hasil pengujian didapatkan dengan menggunakan rentang jarak yang berbeda-beda. Gelombang ultrasonik melakukan pancar-terima sehingga waktu yang dibutuhkan menjadi 2x, dan hal ini berpengaruh pada perhitungan jaraknya. Pengujian *buzzer* yang digunakan sebagai indikator untuk mengetahui apakah ada halangan di depan kursi roda. Hasil pengujian dapat membuktikan bahwa sensor ultrasonik bekerja berdasarkan kemampuan objek memantulkan kembali gelombang ultrasonik yang dikirim oleh sensor ultrasonik, gangguan pada pendektsian sensor ultrasonik dapat diakibatkan oleh karakteristik objek yang kurang mampu untuk memantulkan gelombang suara dengan baik, jika objek dapat memantulkan gelombang suara dengan baik dapat ditandai dengan *buzzer* yang berbunyi.

b. Analisis pengujian gerak motor DC

Penggunaan motor DC ini sebagai indikator gerakan kursi roda, ada lima gerakan kursi roda yang diinginkan seperti ditunjukkan pada Tabel 15. Pengujian kursi roda dengan gerakan maju, motor kanan dan motor kiri berputar ke depan. Pengujian kursi roda dengan gerakan mundur, motor kanan dan motor kiri berputar ke belakang. Pengujian kursi roda dengan gerakan ke kanan, motor kanan dalam keadaan diam sedangkan motor kiri berputar belok ke kanan. Pengujian kursi roda dengan gerakan ke kiri, motor kanan dalam keadaan berputar belok ke kiri sedangkan motor kiri diam. Dari hasil pengujian gerakan motor DC dapat

dilihat bahwa putaran motor DC pada motor kanan dan motor kiri sudah sesuai dengan gerakan kursi roda yang diinginkan.

Pengujian gerak motor DC dengan sensor ultrasonik yang ditunjukkan pada Tabel 16 dilakukan untuk mengetahui apakah sensor ultrasonik bagian depan dan belakang sudah sesuai dengan gerakan yang diinginkan. Pengujian sensor dilakukan dengan memberi halangan dan tanpa halangan di depan kursi roda. Gerakan kursi roda sudah sesuai dengan gerakan yang diinginkan oleh pengguna kursi roda. Pengujian dilakukan sebanyak empat kali, yaitu maju, mundur, kanan dan kiri. Dari hasil pengujian tersebut dapat dilihat bahwa kursi roda dapat berhenti ketika di depan sensor ultrasonik diberikan halangan, dan kursi roda tetap berjalan jika tidak ada halangan yang menghalangi gerakan kursi roda.

### 3. Analisis pengujian secara keseluruhan

Dari hasil pengujian alat secara keseluruhan didapatkan bahwa alat sudah bekerja sesuai dengan yang diharapkan. Alat sudah baik dalam merespon gerakan yang diinginkan dapat dilihat pada Tabel 16 dan Tabel 17. Hasil pengujian secara keseluruhan, kursi roda dapat bergerak sesuai dengan gerakan jari pengguna kursi roda. Saat diberi penghalang maka kursi roda berjalan dan *buzzer* berbunyi sedangkan saat tidak diberi penghalang maka kursi roda tetap berjalan. Hasil pengujian secara keseluruhan dapat dilihat pada Tabel 18.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **A. Kesimpulan**

Dari pembuatan simulator kursi roda otomatis dengan sensor *flex* berbasis mikrokontroler ini dapat diperoleh kesimpulan sebagai berikut:

1. Pembuatan perangkat keras simulator kursi roda otomatis dengan sensor *flex* berbasis mikrokontroler dirangkai menggunakan baterai Li-Po 12 V 1450 mAh sebagai catu daya pada *driver* motor DC yang digunakan untuk menggerakkan motor DC. Sedangkan untuk mengaktifkan mikrokontroler arduino uno berfungsi sebagai kendali utama diperlukan sumber tegangan dari baterai 9 V, selain itu berfungsi sebagai catu daya sensor ultrasonik, *buzzer*, sensor *flex*.
2. Pembuatan perangkat lunak program yang digunakan untuk menjalankan mikrokontroler arduino uno menggunakan arduino IDE sebagai pembuatan *source code* program menggunakan bahasa C. Pada program arduino IDE menggunakan *library* untuk menjalankan sistem dari driver motor L298N dan sensor *flex*.
3. Hasil unjuk kerja secara keseluruhan dari simulator kursi roda otomatis dengan sensor *flex* berbasis mikrokontroler memiliki unjuk kerja yaitu kursi roda dapat bergerak maju, mundur, kanan, kiri sesuai perintah pengguna kursi roda dan ketika ada penghalang *buzzer* berbunyi dan kursi roda berhenti.

## **B. Keterbatasan Alat**

Simulator kursi roda otomatis dengan sensor *flex* berbasis mikrokontroler memiliki keterbatasan antara lain:

1. Sensor *flex* sangat sensitif dikarenakan pada saat sensor dilengkungkan maka nilai resistansinya akan berubah-ubah, dalam menggerakkan sarung tangan harus dilakukan secara hati-hati agar sesuai dengan nilai yang telah dibuat pada program arduino IDE.
2. Kecepatan motor DC lambat karena tegangan yang diperoleh motor DC tidak penuh, sebab melewati *driver* motor sehingga daya yang diperoleh sedikit.

## **C. Saran**

Dalam pembuatan proyek akhir ini tentunya terdapat beberapa kekurangan, sehingga diperlukan pengembangan guna menyempurnakan proyek akhir ini. Adapun saran yang dapat diberikan adalah sebagai berikut:

1. Gunakan sarung tangan yang terbuat dari bahan yang tidak berserat kasar dan berbahan tipis. Hal ini dapat menghambat proses ketika memasukkan sensor *flex* ke dalam selubung wadah sensor *flex* yang berada di bagian pada setiap jari tangan.
2. Pastikan kabel penghubung tertancap sempurna pada masing-masing komponen dan menggunakan motor DC dengan arus yang lebih kecil dan torsi yang lebih besar.

