

**RANCANG BANGUN ROBOT OMNI WHEEL**

**PENGIKUT DINDING**

**SKRIPSI**

**TEKNIK ELEKTRO KONSENTRASI TEKNIK ELEKTRONIKA**

Ditujukan untuk memenuhi persyaratan  
memperoleh gelar Sarjana Teknik



**BRIAN REZA KAWALTA TARIGAN SIBERO**

**NIM. 125060301111031**

**UNIVERSITAS BRAWIJAYA**

**FAKULTAS TEKNIK**

**MALANG**

**2017**

LEMBAR PENGESAHAN

RANCANG BANGUN ROBOT *OMNI WHEEL* PENGIKUT  
DINDING

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI TEKNIK ELEKTRONIKA

Ditujukan untuk memenuhi persyaratan  
memperoleh gelar Sarjana Teknik



**BRIAN REZA KAWALTA TARIGAN SIBERO**

**NIM. 125060301111031**

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing  
pada tanggal ..... 2017

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Ir. Ponco Siwindarto, M.Eng.Sc.  
NIP. 19590304 198903 1 001

Raden Arief Setyawan, S.T., M.T.  
NIP. 19750819 199903 1 001

Mengetahui,

Ketua Jurusan Teknik Elektro

M. Aziz Muslim, S.T., M.T., Ph.D.  
NIP. 19741203 200012 1 001

JUDUL SKRIPSI:

RANCANG BANGUN ROBOT *OMNI WHEEL* PENGIKUT DINDING

Nama Mahasiswa : BRIAN REZA KAWALTA TARIGAN SIBERO

NIM : 125060301111031

Program Studi : TEKNIK ELEKTRO

Konsentrasi : TEKNIK ELEKTRONIKA

Komisi Pembimbing :

Ketua : Dr. Ir. Ponco Siwindarto, M.Eng.Sc. ....

Anggota : Raden Arief Setyawan, S.T., M.T. ....

Tim Dosen Pengaji : ....

Dosen Pengaji 1 : Adharul Muttaqin, S.T., M.T. ....

Dosen Pengaji 2 : Akhmad Zainuri, S.T., M.T. ....

Dosen Pengaji 3 : Dr. Eng. Panca Mudjirahardjo, S.T., M.T. ....

Tanggal Ujian : 27 Januari 2017

SK Pengaji : 173/UN10.F07/ SK/ 2017

## **PERNYATAAN ORISINALITAS SKRIPSI**

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Malang, 10 Januari 2017

**Mahasiswa,**

**Brian Reza Kawalta Tarigan Sibero**

**NIM. 125060301111031**

*Teriring Ucapan Terima Kasih kepada:*

*Bapak dan Mamak Tersayang*

*Adek-Adekkku Tercinta*

**UNIVERSITAS BRAWIJAYA**



*Pengataken Bujur Man*

*Bapak, Mamak ras Agi-Agiku*

## RINGKASAN

**Brian Reza Kawalta Tarigan Sibero**, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Januari 2017, *Rancang Bangun Robot Omni Wheel Pengikut Dinding*, Dosen Pembimbing : Ponco Siwindarto dan R. Arief Setyawan.

Kontes Robot Pemadam Api Indonesia merupakan divisi dimana robot memiliki misi mencari dan memadamkan api di arena lapangan berbentuk simulasi interior suatu rumah. Robot yang berhasil menemukan dan memadamkan api tercepat dinyatakan sebagai pemenang. Sehingga diperlukan sebuah mekanik yang dapat bermanuver dan bernavigasi dengan cepat.

Dalam penelitian ini robot *Omni Wheel* yang diciptakan menggunakan mikrokontroler Arduino Mega 2560 dengan spesifikasi dimensi robot yakni panjang 21 centimeter, lebar 23 centimeter, dan tinggi 17 centimeter dengan menggunakan roda sebagai penggeraknya. Roda yang digunakan adalah jenis *Omni Wheel* dengan *platform* segitiga dimana pemasangan antar *omni wheel* berjarak  $120^\circ$  dengan menggunakan motor DC *brushed* sehingga Robot dapat bergerak ke segala arah tanpa memutar badan robot terlebih dahulu. Pengujian ini dilakukan hanya dengan bernavigasi lurus sejauh 7 meter sehingga Sistem Navigasi yang digunakan yakni metode *Wall Following* dengan penggunaan sensor ultrasonik PING))) sebagai pembacaan jarak.

Hasil pengujian jarak sensor ultrasonik PING))) memiliki rata-rata kesalahan pengukuran sebesar 0,568 cm. Untuk hasil pengujian keseluruhan sistem dimana waktu rata-rata dari 40 kali percobaan mengikuti dinding sepanjang 7 meter adalah 13,7225 detik dengan rata-rata keseluruhan osilasi sebesar 15,572 cm.

Kata kunci: Arduino Mega 2560, *Platform Omni Wheel*, PING))), *Wall Following*



## SUMMARY

**Brian Reza Kawalta Tarigan Sibero**, *Department of Electrical Engineering, Faculty of Engineering, University of Brawijaya, January 2017, Design Build of the Omni Wheel Robot Followers Wall, Academic Supervisor : Ponco Siwindarto and R. Arief Setyawan.*

*Fire Fighting Indonesian Robot Contest is a division where the robot has a mission to locate and extinguish the fire in arena form of simulated interior of a house. The robot which can successfully locate and extinguish fires fastest is declared the winner. So, we need a mechanic who can maneuver and navigate quickly.*

*In this study Omni Wheel robot that was created using a microcontroller Arduino Mega 2560 with the robot dimensions specification 21 centimeters long, 23 centimeters wide and 17 centimeters high by using the wheel as the driving force. The wheels used were kind of Omni Wheel with triangular platform where an omni wheel mounting within 120° using brushed DC motors so that the robot can move in any direction without turning the robot body first. This testing is done by simply navigating straight as far as 7 meters so that navigation systems that used the method Wall Following with the use of ultrasonic sensors PING))) as the reading distance.*

*Results of testing ultrasonic distance sensor PING))) have an average error of measurement of 0.568 cm. For the entire testing system where the average time of 40 attempts to follow the wall along the 7 meter is 13.7225 seconds with an overall average of oscillation of 15.572 cm.*

Keywords: Arduino Mega 2560, Platform Omni Wheel, PING))), Wall Following.



## PENGANTAR

Puji dan Syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena atas segala petunjuk dan berkatNya lah skripsi ini dapat diselesaikan. Skripsi berjudul "*RANCANG BANGUN ROBOT OMNI WHEEL PENGIKUT DINDING*" ini disusun untuk memenuhi persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Universitas Brawijaya.

Penulis menyadari bahwa dalam penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada:

- ) Tuhan Yang Maha Esa yang telah memberikan kasihNya, kemudahan dan berkatNya.
- ) Kedua orang tua Bapak Darta Tarigan dan Ibu Herlianta br Ginting selaku orang tua, Tifani br Tarigan dan Bima Josua Tarigan selaku adik penulis, atas segala kasih, inspirasi, dukungan, nasehat, perhatian, dan kesabarannya kepada penulis, serta telah banyak mendoakan kelancaran penulis sehingga terselesaikannya skripsi ini.
- ) Bapak M. Aziz Muslim, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya.
- ) Bapak Hadi Suyono, S.T., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya.
- ) Bapak Dr. Ir. Ponco Siwindarto, M.Eng.Sc. selaku Dosen Pembimbing I dan Bapak R. Arief Setyawan, S.T., M.T selaku Dosen Pembimbing II atas segala bimbingan, pengarahan, saran, dan kritik yang telah diberikan selama proses pengerjaan skripsi.
- ) Ibu Ir. Nurussa'adah, MT. selaku Ketua Kelompok Dosen Keahlian Elektronika Jurusan Teknik Elektro Universitas Brawijaya yang telah meluangkan banyak waktu di sela-sela kesibukan beliau dan atas segala bimbingan, pengarahan, saran, dan kritik yang telah diberikan selama perkuliahan.
- ) Seluruh dosen pengajar Jurusan Teknik Elektro Universitas Brawijaya.
- ) Seluruh staff Jurusan Teknik Elektro Universitas Brawijaya.



- ) Keluarga Ginting dan Tarigan Mergana di Pancur Batu dan Gedung Johor yang selalu memberikan semangat, motivasi, bimbingan selama berkuliah di Universitas Brawijaya.
- ) Permata GBKP Perpulungan Malang terutama Riski Manta Simanjorang (Preman 2 Tlogomas) yang sudah menjadi keluarga yang selalu memotivasi, membantu, dan dukungannya dalam membantu penyelesaian skripsi.
- ) Permata 2012, Pebrianti, Hanis, Dorsay, Elisa, Vera, Eva, Ita, Amo, Aga, Kristo, Krisna, Oga, Brian, Adena yang selalu memberikan dukungan dan cintanya dalam sama-sama menyelesaikan perkuliahan di Universitas Brawijaya.
- ) Turang Galang-Kitik, Dian, Pebri, Reli, Keling, Listra, Jupe, Amo, Wina, Dora, Inke, Desi, Felisia, Elmi, Elva, Gita, Liri, Icha, Magda, Maria, Yolanda, Devi, Enda, dan Feber yang selalu memberikan masukan, candaan, bantuan, dan menemani saat jomblo.
- ) Keluarga “Robot Rolas” atau asisten Laboratorium Mekaronika dan Robotika, Ricky, Sofyan, Fikrul, Agus, Ronny, Juli, Firman, Wiwin, Septian, Anggi atas segala dukungan dan segala pengalaman, kebersamaan dan bantuan selama 4 setengah tahun ini.
- ) Adik-adik TIM Robot 2013, 2014, dan 2015 yang selalu mendukung, membantu, dan menemani selama ini.
- ) Saudara - saudari “Voltage” dan “Voctron Elektronika” angkatan 2012 atas segala bantuan dan kebersamaan yang telah diberikan selama 4 setengah tahun ini.
- ) Semua pihak yang telah memberikan bantuan serta dukungan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, penulis sangat memerlukan dan mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi serta bagi masyarakat.

Malang, 10 Januari 2017

Penulis

## DAFTAR ISI

<b>PENGANTAR .....</b>	i
<b>DAFTAR ISI .....</b>	iii
<b>DAFTAR GAMBAR .....</b>	v
<b>DAFTAR TABEL .....</b>	vii
<b>DAFTAR LAMPIRAN.....</b>	ix
<b>BAB I : PENDAHULUAN .....</b>	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	2
1.5 Manfaat .....	2
<b>BAB II : TINJAUAN PUSTAKA .....</b>	3
2.1 Lapangan Kontes Robot Pemadam Api Indonesia 2013 .....	3
2.2 <i>Arduino Mega 2560</i> .....	3
2.3 <i>Omni Wheel</i> .....	4
2.4 Sensor Ultrasonik PING))) .....	5
2.5 Motor DC <i>Brushed</i> .....	6
2.6 <i>L298 Dual H – Bridge Driver Motor</i> .....	8
2.7 Navigasi Wall Following .....	9
2.8 <i>Omni Wheel Robot</i> .....	10
<b>BAB III : METODE PENELITIAN .....</b>	11
3.1 Spesifikasi.....	11
3.2 Perancangan Alat ( Desain Alat) .....	11
3.3 Pengujian Alat .....	16
<b>BAB IV : HASIL DAN PEMBAHASAN.....</b>	19
4.1 Pengujian Sensor Ultrasonik.....	19
4.2 Pengujian <i>Driver</i> Pengendali Motor DC .....	26
4.3 Pengujian Keseluruhan Sistem .....	29
<b>BAB V : KESIMPULAN DAN SARAN .....</b>	51
5.1 Kesimpulan .....	51
5.2 Saran .....	51



**Daftar Pustaka .....** ..... 52



## DAFTAR GAMBAR

Gambar 2.1 Lapangan KRPAI 2013 .....	3
Gambar 2.2 Board <i>Arduino Mega 2560</i> .....	4
Gambar 2.3 <i>Omni Wheel</i> .....	4
Gambar 2.4 Sensor Ultrasonik PING))).....	5
Gambar 2.5 Ilustrasi Cara Kerja Modul PING))).....	5
Gambar 2.6 Komunikasi Mikrokontroler dengan PING))).....	6
Gambar 2.7 Ilustrasi Motor DC <i>Brushed</i> .....	7
Gambar 2.8 L298N Dual H – Bridge Driver Motor .....	8
Gambar 2.9 Navigasi Wall Following.....	9
Gambar 2.10 Bentuk <i>platform</i> segitiga menggunakan tiga buah <i>omni wheel</i> dengan perpindahan badan <i>robot</i> .....	10
Gambar 3.1 Blok Diagram Sistem .....	12
Gambar 3.2 Mekanika <i>Robot</i> .....	13
Gambar 3.3 Jembatan <i>Shield Driver</i> L298N.....	14
Gambar 3.4 Diagram Alir Robot.....	15
Gambar 3.5 Blok Diagram Pengujian Sensor Ultrasonik PING))) .....	16
Gambar 3.6 Blok Diagram Pengujian Pengendalian Arah Putar Motor DC .....	17
Gambar 3.7 Blok Diagram Pengujian Respon <i>Driver Motor</i> L298N terhadap Masukan Sinyal PWM .....	17
Gambar 4.1 Blok Diagram Pengujian Sensor Ultrasonik PING))) .....	19
Gambar 4.2 Tampilan Hasil Pengujian pada Jarak 5 cm (a) Menggunakan Penggaris (b) Menggunakan <i>Serial Monitor</i> .....	21
Gambar 4.3 Tampilan Hasil Pengujian pada Jarak 10 cm (a) Menggunakan Penggaris (b) Menggunakan <i>Serial Monitor</i> .....	22
Gambar 4.4 Tampilan Hasil Pengujian pada Jarak 15 cm (a) Menggunakan Penggaris (b) Menggunakan <i>Serial Monitor</i> .....	23