## DAFTAR PUSTAKA

- Afif, M., T. Dan Pratiwi, I., A., P. (2015). *Analisis Perbandingan Baterai Lithium-Ion, Lithium-Polymer, Lead Acid dan Nickel-Metal Hydride pada Penggunaan Mobil Listrik – Review*. Universitas Brawijaya. Malang.
- Ardiansyah, A. Dan Hidayatama, O. (2013). *Rancang Bangun Prototipe Elevator Menggunakan Microcontroller Arduino ATMEGA 328P*. Universitas Mercu Buana. Jakarta. Vol. 4 (3): 102.
- Arindya, Radita, S.T., M.T. (2013). *Penggunaan dan Pengaturan Motor Listrik*. Graha Ilmu, Yogyakarta.
- Arduino.(2009). *ArduinoBoardUno* .[www.arduino.cc/en/Main/ArduinoBoardUno](http://www.arduino.cc/en/Main/ArduinoBoardUno). Diakses pada tanggal 2 Maret 2018.
- D. Petruzella, Frank. (2001). *Elektronika Industri*. Andi. Yogyakarta.
- Depdikbud. (1997). Undang-Undang RI Nomor 4, tahun 1997, tentang Penyandang Disabilitas.
- Dwika, O. dan Widodo, R. (2012). *Kursi Roda Elektrik Menggunakan Joystik Berbasis Mikrokontroler ATMEGA 8535*. Universitas Negeri Jakarta. Jakarta.
- Departemen Pendidikan Nasional. (2005). *Pedoman Pengertian Simulator*. Jakarta: Depdiknas.
- Farastya, P. (2017). *7 Ragam Kursi Roda Unik dan Bermanfaat untuk Segala Kondisi*. <http://www.medicalogy.com/blog/7-ragam-kursi-roda-unik-dan-bermanfaat-untuk-segala-kondisi/>. Diakses pada tanggal 23 Februari 2018.
- Hamdani, M. (2010). *Pengendalian Kecepatan Putaran Motor DC Terhadap Perubahan Temperatur dengan Sistem Modulasi Lebar Pulsa*. Universitas Indonesia. Jakarta.
- Hardiyanto, R.D., Rochim, A.F., dan Windasari, I.P. (2015). *Pembuatan Penghitung Jumlah Mobil Otomatis Berbasis Mikrokontroler ATMega 8535 Menggunakan Sensor Ultrasonik*. Jurnal Teknologi dan Sistem Komputer, Vol. 3, No. 2, April 2015 (e-ISSN: 2338-0403). Universitas Diponegoro. Semarang.

- Herdianto, C. (2017). *Robotics, Tutorials snd Experiences*. <https://candraherdianto.blogspot.co.id/>. Diakses pada tanggal 2 Maret 2018.
- Ikram, F.D., Aisuwarya, R., dan Hersyah, M.H. (2016). *Rancang Bangun Lengan Robot Pengikut Pergerakan Lengan dan Jari Manusia Menggunakan Flex Sensor*. Universitas Andalas. Sumatera Barat.
- Infodatin. (2014). *Penyandang Disabilitas Pada Anak*. Jakarta: Pusat Data dan Informasi Kementerian Kesehatan RI.
- Kementerian Kesehatan RI. (2014). *Situasi Penyandang Disabilitas*. [Versi Elektronik]. Buletin Jendela Data dan Informasi Kesehatan, ISSN 2088-270X, 6.
- Kusuma, Alfian Anta. (2015). *Lengan Robot Peniru Gerakan Tangan Manusia*. Universitas Sanata Dharma. Yogyakarta.
- Kuswadi, Sonk. (2000). “*Kendali Cerdas*”. EEPIS Press.
- Liem, Hatta, Y.K., Pujiyo, dkk. (2012). *Rancang Bangun Kursi Roda Elektrik Menggunakan Perintah Suara Berbasis Aplikasi Android*. Institut Teknologi Sepuluh Nopember. Surabaya.
- Maria, V. dan Dwi W.U. (2013). *Rancang Bangun Jari Tangan Robot Pengikut Pergerakan Jari Tangan Manusia*. Jurnal, Palembang: STMIK MDP.
- Musbikhin. (2014). *Baterai Li-Po (Lithium-Polimer)*. [www.musbikhin.com](http://www.musbikhin.com). Diakses pada tanggal 10 Februari 2018.
- Nyebarilmu. (2017). *Tutorial Arduino Mengakses Driver Motor L298N*. <https://www.nyebarilmu.com/tutorial-arduino-mengakses-driver-motor-l298n/>. Diakses pada tanggal 2 Maret 2018.
- Oktarima. (2013). *Rancang Bangun Pengukur Level Tanki Bahan Bakar Pada SPBU dengan Aplikasi Mikrokontroler Berbasis Teamviewer*. Politeknik Negeri Sriwijaya. Palembang.
- Sanjaya, W. (2008). *Strategi Pembelajaran*, h.159. Pernada Media Group. Jakarta.
- Saputra, P. (2017). *Smart Home dengan Speech Recognition Melalui Bluetooth Berbasis Android*. Universitas Negeri Yogyakarta. Yogyakarta.
- Sarwoko, J. (2011). *Cara Kerja Sensor HC-SR04*. <http://www.accudiy.com>. Diakses pada tanggal 10 Februari 2018.

Setiawan, Iwan. (2009). *Buku Ajar Sensor dan Transduser*. In: *Sensor dan Transduser*. Faculty of Engineering, Diponegoro University. Diakses pada 31 Oktober 2017, dari <http://eprints.undip.ac.id/>.

Sparkfun databaseheet. (2014). *Dataseheet Flex Sensor*. [www.sparkfun.com](http://www.sparkfun.com). Diakses pada tanggal 2 Maret 2018.