Gambar 4.5 (Tampilan Hasil Pengujian pada Jarak 20 cm	
(a) Menggunakan Penggaris	
(b) Menggunakan <i>Serial Monitor</i> .....	24
Gambar 4.6 Tampilan Hasil Pengujian pada Jarak 25 cm	
(c) Menggunakan Penggaris	
(d) Menggunakan <i>Serial Monitor</i> .....	25
Gambar 4.7 Sinyal Keluaran <i>Driver L298N</i> terhadap Sinyal Masukan	
PWM MK Sebesar 50% .....	28
Gambar 4.8 Robot Menelusuri Objek Dinding Menggunakan	
Ultrasonik Depan Dengan Gerak Ke Kanan .....	30
Gambar 4.9 Robot Menelusuri Objek Dinding Menggunakan	
Ultrasonik Depan Dengan Gerak Ke Kiri .....	32
Gambar 4.10 Robot Menelusuri Objek Dinding Menggunakan	
Ultrasonik Kanan Dengan Gerak Maju.....	35
Gambar 4.11 Robot Menelusuri Objek Dinding Menggunakan	
Ultrasonik Kanan Dengan Gerak Mundur .....	37
Gambar 4.12 Robot Menelusuri Objek Dinding Menggunakan	
Ultrasonik Kiri Dengan Gerak Maju.....	40
Gambar 4.13 Robot Menelusuri Objek Dinding Menggunakan	
Ultrasonik Kiri Dengan Gerak Mundur .....	42
Gambar 4.14 Robot Menelusuri Objek Dinding Menggunakan	
Ultrasonik Belakang Dengan Gerak Ke Kanan .....	45
Gambar 4.15 Robot Menelusuri Objek Dinding Menggunakan	
Ultrasonik Belakang Dengan Gerak Ke Kiri .....	48



## DAFTAR TABEL

Tabel 4.1 Hasil Pengujian Sensor Ultrasonik PING))) .....	20
Tabel 4.2 Hasil Pengujian Pengendalian Arah Putar Motor DC 1 .....	26
Tabel 4.3 Hasil Pengujian Pengendalian Arah Putar Motor DC 2.....	27
Tabel 4.4 Hasil Pengujian Pengendalian Arah Putar Motor DC 3.....	27
Tabel 4.5 Hasil Pengujian <i>Duty Cycle Driver</i> L298N terhadap Sinyal Masukan PWM .....	28
Tabel 4.6 Hasil Osilasi dan Standar Deviasi Robot Menggunaan Ultrasonik Depan dengan Gerak ke Kanan .....	31
Tabel 4.7 Hasil Pengujian Robot Menggunakan Ultrasonik Depan dengan Gerak Ke Kanan .....	31
Tabel 4.8 Hasil Osilasi dan Standar Deviasi Robot Menggunakan Ultrasonik Depan dengan Gerak ke Kiri .....	33
Tabel 4.9 Hasil Pengujian Robot Menggunakan Ultrasonik Depan dengan Gerak Ke Kiri .....	34
Tabel 4.10 Hasil Osilasi dan Standar Deviasi Robot Menggunakan Ultrasonik Kanan dengan Gerak Maju.....	36
Tabel 4.11 Hasil Pengujian Robot Menggunakan Ultrasonik Kanan dengan Gerak Maju .....	36
Tabel 4.12 Hasil Osilasi dan Standar Deviasi Robot Menggunakan Ultrasonik Kanan dengan Gerak Mundur .....	38
Tabel 4.13 Hasil Pengujian Robot Menggunakan Ultrasonik Kanan dengan Gerak Mundur.....	39
Tabel 4.14 Hasil Osilasi dan Standar Deviasi Robot Menggunakan Ultrasonik Kiri dengan Gerak Maju.....	40
Tabel 4.15 Hasil Pengujian Robot Menggunakan Ultrasonik Kiri dengan Gerak Maju .....	41
Tabel 4.16 Hasil Osilasi dan Standar Deviasi Robot Menggunakan Ultrasonik Kiri dengan Gerak Mundur .....	44
Tabel 4.17 Hasil Pengujian Robot Menggunakan Ultrasonik Kiri dengan Gerak ke Mundur.....	44

Tabel 4.18 Hasil Osilasi dan Standar Deviasi Robot Menggunakan Ultrasonik Belakang dengan Gerak Kanan .....	46
Tabel 4.19 Hasil Pengujian Robot Menggunakan Ultrasonik Belakang dengan Gerak ke Kanan .....	47
Tabel 4.20 Hasil Osilasi dan Standar Deviasi Robot Menggunakan Ultrasonik Belakang dengan Gerak ke Kiri .....	49
Tabel 4.21 Hasil Pengujian Robot Menggunakan Ultrasonik Belakang dengan Gerak Ke Kiri .....	49



## DAFTAR LAMPIRAN

### Judul

LAMPIRAN 1. DOKUMENTASI ALAT .....	53
LAMPIRAN 2. RANGKAIAN <i>SHIELD PCB</i> .....	56
LAMPIRAN 3. LISTING PROGRAM .....	58
LAMPIRAN 4. DATASHEET .....	70



### 1.1 Latar Belakang

Kontes Robot Indonesia (KRI) adalah kegiatan kontes bidang robotika yang dapat diikuti tim mahasiswa dari institusi atau Perguruan Tinggi Negeri dan Swasta yang terdaftar di Kementerian Riset, Teknologi dan Pendidikan Tinggi. Pada KRI terdapat 5 (lima) divisi, yaitu: Kontes Robot ABU Indonesia (KRAI), Kontes Robot Pemadam Api Indonesia (KRPAI) tipe beroda, Kontes Robot Pemadam Api Indonesia (KRPAI) tipe berkaki, Kontes Robot Seni Tari Indonesia (KRSTI), dan Kontes Robot Sepak Bola Indonesia (KRSBI).

Kontes Robot Pemadam Api Indonesia Divisi Beroda merupakan suatu divisi dimana robot menggunakan roda sebagai alat geraknya dengan misi mencari dan memadamkan api di arena lapangan berbentuk simulasi interior suatu rumah. Pada divisi ini yang diutamakan adalah kemampuan robot bernavigasi dan bermanuver serta kecepatan dalam menyelesaikan misinya tersebut. Robot yang berhasil menemukan dan memadamkan api tercepat dinyatakan sebagai pemenang. (Dirlitabmas Dikti,2012)

Pada Kontes Robot Pemadam Api Indonesia Divisi Beroda, robot wajib menggunakan roda dalam pergerakannya untuk menyusuri ruangan, lorong dan menyelesaikan misinya sehingga diperlukan sebuah mekanik yang terfokus pada roda yang dapat bekerja tanpa menghambat kecepatan robot dalam bermanuver sehingga digunakan *Omni Wheel* yang dapat bergerak tanpa memutar badan robot terlebih dahulu. Dalam hal ini akan digunakan *Omni Wheel Robot* dengan *platform* segitiga dengan pemasangan menggunakan motor dc. Selain dari sisi mekanik, untuk dapat menyusuri lorong maka sebuah robot dilengkapi dengan sensor yang dapat mendeteksi keberadaan dinding sehingga digunakanlah sensor ultrasonik *PING*)) dimana sistem pergerakan yang akan dilakukan yakni dengan mengikuti sisi dinding (*wall following*).

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dibahas di atas maka dapat disusun rumusan masalah sebagai berikut :

- 1) Bagaimana hasil pengukuran sensor ultrasonik.
- 2) Bagaimana respon sinyal keluaran driver motor terhadap masukan pwm mikrokontroler.
- 3) Bagaimana merancang robot *omni wheel* dengan *platform* segitiga.
- 4) Bagaimana rata-rata lama waktu dan jarak osilasi robot *omni wheel* dalam mengikuti dinding sejauh 7 meter.