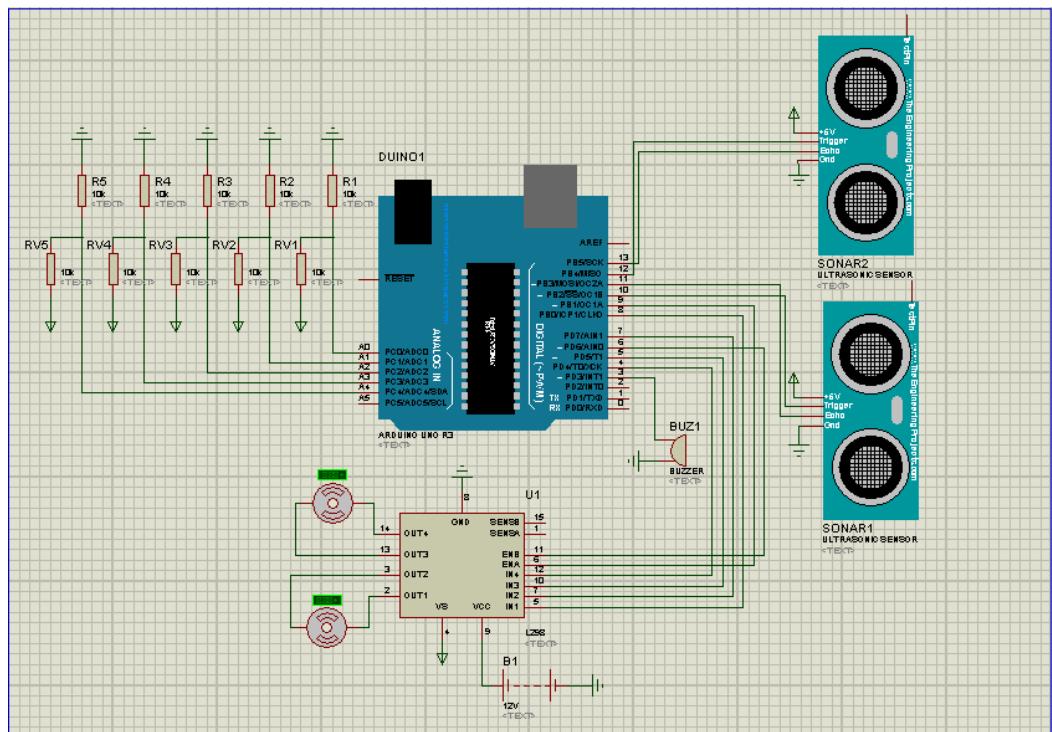
Sulistyowati, R. dan Febriantoro, Dedi Dwi. (2012). *Perancangan Prototype Sistem Kontrol dan Monitoring Pembatas Daya Listrik Berbasis Mikrokontroler*. Jurnal IPTEK Vol 16 No.1 Mei 2012: 27. Institut Teknologi Adhi Tama Surabaya. Surabaya.

Taufik, R. (2014). *Perancangan dan Implementasi Pengontrol Robot Jari Tangan Menggunakan Sensor Flex*. Universitas Komputer Indonesia. Bandung.

Umam, Fatkhul. (2011). *Penjejak Objek pada Kursi Roda Cerdas Autonomous*. Skripsi S-1, Universitas Diponegoro, Semarang.

# LAMPIRAN

## Lampiran 1. Skema Rangkaian Keseluruhan



## SKEMA RANGKAIAN KESELURUHAN

KETERANGAN

A4 No.01



TEK. ELEKTRONIKA D3  
FAKULTAS TEKNIK  
UNIV. NEGERI YOGYAKARTA

SKA. 1:1

DIS. SRI W

DIG. AFFIN

DIP. SRI W

NIM. 15507134002

## Lampiran 2. Daftar Komponen

1. *Sensor flex*
2. *Driver motor L293N*
3. *Sensor Ultrasonik HC-SR04*
4. *Motor DC*
5. *Buzzer*
6. *Arduino Uno R3*
7. *Baterai Li-Po 12 V 1450 mAh*
8. *Baterai 9 V*
9. *Resistor 10 KΩ*
10. *Black box*
11. *Kabel jumper*
12. *LED warna merah*
13. *Pipa paralon*
14. *T pipa paralon*
15. *Siku pipa paralon*
16. *Gibus asturo*
17. *Pyrex warna silver*

DAFTAR KOMPONEN			KETERANGAN	
 TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. 1:1	DIS. SRI W	A4	No.02
	DIG. AFFIN	DIP. SRI W	NIM. 15507134002	

**Lampiran 3. Foto Alat**



Gambar 36. Foto Alat Secara Keseluruhan

**FOTO ALAT**

KETERANGAN

A4 No.03



TEK. ELEKTRONIKA D3  
FAKULTAS TEKNIK  
UNIV. NEGERI YOGYAKARTA

SKA. 1:1

DIS. SRI W

DIG. AFFIN

DIP. SRI W

NIM. 15507134002



Gambar 37. Isi Box Kontrol

FOTO ALAT		KETERANGAN	
TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. 1:1	DIS. SRI W	A4   No.03
	DIG. AFFIN	DIP. SRI W	NIM. 15507134002

#### Lampiran 4. Petunjuk Pengoperasian Alat

1. Pengoperasian alat ini dapat dilakukan dengan cara berikut.
2. Menghidupkan saklar yang terhubung pada baterai 9 Volt dalam kondisi ON.
3. Menghubungkan kabel konektor baterai Li-Po dengan power supply modul motor L298N.
4. Memastikan di depan dan di belakang kursi roda tidak ada halangan yang menghalangi kursi roda, agar kursi roda dapat bergerak
5. Memasang sarung tangan yang sudah disambungkan dengan sensor gerak pada tangan
6. Menggerakkan jari tengah ke arah atas sebagai intruksi maju, menggerakkan jari tengah ke arah bawah sebagai intruksi mundur, menggerakkan jari telunjuk ke arah atas sebagai intruksi belok kanan, dan menggerakkan jari telunjuk ke arah bawah sebagai intruksi belok kiri
7. Kursi roda akan merespon sesuai intruksi yang diberikan oleh pengguna kursi roda
8. Mencoba dengan memberikan penghalang pada bagian depan atau pada bagian belakang kursi roda
9. Buzzer akan berbunyi ketika ada penghalang yang menghalangi kursi roda

#### PETUNJUK PENGOPERASIAN ALAT

KETERANGAN

A4 No.04



TEK. ELEKTRONIKA D3  
FAKULTAS TEKNIK  
UNIV. NEGERI YOGYAKARTA

SKA. 1:1

DIS. SRI W

DIG. AFFIN

DIP. SRI W

NIM. 15507134002

## Lampiran 5. Listing Program

```
#include <L298N.h>                                #define IN2 7
const int FLEX_PIN0 = A0;                            #define IN3 5
const int FLEX_PIN1 = A1;                            #define IN4 4
// Pin connected to voltage divider output      //create a motor instance
const int FLEX_PIN2 = A2;                            L298N motor(EN, IN1, IN2);
const int FLEX_PIN3 = A3;                            L298N motor2(EN2, IN3, IN4);

// Pin connected to voltage divider output
const int FLEX_PIN4 = A4;                            #define trigPin1 10
const float VCC = 4.98;                            #define echoPin1 11
// Measured voltage of Arduinio 5V line           #define trigPin2 12
const float R_DIV = 10000.0;                         #define echoPin2 13
// Measured resistance of 3.3k resistor           #define Buzzer 3
distance2;                                         long duration1, distance1, duration2,
//pin definition
#define EN 9                                         void setup(){
#define EN2 6                                         Serial.begin(9600);
#define IN1 8                                         //Set pins as outputs
```

LISTING PROGRAM			KETERANGAN	
		A4	No.05	
 TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. 1:1	DIS. SRI W		
	DIG. AFFIN	DIP. SRI W	NIM. 15507134002	

```

pinMode(FLEX_PIN0, INPUT);           delay(200);

pinMode(FLEX_PIN1, INPUT);

pinMode(FLEX_PIN2, INPUT);           int flexADC1 = analogRead(FLEX_PIN1);

pinMode(FLEX_PIN3, INPUT);           float flexV1 = flexADC1 * VCC / 1023.0;

pinMode(FLEX_PIN4, INPUT);           float flexR1 = R_DIV * (VCC / flexV1 - 1.0);

                                         Serial.print("Volt1 : " + String(flexV1) + " volt ");

pinMode(trigPin1, OUTPUT);          delay(200);

pinMode(echoPin1, INPUT);

pinMode(trigPin2, OUTPUT);          int flexADC2 = analogRead(FLEX_PIN2);

pinMode(echoPin2, INPUT);           float flexV2 = flexADC2 * VCC / 1023.0;

pinMode(Buzzer, OUTPUT);            float flexR2 = R_DIV * (VCC / flexV2 - 1.0);

                                         Serial.print("Volt2 : " + String(flexV2) + " volt ");

}

                                         delay(200);

void loop(){

int flexADC0 = analogRead(FLEX_PIN0);

float flexV0 = flexADC0 * VCC / 1023.0;

float flexR0 = R_DIV * (VCC / flexV0 - 1.0);

Serial.print("Volt : " + String(flexV0) + " volt ");

```

<b><i>LISTING PROGRAM</i></b>			KETERANGAN	
		A4	No.05	
	TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. 1:1	DIS. SRI W	
		DIG. AFFIN	DIP. SRI W	NIM. 15507134002

```

int flexADC3 = analogRead(FLEX_PIN3);
float flexV3 = flexADC3 * VCC / 1023.0;
float flexR3 = R_DIV * (VCC / flexV3 - 1.0);
Serial.print("Volt3 : " + String(flexV3) + " volt ");
delay(200);

int flexADC4 = analogRead(FLEX_PIN4);
float flexV4 = flexADC4 * VCC / 1023.0;
float flexR4 = R_DIV * (VCC / flexV4 - 1.0);
Serial.println("Volt4 : " + String(flexV4) + " volt ");
delay(200);

digitalWrite(trigPin1, LOW);
delayMicroseconds(2);
digitalWrite(trigPin1, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin1, LOW);
delayMicroseconds(10);

duration1 = pulseIn(echoPin1, HIGH);
distance1 = (duration1/2) / 29.1;
if (distance1 < 30){ digitalWrite(Buzzer,HIGH);
motor.setSpeed(0);

motor2.setSpeed(0);
motor.forward();
motor2.forward();
}
else {
digitalWrite(Buzzer,LOW);
}
if (distance1 >= 180 || distance1 <= 0)
{
Serial.println("Out of range1");
}
else {
Serial.print(distance1);
Serial.println(" cm");
}

```

## ***LISTING PROGRAM***

KETERANGAN

A4      No.05



TEK. ELEKTRONIKA D3  
FAKULTAS TEKNIK  
UNIV. NEGERI YOGYAKARTA

SKA. 1:1

DIS. SRI W

DIG. AFFIN

DIP. SRI W

NIM. 15507134002

```

delay(1000);                                }

}
digitalWrite(trigPin2, LOW);
delayMicroseconds(2);
digitalWrite(trigPin2, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin2, LOW);
delayMicroseconds(10);
duration2 = pulseIn(echoPin2, HIGH);
distance2 = (duration2/2) / 29.1;
delay(1000);
delay(100);

if (distance2 < 30){
digitalWrite(Buzzer,HIGH);                  maju ();
motor.setSpeed(0);                         mundur();
motor2.setSpeed(0);                        kiri ();
motor.forward();                          kanan ();
motor2.forward();                         berhenti ();
}

else {
digitalWrite(Buzzer,LOW);
}

```

<b><i>LISTING PROGRAM</i></b>		KETERANGAN	
A4	No.05		
 TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. 1:1 DIG. AFFIN	DIS. SRI W DIP. SRI W	NIM. 15507134002

```

void maju (){

    int flexADC0 = analogRead(FLEX_PIN0);           if(flexV1<=4.1){

        float flexV0 = flexADC0 * VCC / 1023.0;      // posisi nol atau tidak jalan maju/ mundur

        if(flexV0>=4.7){                           motor.setSpeed(255);

            // posisi nol atau tidak jalan maju/ mundur          motor2.setSpeed(255);

            motor.setSpeed(255);                      motor.backward();

            motor2.setSpeed(255);                     motor2.backward();

            motor.forward();                         delay(3000);

            motor2.forward();                       }

            delay(3000);                           void kanan (){

        }

    }

}

void mundur (){

    int flexADC1 = analogRead(FLEX_PIN1);           if(flexV2<=4.1){

        float flexV1 = flexADC1 * VCC / 1023.0;      // posisi nol atau tidak jalan maju/ mundur