## 1.3 Batasan Masalah

Dengan mengacu pada rumusan masalah yang telah dibuat, maka hal-hal yang berkaitan dengan perancangan akan diberi batasan agar tidak terjadi banyak kemungkinan yang terjadi yaitu:

- 1) Sistem bersifat *close loop*, sebatas bennavigasi.
- 2) *Mobile Robot* tidak menggunakan kontroler.
- 3) Sensor ultrasonik yang dipakai adalah sensor ultrasonik tipe PING))) untuk mengukur jarak.
- 4) Pengujian dilakukan pada lintasan lurus yang memiliki dinding padat sepanjang 7 meter.

## 1.4 Tujuan

Tujuan penelitian ini adalah untuk membuktikan robot *omni wheel* dapat bennavigasi mengikuti dinding.

## 1.5 Manfaat

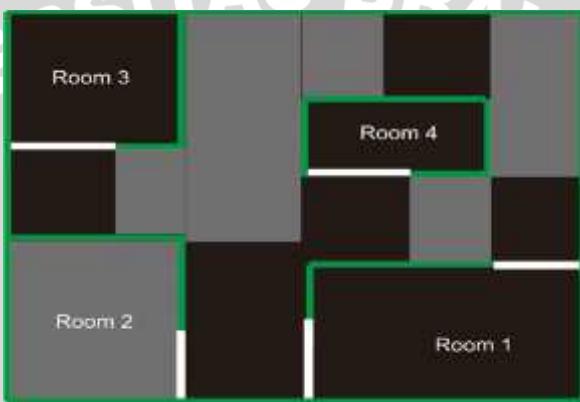
Manfaat dari penelitian ini adalah untuk membantu pembuatan robot KRPAI menggunakan *Omni Wheel*.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Lapangan Kontes Robot Pemadam Api Indonesia 2013

Lapangan/arena mensimulasikan interior dari sebuah rumah dengan 4 ruangan. Lapangan terbuat dari papan multipleks dengan ketebalan 1,8 s.d. 2cm dan berukuran 248cm x 248cm x 30cm. Di dalam lapangan terdapat 4 ruangan dengan posisi tetap namun dua diantaranya (ruang1 dan 4, lihat gambar 2.1) memiliki pintu yang dapat digeser posisinya.

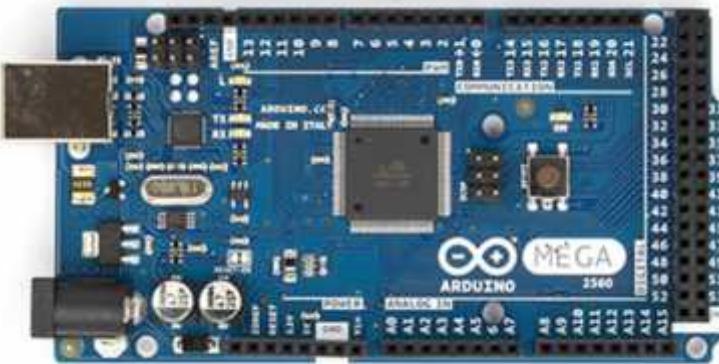


Gambar 2.1 Lapangan KRPAI 2013

Sumber: (Dirlitabmas Dikti.2012)

#### 2.2 Arduino Mega 2560

*Arduino Mega* adalah papan mikrokontroler berbasiskan ATmega2560. *Arduino Mega* 2560 memiliki 54 pin digital input/output, dimana 15 pin dapat digunakan sebagai output PWM, 16 pin sebagai input analog, dan 4 pin sebagai UART (port serial hardware), 16 MHz kristal osilator, koneksi USB, jack power, header ICSP, dan tombol reset. Ini semua yang diperlukan untuk mendukung mikrokontroler. Cukup dengan menghubungkannya ke komputer melalui kabel USB atau power dihubungkan dengan adaptor AC-DC atau baterai untuk mulai mengaktifkannya. *Arduino Mega* 2560 kompatibel dengan sebagian besar shield yang dirancang untuk *Arduino Duemilanove* atau *Arduino Diecimila*. *Arduino Mega* 2560 adalah versi terbaru yang menggantikan versi *Arduino Mega* yang ditunjukkan pada gambar 2.2.



Gambar 2.2 Board Arduino Mega 2560

Sumber: [www.arduino.cc](http://www.arduino.cc)

### 2.3 *Omni Wheel*

*Omni wheel* adalah rancangan roda khusus yang tidak hanya mempunyai roda tunggal, tetapi banyak roda dalam satu roda inti. Ada roda inti besar, dan sepanjang tepi ada banyak roda kecil tambahan yang mempunyai sumbu tegak lurus terhadap sumbu roda inti. Roda omni telah digunakan bertahun-tahun dalam dunia robot industri dan logistik.

Roda omni atau roda poli, mirip dengan roda mecanum dengan cakram kecil di sekitar lingkar yang tegak lurus terhadap arah putar. Efeknya yaitu roda dapat digerakkan dengan kekuatan penuh, dan juga akan meluncur lateral dengan sangat mudah. Roda ini sering digunakan dalam sistem penggerak holonomic.

Gambar 2.3 *Omni Wheel*Sumber: [www.elecfreaks.com](http://www.elecfreaks.com)

## 2.4 Sensor Ultrasonik PING)))

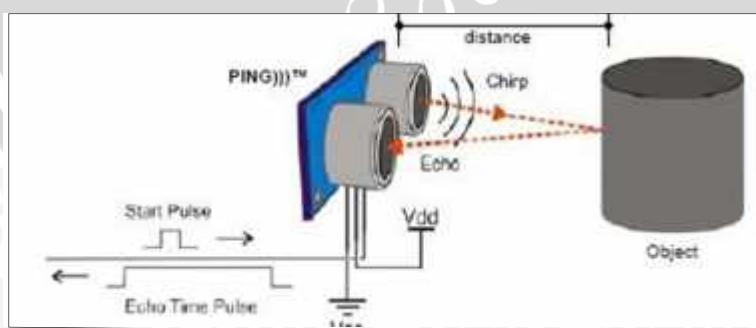
Sensor ultrasonik PING))) produksi Parallax yang berupa modul siap pakai lengkap dengan pengirim dan penerima. Sensor ultrasonik digunakan untuk mendeteksi jarak dengan frekuensi 40 kHz. Sinyal data sensor PING))) ini akan masuk ke kaki mikrokontroler. Bentuk fisik sensor ultrasonik PING))) ditunjukkan pada Gambar 2.4.