```

<b><i>LISTING PROGRAM</i></b>			KETERANGAN	
		A4	No.05	
 TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. 1:1	DIS. SRI W		
	DIG. AFFIN	DIP. SRI W	NIM. 15507134002	

```

motor.setSpeed(0);
motor2.setSpeed(255);
motor.forward();
motor2.forward();
delay(3000);
}

void kiri (){
    int flexADC3 = analogRead(FLEX_PIN3);
    float flexV3 = flexADC3 * VCC / 1023.0;
    if(flexV3<=3.8){
        // posisi nol atau tidak jalan maju/ mundur
        motor.setSpeed(255);
        motor2.setSpeed(0);
        motor.forward();
        motor2.forward();
        delay(3000);
    }
}
}

void berhenti (){
    int flexADC0 = analogRead(FLEX_PIN0);
    float flexV0 = flexADC0 * VCC / 1023.0;
    int flexADC4 = analogRead(FLEX_PIN4);
    float flexV4 = flexADC4 * VCC / 1023.0;

    if(flexV0<=4.55&&flexV4<=3.9){
        // posisi nol atau tidak jalan maju/ mundur
        motor.setSpeed(0);
        motor2.setSpeed(0);
        motor.forward();
        motor2.forward();
        delay(3000);
    }
}

```

<b>LISTING PROGRAM</b>		KETERANGAN	
A4	No.05		
TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. 1:1	DIS. SRI W	
	DIG. AFFIN	DIP. SRI W	NIM. 15507134002

## Lampiran 6. Datasheet Sensor Flex





### FLEX SENSOR FS

**Features**

- Angle Displacement Measurement
- Bends and Flexes physically with motion device
- Possible Uses
- Robotics
- Gaming (Virtual Motion)
- Medical Devices
- Computer Peripherals
- Musical Instruments
- Physical Therapy
- Simple Construction
- Low Profile

---

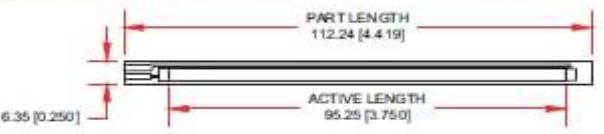
**Mechanical Specifications**

- Life Cycle: >1 million
- Height: ≤0.43mm (0.017")
- Temperature Range: -35°C to +80°C

**Electrical Specifications**

- Flat Resistance: 10K Ohms ±30%
- Bend Resistance: minimum 2 times greater than the flat resistance at 180° pinch bend (see "How It Works" below)
- Power Rating : 0.5 Watts continuous; 1 Watt Peak

**Dimensional Diagram - Stock Flex Sensor**



**How to Order - Stock Flex Sensor**

FS	—	L	—	0095	—	103	—	ST
Series	—	Model	—	Active Length	—	Resistance	—	Connectors
FS = Flex Sensor	—	L = Linear	—	0095 = 95.25mm	—	103 = 10 KOhms	—	ST = Solder Tab

**How It Works**



spectrasymbol.com
v.2014 Rev A - Page 1
(888) 795-2283

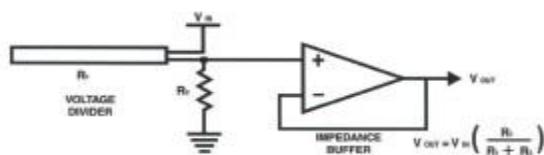
**DATASHEET SENSOR FLEX**

KETERANGAN

A4	No.06
 TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. 1:1
	DIS. SRI W
   	DIG. AFFIN
	DIP. SRI W
   	NIM. 15507134002

## Schematics

### BASIC FLEX SENSOR CIRCUIT:

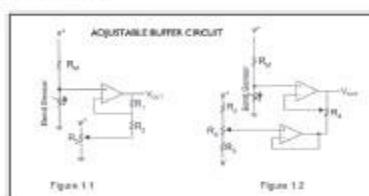


Following are notes from the ITP Flex Sensor Workshop

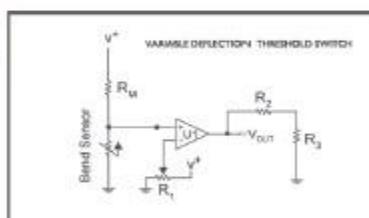
"The impedance buffer in the [Basic Flex Sensor Circuit] (above) is a single sided operational amplifier, used with these sensors because the low bias current of the op amp reduces error due to source impedance of the flex sensor as voltage divider. Suggested op amps are the LM358 or LM324."

"You can also test your flex sensor using the simplest circuit, and skip the op amp."

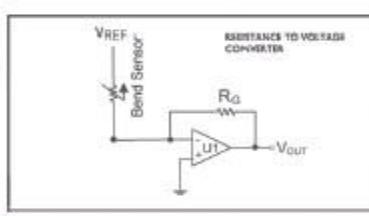
"Adjustable Buffer - a potentiometer can be added to the circuit to adjust the sensitivity range."



"Variable Deflection Threshold Switch - an op amp is used and outputs either high or low depending on the voltage of the inverting input. In this way you can use the flex sensor as a switch without going through a microcontroller."



"Resistance to Voltage Converter - use the sensor as the input of a resistance to voltage converter using a dual sided supply op-amp. A negative reference voltage will give a positive output. Should be used in situations when you want output at a low degree of bending."