Gambar 2.4 Sensor Ultrasonik PING)))

Sumber: Parallax, 2013

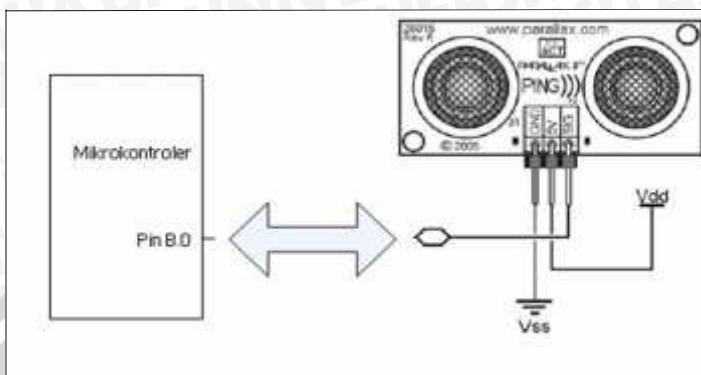
Modul PING))) mengukur jarak objek dengan cara memancarkan gelombang ultrasonik (40kHz) selama  $t_{BURST}$  (200  $\mu$ s) kemudian menunggu pantulannya. Modul PING))) memancarkan gelombang ultrasonik sesuai dengan masukan kontrol pin SIG. Gelombang ultrasonik ini melalui udara dengan kecepatan kurang lebih 344 meter per detik, mengenai objek dan memantul kembali ke modul PING))). Modul PING))) akan mengeluarkan pulsa *high* pada pin SIG setelah memancarkan gelombang ultrasonik. Setelah pantulan gelombang terdeteksi, modul PING))) akan membuat pin SIG *low*. Lebar pulsa *High* ( $t_{IN}$ ) ini sesuai dengan lama waktu tempuh gelombang ultrasonik untuk 2 kali jarak objek, sehingga jarak objek yang terukur adalah [ ( $t_{IN}$  s X 344m/s) / 2 ] meter. Ilustrasi cara kerja modul PING))) ditunjukkan pada Gambar 2.5.



Gambar 2.5 Ilustrasi Cara Kerja Modul PING)))

Sumber: Parallax, 2008

Sensor PING))) ini memiliki 3 buah kaki atau pin yakni pin VCC, pin GND dan pin SIG. Pin yang digunakan untuk mengirim data adalah pin SIG. Komunikasi mikrokontroler dengan PING ditunjukkan pada Gambar 2.6.



Gambar 2.6 Komunikasi Mikrokontroler dengan PING)))

Sumber: Parallax, 2008

Sensor PING))) ini memiliki beberapa kelebihan diantaranya bahwa sensor ini memiliki tingkat akurasi yang tinggi, sensor ini mampu mendeteksi benda di depannya walaupun sudutnya tidak tegak lurus, dan sensor ini mampu mendeteksi benda walaupun benda tersebut berukuran kecil.

## 2.5 Motor DC *Brushed*

Prinsip kerja motor DC *brushed* sesuai dengan hukum kemagnetan Lorenz, yaitu membangkitkan fluksi magnet pada suatu konduktor berarus dalam medan magnet sehingga timbul gugal induksi.

Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet. Arah medan magnet dapat ditentukan dengan kaidah tangan kiri.

Kaidah tangan kiri untuk motor menunjukkan arah arus yang mengalir didalam sebuah konduktor yang berada dalam medan magnet. Jari tengah menunjukkan arah arus yang mengalir pada konduktor, jari telunjuk menunjukkan arah medan magnet dan ibu jari menunjukkan arah gaya putar (Kilian, 2002:296). Adapun besarnya gaya yang bekerja pada konduktor tersebut dapat dirumuskan dalam Persamaan:

$$F = B \cdot I \cdot L \sin \theta$$

Dimana :

$F$  = Gaya Pada Konduktor (Newton)

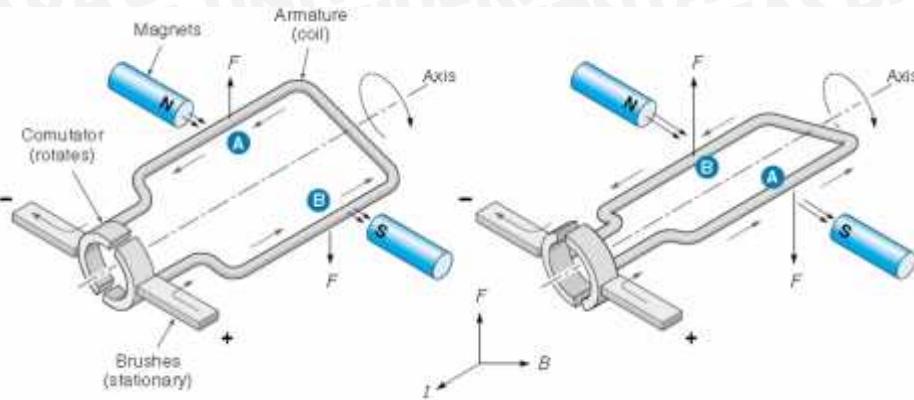
$B$  = kerapatan fluks magnet (gauss)

$L$  = panjang konduktor (meter)



$I$  = arus listrik (ampere)

$\nabla$  = Sudut antara arus dan medan magnet



Gambar 2.7 Ilustrasi Motor DC *Brushed*

Sumber : Kilian (2002:297)

Ilustrasi cara kerja motor DC yang mempunyai satu lilit kawat a–b berada di dalam medan magnet ditunjukkan dengan Gambar 2.7. Lilitan ini dapat berputar dengan bebas, lilitan ini biasa disebut dengan jangkar (*armature*).

Pada jangkar diberikan arus yang berasal dari sumber yang terhubung dengan sikat (*brushes*). Sikat-sikat ini terpasang pada sebuah cincin yang terbelah dua, yang disebut cincin belah (*comutator*). Adapun tujuan dari konstruksi ini adalah agar lilitan kawat dapat berputar apabila ada arus listrik yang melewatkinya.

Pada kawat yang berada di kanan arus mengalir dari depan ke belakang. Pada kawat yang berada di bagian kiri, arus mengalir dari belakang ke depan kawat a dan b secara bergantian berada di kiri dan kanan. Karena itu arah arus di a dan arah arus di b selalu bersifat bolak-balik. Pembalikan arah arus itu terjadi pada saat lilitan kawat melintasi posisi vertikal.

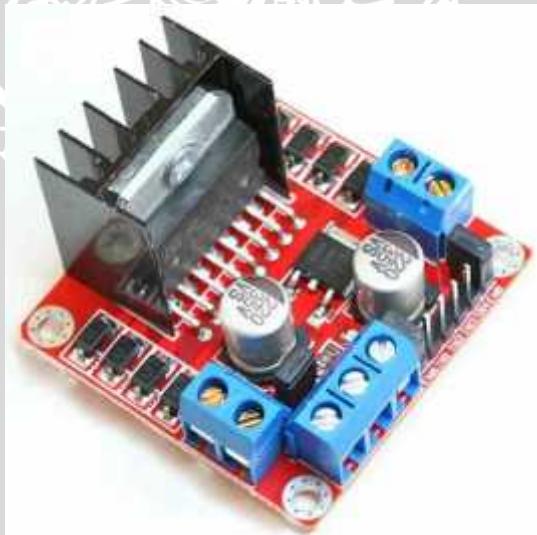
Bagian *comutator* berfungsi sebagai penyearah mekanik. Fluksi magnet yang ditimbulkan magnet permanen disebut medan magnet motor. Adapun gaya yang bekerja pada penghantar b adalah ke atas, sementara gaya yang bekerja pada penghantar a adalah ke bawah. Gaya-gaya yang bekerja sama kuatnya, sehingga terdapat kopel yang bekerja pada kawat sehingga lilitan jangkar dapat berputar. Setelah berputar  $180^\circ$  arah arus berbalik, pada saat itu penghantar a dan penghantar b bertukar tempat. Akibatnya arah gerak putaran tidak berubah.