DATASHEET SENSOR FLEX		KETERANGAN	
		A4	No.06
TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. 1:1 DIS. SRI W	DIG. AFFIN	DIP. SRI W NIM. 15507134002

## Lampiran 7. Datasheet Sensor Ultrasonik HC-SR04



Tech Support: services@elecfreaks.com

### Ultrasonic Ranging Module HC - SR04

#### Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) If the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

#### Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

#### Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
Measuring Angle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45×20×15mm

### DATASHEET SENSOR ULTRASONIK HC-SR04

KETERANGAN

A4 No.07



TEK. ELEKTRONIKA D3  
FAKULTAS TEKNIK  
UNIV. NEGERI YOGYAKARTA

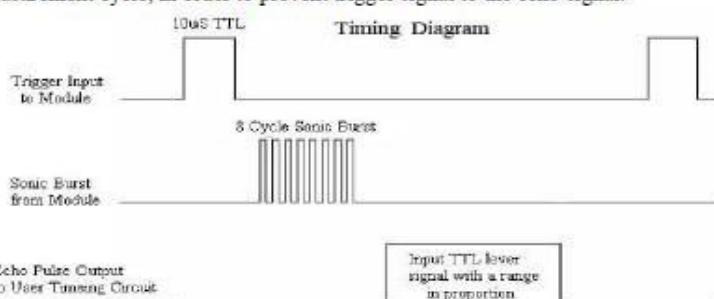
SKA. 1:1  
DIS. SRI W  
DIG. AFFIN  
DIP. SRI W

NIM. 15507134002



### Timing diagram

The Timing diagram is shown below. You only need to supply a short 10 $\mu$ s pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula:  $uS / 58 = \text{centimeters}$  or  $uS / 148 = \text{inch}$ ; or: the range = high level time \* velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



## DATASHEET SENSOR ULTRASONIK

### HC-SR04

KETERANGAN

A4 No.07



TEK. ELEKTRONIKA D3  
FAKULTAS TEKNIK  
UNIV. NEGERI YOGYAKARTA

SKA. 1:1 DIS. SRI W

DIG. AFFIN DIP. SRI W

NIM. 15507134002

### **Attention:**

- The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
- When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise ,it will affect the results of measuring.

[www.ElecFreaks.com](http://www.ElecFreaks.com)



<b>DATASHEET SENSOR ULTRASONIK HC-SR04</b>		KETERANGAN	
 TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. 1:1	DIS. SRI W	A4      No.07
	DIG. AFFIN	DIP. SRI W	NIM. 15507134002

## Lampiran 8. Datasheet Driver Motor L298N

**L298 Dual H-Bridge Motor Driver**      User's Guide

---

### Overview

The Motor Shield is based on the L298, which is a dual full-bridge driver designed to drive inductive loads such as relays, solenoids, DC and stepping motors. It lets you drive two DC motors, controlling the speed and direction of each one independently.

### Summary

Operating Voltage: 4V to 35V  
 Motor controller L298N, Drives 2 DC motors or 1 stepper motor  
 Max current: 2A per channel or 4A max  
 Free running stop and brake function  
 Chip: ST L298N  
 Logic power supply: 5V  
 Max power: 25W  
 Weight: 35g  
 Size: 55mm x 60mm x 30mm  
 Storage temperature: -25 to +135°C




---

**L298 Dual H-Bridge Motor Driver**      User's Guide

---

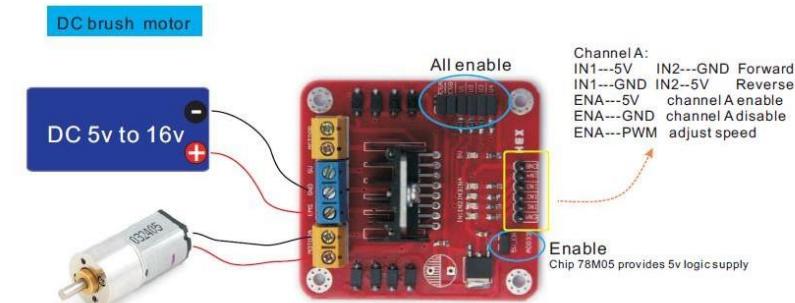
	<p>CSA: Between this pin and ground is connected the sense resistor to control the current of the load.          Enable-----Ignore current detection function</p> <p>CSB: Between this pin and ground is connected the sense resistor to control the current of the load.          Enable-----Ignore current detection function</p>	
	<p>5V-EN: Enable----78M05 works, output DC 5V          Disable----78M05 does not work. Need input DC 5V          The module needs DC 5V always, for logic supply.</p>	
	<p>The pull-up resistor is enabled.          U1---Enable In1 pull-up resistor [10k].          U2---Enable In2 pull-up resistor [10k].          U3---Enable In3 pull-up resistor [10k].          U4---Enable In4 pull-up resistor [10k].</p>	
	<p>IN1 IN2 : TTL Compatible Inputs of the Bridge A          IN3 IN4 : TTL Compatible Inputs of the Bridge B.          ENA ENB: TTL Compatible Enable Input: the L state disables the bridge A(enable A) and/or the bridge B (enable B).</p>	<p>5V_EN:          Enable: [5V] can output DC 5V.          Disable:[5V] need input DC 5V.</p>

---

<b>DATASHEET DRIVER MOTOR L298N</b>		KETERANGAN	
 <b>TEK. ELEKTRONIKA D3</b> <b>FAKULTAS TEKNIK</b> <b>UNIV. NEGERI YOGYAKARTA</b>	<b>SKA. 1:1</b>	<b>DIS. SRI W</b>	<b>A4</b>
<b>DIG. AFFIN</b>	<b>DIP. SRI W</b>	<b>NIM. 15507134002</b>	