## 2.6 L298 Dual H - Bridge Driver Motor

*L298 Dual H - Bridge Driver Motor* adalah pengendali dengan tegangan tinggi, arus tinggi dengan dual full-bridge driver yang dirancang untuk menerima tingkat TTL logika standar dan pengendali beban induktif seperti relay, solenoida, DC dan motor stepper. Bentuk *L298 Dual H - Bridge Driver Motor* ditunjukkan pada gambar 2.8.

Spesifikasi :

- ) *Driver Voltage* : 5VDC – 46VDC
- ) *Driver Peak Current* : 2A
- ) *Logic Voltage* : 5VDC
- ) *Logic Power Output* : 5VDC at 1A max
- ) *Logic Current* : 0-36 mA
- ) *Max Driver Power* : 25W

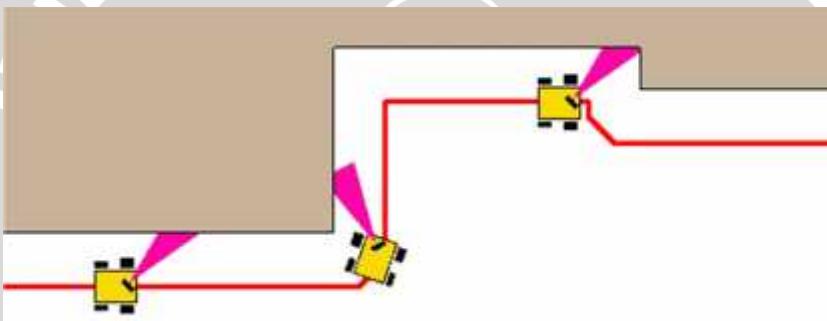


Gambar 2.8 L298N Dual H - Bridge Driver Motor

Sumber: [www.instructables.com](http://www.instructables.com)

## 2.7 Navigasi *Wall Following*

*Wall following* merupakan salah satu sistem navigasi yang digunakan untuk menyusuri kontur dinding. Metode ini biasanya digunakan robot yang memiliki kemampuan menyusuri dinding atau labirin untuk menyelesaikan misi misi tertentu. Pada dasarnya algoritma ini bertujuan untuk menjaga agar jarak robot pada dinding tetap pada batas yang diinginkan sementara robot terus bergerak maju. Bentuk algoritma, jumlah sensor, dan jenis sensor yang digunakan untuk metode navigasi *wall following* sangat beragam sesuai dengan perilaku robot yang diharapkan. Metode *wall following* dibagi menjadi dua yaitu *left wall following* dan *right wall following* seperti ditunjukkan dengan Gambar 2.9.



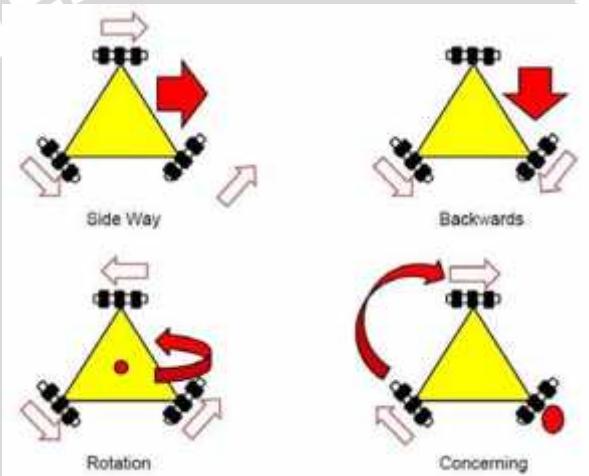
Gambar 2.9 Navigasi *Wall Following*

Sumber :<http://robot.cmpe.boun.edu.tr>

Kedua metode tersebut pada dasarnya sama yaitu menggunakan referensi robot terhadap dinding, bedanya hanya pada *Left Wall Following* menggunakan referensi dinding kiri dan *Right Wall Following* menggunakan referensi dinding kanan. Berdasarkan kondisi ruangan diambil jarak dinding terhadap robot untuk dijadikan referensi batas minimal dan batas maksimal jarak yang diperbolehkan antara robot dengan dinding kiri dan kanan.

## 2.8 *Omni Wheel Robot*

*Omni wheel robot* adalah merupakan sebuah rancangan *mobile robot* yang menggunakan roda *omni* (*omni wheel*) sehingga bisa bergerak ke segala arah, di sudut manapun, tanpa memutar badan robot terlebih dahulu. Beberapa *omni wheel robot* menggunakan *platform* segitiga dan juga persegi empat. Dimana dalam *platform* segitiga tersebut, *robot omni* menggunakan tiga buah roda *omni* yang terpisah sejauh  $120^\circ$  dan pada *platform* persegi empat, *robot omni* menggunakan empat buah roda *omni*. Bentuk *platform* segitiga menggunakan tiga buah *omni wheel* dengan perpindahan badan *robot* ditunjukkan pada gambar 2.10.



Gambar 2.10 Bentuk *platform* segitiga menggunakan tiga buah *omni wheel* dengan perpindahan badan *robot*

Sumber: [www.nexusrobot.com](http://www.nexusrobot.com)

## BAB III

### METODE PENELITIAN

Untuk menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian maka diperlukan langkah-langkah metode untuk menyelesaikan masalah tersebut. Adapun langkah-langkah yang perlu dilakukan untuk merealisasikan sistem yang dirancang adalah Spesifikasi alat, Perancangan Alat, dan Pengujian Alat.

#### 3.1 Spesifikasi

Spesifikasi alat secara *global* ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi alat yang direncanakan adalah sebagai berikut:

- 1) Mikrokontroler yang digunakan adalah *Arduino Mega 2560* yang berfungsi untuk mengolah data, sensor ultrasonik, dan driver motor.
- 2) *Omni Wheel* digunakan sebagai roda robot yang memiliki diameter 4,8 cm.
- 3) Menggunakan tiga buah motor DC sebagai aktuator.
- 4) *PING*) sebagai pendekripsi jarak robot terhadap dinding.
- 5) Mekanika Robot berbahan dasar mika *acrylic*.
- 6) Dimensi robot berukuran panjang 21 cm, lebar 23 cm, dan tinggi 17 cm.
- 7) Sistem navigasi *wall following* yang diterapkan pada robot.

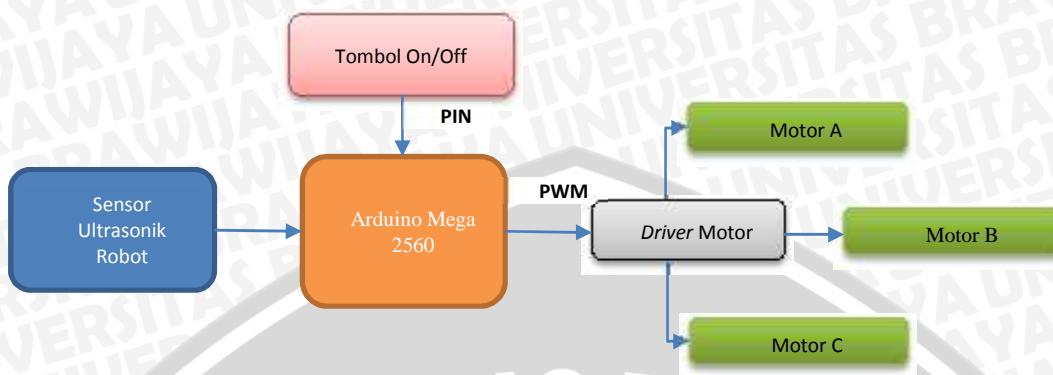
#### 3.2 Perancangan Alat (Desain Alat)

Perancangan dan pembuatan alat dalam penelitian ini dibagi menjadi dua bagian, yaitu *hardware* dan *software*.

##### 3.2.1 Perancangan Perangkat Keras (*Hardware*)

Secara garis besar perancangan dan pembuatan perangkat keras dapat dibagi menjadi beberapa bagian yaitu Perancangan Diagram Blok Sistem, Perancangan Mekanika Robot dan Perancangan Rangkaian *Driver*.

### 3.2.1.1 Perancangan Blok Diagram Sistem



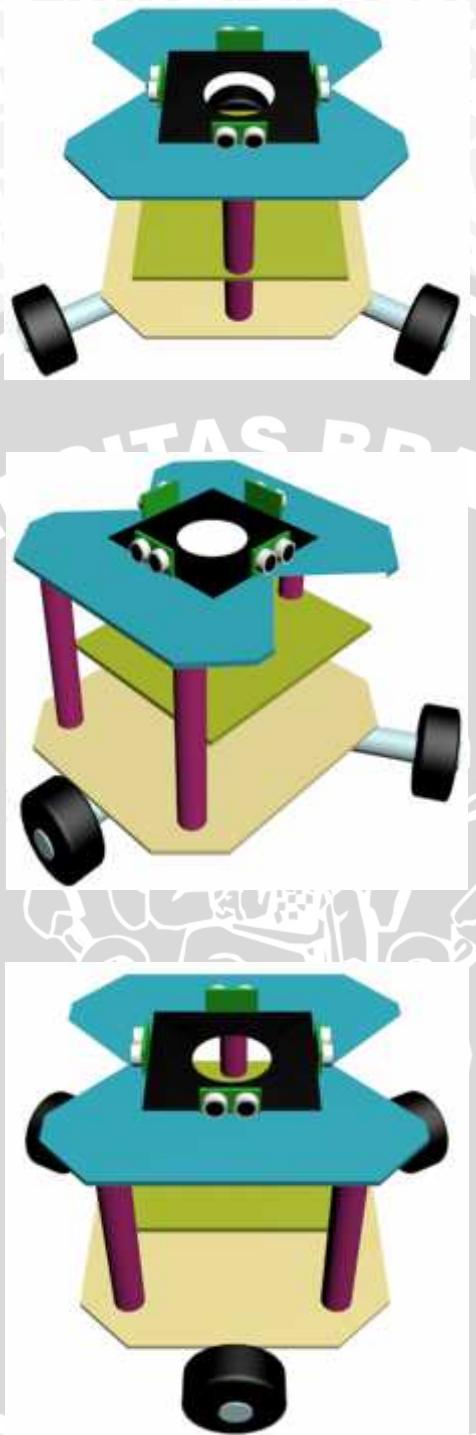
Gambar 3.1 Blok Diagram Sistem

Fungsi masing-masing bagian dalam diagram blok ini adalah sebagai berikut:

- 1) Arduino Mega 2560 berfungsi sebagai pengontrol utama dari sensor ultrasonik, tombol, dan motor DC.
- 2) Tombol *on/off* berfungsi untuk menghidupkan dan mematikan *mobile robot*.
- 3) Sensor Ultrasonik berfungsi untuk mendeteksi jarak robot terhadap objek dinding.
- 4) Modul pengendali/*driver* motor DC digunakan sebagai antarmuka mikrokontroller dengan aktuator.

Prinsip kerja sistem ini adalah awalnya robot akan mengukur jarak terhadap objek dinding yang ada di depannya menggunakan sensor ultrasonik. Apabila robot mendeteksi bahwa nilai jaraknya lebih besar daripada nilai batas yang ditentukan maka robot akan bergerak mendekati objek dinding sedangkan apabila robot mendeteksi bahwa nilai jaraknya lebih kecil daripada nilai batas yang ditentukan maka robot akan bergerak menjauhi objek dinding, dan apabila robot mendeteksi bahwa nilai jaraknya sama dengan nilai batas yang ditentukan maka robot akan bergerak menelusuri dinding (*Wall Following*) dengan *omni wheel*.

### 3.2.1.2 Perancangan Mekanika Robot

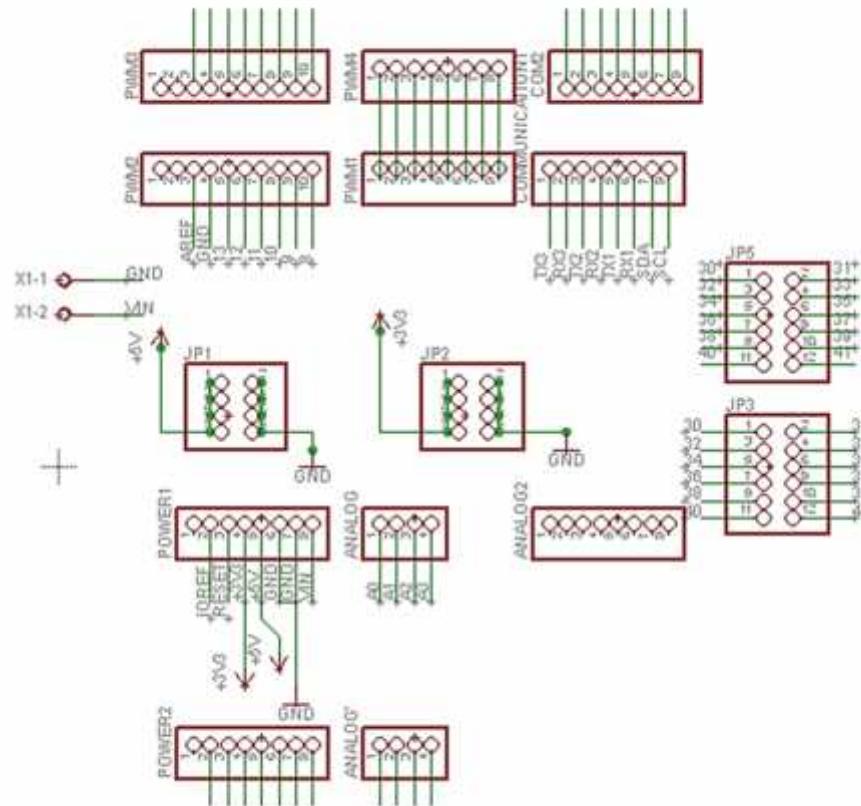


Gambar 3.2 Mekanika Robot

Dalam *plotform* mekanik segitiga tersebut, robot omni menggunakan tiga buah roda omni yang terpisah sejauh  $120^\circ$ .