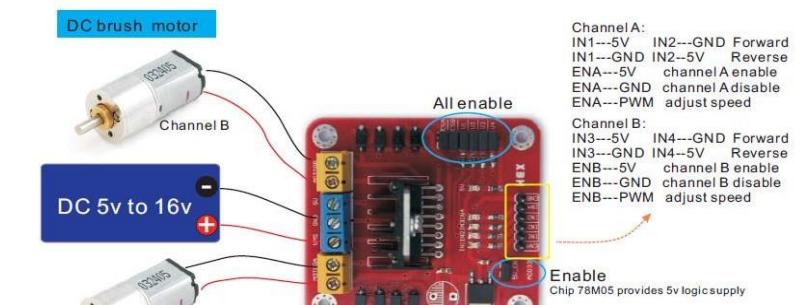
## L298 Dual H-Bridge Motor Driver

User's Guide



## L298 Dual H-Bridge Motor Driver

User's Guide



## DATASHEET DRIVER MOTOR L298N

KETERANGAN

A4 No.08



TEK. ELEKTRONIKA D3  
FAKULTAS TEKNIK  
UNIV. NEGERI YOGYAKARTA

SKA. 1:1

DIS. SRI W

DIG. AFFIN

DIP. SRI W

NIM. 15507134002

## L298 Dual H-Bridge Motor Driver

User's Guide

Example Code :

Platform: Arduino  
Target: 4-wire 2-phase stepper motor

```
#include <Stepper.h>
//ENA=9 IN1=2 IN2=3 IN3=4 IN4=5 ENB=10
// change this to the number of steps on your motor
#define STEPS 200
// create an instance of the stepper class, specifying
// the number of steps of the motor and the pins it's
// attached to.
Stepper stepper(STEPS, 2, 3, 4, 5);
void setup()
{
  stepper.setSpeed(300);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
}
void loop()
{
  digitalWrite(8, HIGH);
  stepper.step(50);
  delay(500);
  stepper.step(-50);
  delay(500);
  stepper.step(200);
  delay(500);
  stepper.step(-200);
  delay(500);
}
```



Platform: Arduino  
Target: 2-wire DC brush motor

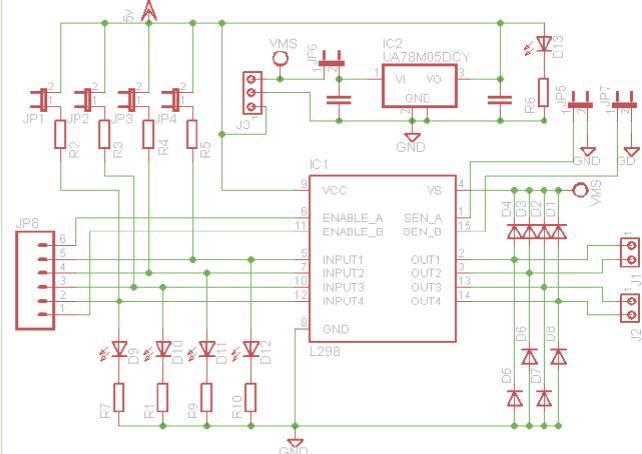
```
// motor A
int dir1PinA = 2;
int dir2PinA = 3;
int speedPinA = 9;
// motor B
int dir1PinB = 4;
int dir2PinB = 5;
int speedPinB = 10;
unsigned long time;
int speed;
int dir;
void setup() {
  pinMode(dir1PinA, OUTPUT);
  pinMode(dir2PinA, OUTPUT);
  pinMode(speedPinA, OUTPUT);
  pinMode(dir1PinB, OUTPUT);
  pinMode(dir2PinB, OUTPUT);
  pinMode(speedPinB, OUTPUT);
  time = millis();
  speed = 0;
  dir = 1;
}
void loop() {
  digitalWrite(speedPinA, speed);
  digitalWrite(speedPinB, 255 - speed);
  // set direction
  if (1 == dir) {
    digitalWrite(dir1PinA, LOW);
    digitalWrite(dir2PinB, HIGH);
    digitalWrite(dir1PinB, HIGH);
    digitalWrite(dir2PinB, LOW);
  } else {
    digitalWrite(dir1PinA, HIGH);
    digitalWrite(dir2PinA, LOW);
    digitalWrite(dir1PinB, LOW);
    digitalWrite(dir2PinB, HIGH);
  }
  if (millis() - time > 5000) {
    time = millis();
    speed += 20;
    if (speed > 255) {
      speed = 0;
    }
    if (1 == dir) {
      dir = 0;
    } else {
      dir = 1;
    }
  }
}
```



## L298 Dual H-Bridge Motor Driver

User's Guide

Schematic:



## DATASHEET DRIVER MOTOR L298N

KETERANGAN

A4 No.08



TEK. ELEKTRONIKA D3  
FAKULTAS TEKNIK  
UNIV. NEGERI YOGYAKARTA

SKA. 1:1

DIS. SRI W

DIG. AFFIN

DIP. SRI W

NIM. 15507134002

## Lampiran 9. Datasheet Arduino Uno

### Arduino UNO



The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, 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 Uno 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.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [Index of Arduino boards](#).

[Index](#)

<a href="#">Technical Specifications</a>	Page 2
<a href="#">How to use Arduino Programming Environment, Basic Tutorials</a>	Page 6
<a href="#">Terms &amp; Conditions</a>	Page 7
<a href="#">Environmental Policies half sqm of green via Impatto Zero®</a>	Page 7

<b>DATASHEET ARDUINO UNO</b>		KETERANGAN	
		A4	No.09
 TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. 1:1	DIS. SRI W	
	DIG. AFFIN	DIP. SRI W	NIM. 15507134002

## Technical Specification

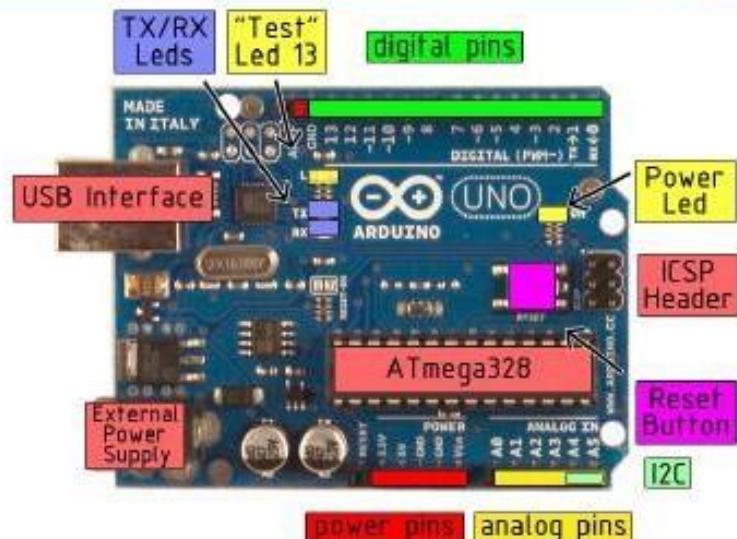


EAGLE file: [arduino-dreamland-uno-design.eagle](#) Schematic: [arduino-uno-schematic.pdf](#)

### Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

### the board



Radiospares RADIONICS



## DATASHEET ARDUINO UNO

KETERANGAN

A4 No.09



TEK. ELEKTRONIKA D3  
FAKULTAS TEKNIK  
UNIV. NEGERI YOGYAKARTA

SKA. 1:1 DIS. SRI W

DIG. AFFIN DIP. SRI W

NIM. 15507134002

## Power

The Arduino Uno 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 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 Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.



**Radiospares**

**RADIONICS**



KETERANGAN

A4 No.09

## DATASHEET ARDUINO UNO



TEK. ELEKTRONIKA D3  
FAKULTAS TEKNIK  
UNIV. NEGERI YOGYAKARTA

SKA. 1:1	DIS. SRI W
DIG. AFFIN	DIP. SRI W

NIM. 15507134002

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- I<sup>2</sup>C: 4 (SDA) and 5 (SCL). Support I<sup>2</sup>C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and Atmega328 ports](#).

## Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an ".inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I<sup>2</sup>C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I<sup>2</sup>C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

## Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the Tools > Board menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).



**Radiospares**

**RADIONICS**



ARMADA INDONESIA

## DATASHEET ARDUINO UNO

KETERANGAN

A4 No.09



TEK. ELEKTRONIKA D3  
FAKULTAS TEKNIK  
UNIV. NEGERI YOGYAKARTA

SKA. 1:1

DIS. SRI W

DIG. AFFIN

DIP. SRI W

NIM. 15507134002

### Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

### USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

### Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



**radiospares**

**RADIONICS**



## DATASHEET ARDUINO UNO

KETERANGAN

A4 No.09



TEK. ELEKTRONIKA D3  
FAKULTAS TEKNIK  
UNIV. NEGERI YOGYAKARTA

SKA. 1:1

DIS. SRI W

DIG. AFFIN

DIP. SRI W

NIM. 15507134002

## 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 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

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



Done compiling

Press Compile button  
(to check for errors)



Upload



TX RX Flashing



Blinking Led!



Radiospares

RADIONICS



## DATASHEET ARDUINO UNO

KETERANGAN

A4 No.09



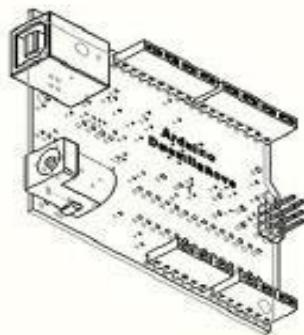
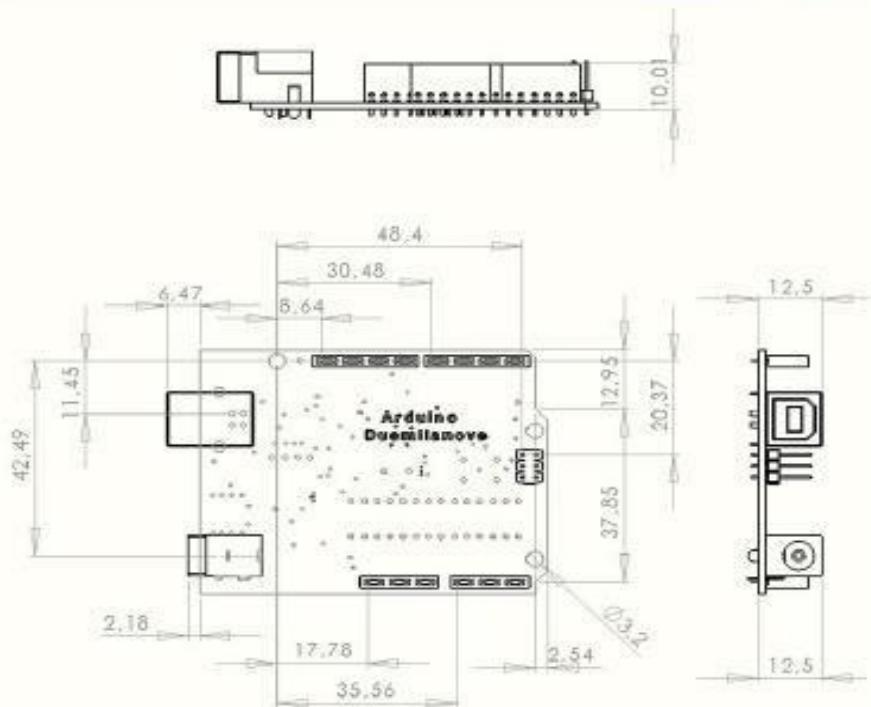
TEK. ELEKTRONIKA D3  
FAKULTAS TEKNIK  
UNIV. NEGERI YOGYAKARTA

SKA. 1:1 DIS. SRI W

DIG. AFFIN DIP. SRI W

NIM. 15507134002

Dimensioned Drawing



*radiospares*

*RADIONICS*



DATASHEET ARDUINO UNO			KETERANGAN	
 TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. 1:1	DIS. SRI W	A4	No.09
	DIG. AFFIN	DIP. SRI W	NIM. 15507134002	

# Terms & Conditions



## 1. Warranties

1.1 The producer warrants that its products will conform to the specifications. This warranty lasts for one (1) year from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or misapplication 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 representations and warranties made under this terms and conditions or (ii) the gross negligence or wilful 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**



## DATASHEET ARDUINO UNO

KETERANGAN

A4 No.09



TEK. ELEKTRONIKA D3  
FAKULTAS TEKNIK  
UNIV. NEGERI YOGYAKARTA

SKA. 1:1	DIS. SRI W
DIG. AFFIN	DIP. SRI W

NIM. 15507134002