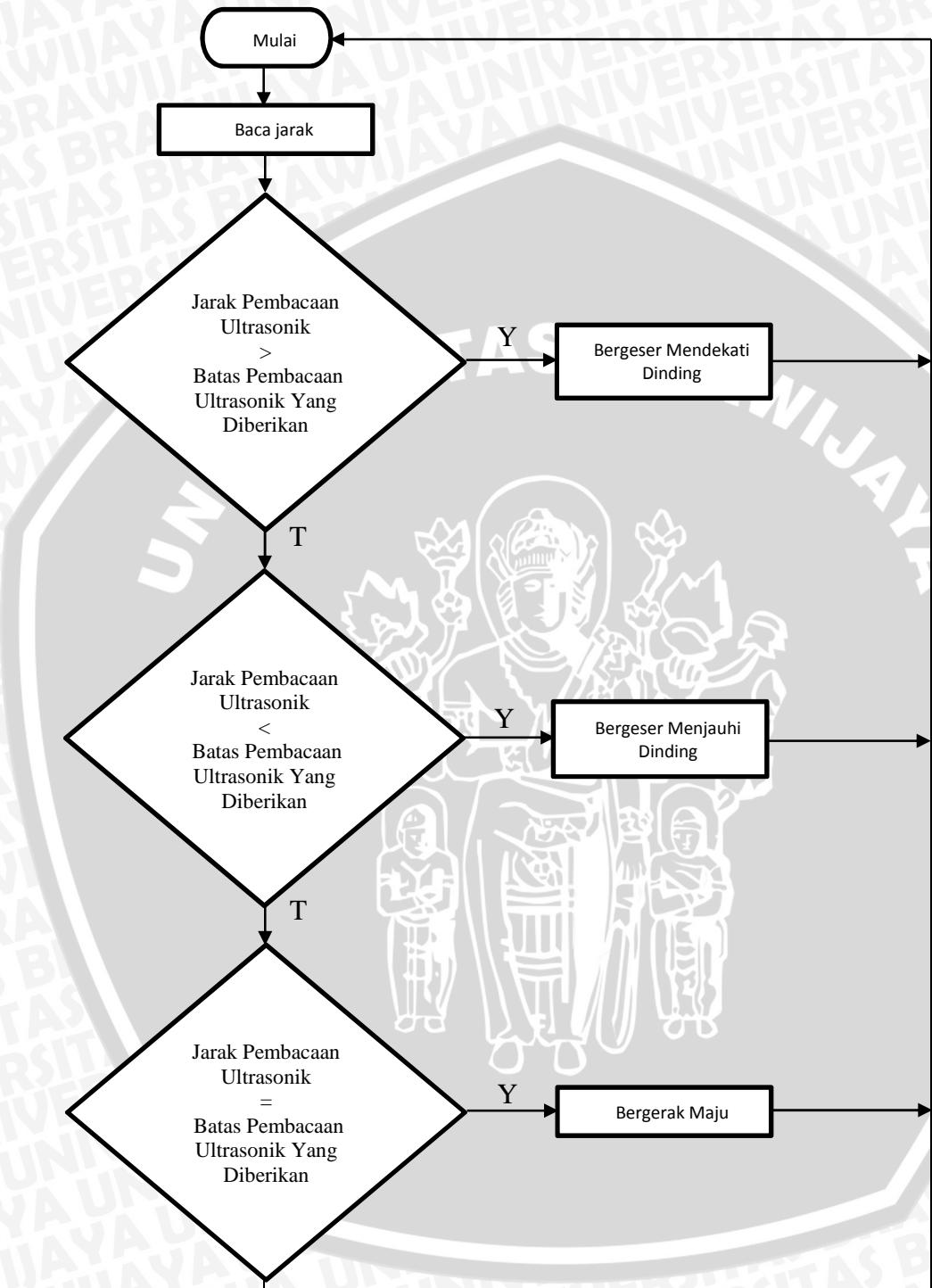
### 3.2.1.3 Perancangan *Shield* Arduino Mega 2560

*Shield* Arduino Mega 2560 adalah papan pcb yang dapat dihubungkan dengan board Arduino Mega 2560 yang digunakan untuk menambah fungsi pin dari Arduino Mega 2560. Fungsi-Fungsi yang perlu penambahan pin seperti pin Vin, 5v, 3v, dan *Ground*. Pada Gambar 3.3 Skematik *Shield* Arduino Mega 2560 yang dirancang.



### 3.2.2.1 Diagram Alir Sistem Utama Robot

Diagram Alir Robot ditunjukkan pada gambar 3.4.



Gambar 3.4 Diagram Alir Robot

Agar robot dapat menelusuri dinding dengan baik maka digunakan teknik *wall following*, yaitu robot akan berjalan dengan menggunakan dinding sebagai acuan. Robot akan mengukur jarak terhadap objek dinding yang ada di depannya menggunakan sensor ultrasonik. Apabila robot membaca bahwa nilai jarak sensor ultrasonik lebih besar daripada nilai batas yang ditentukan maka robot akan bergerak mendekati objek dinding sedangkan apabila robot membaca bahwa nilai jarak lebih kecil daripada nilai batas yang ditentukan maka robot akan bergerak menjauhi objek dinding, dan apabila robot memabaca bahwa nilai jarak sama dengan nilai batas yang ditentukan maka robot akan bergerak maju menelusuri dinding (*Wall Following*) dengan *omni wheel*.

### 3.3 Pengujian Alat

Untuk mengetahui kinerja alat apakah sesuai dengan yang direncanakan maka dilakukan pengujian alat. Pengujian dilakukan pada masing-masing bagian dan kemudian secara keseluruhan sistem. Secara garis besar pengujian yang dilakukan adalah sebagai berikut.

#### 1) Pengujian Sensor Ultrasonik.

Pengujian ini dilakukan dengan tujuan untuk mengetahui apakah pembacaan sensor sesuai dengan jarak sesungguhnya. Prosedur pengujian dilakukan dengan menghubungkan sensor ultrasonik, *Arduino Mega 2560* dan terminal pada computer sesuai dengan Gambar 3.5.



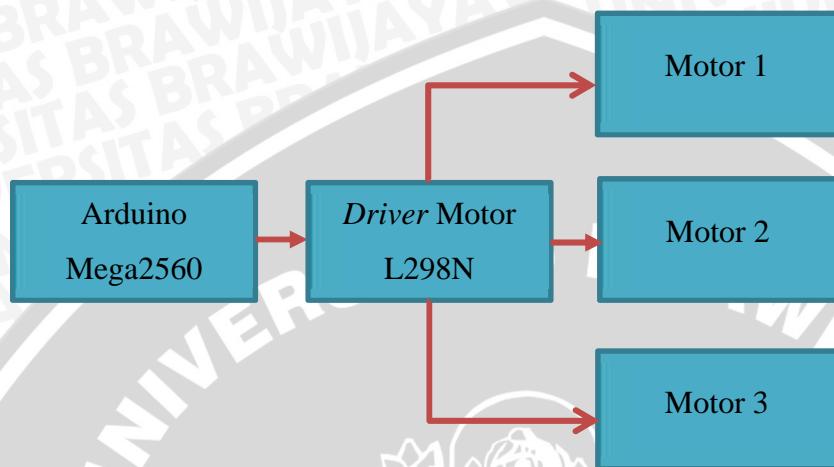
Gambar 3.5 Blok Diagram Pengujian Sensor Ultrasonik PING))

Pada pengujian ini, dinding lapangan diletakkan sejajar menurut letak-letak sensor ultrasonik sehingga sensor ultrasonik dapat mengukur jarak terhadap dinding.

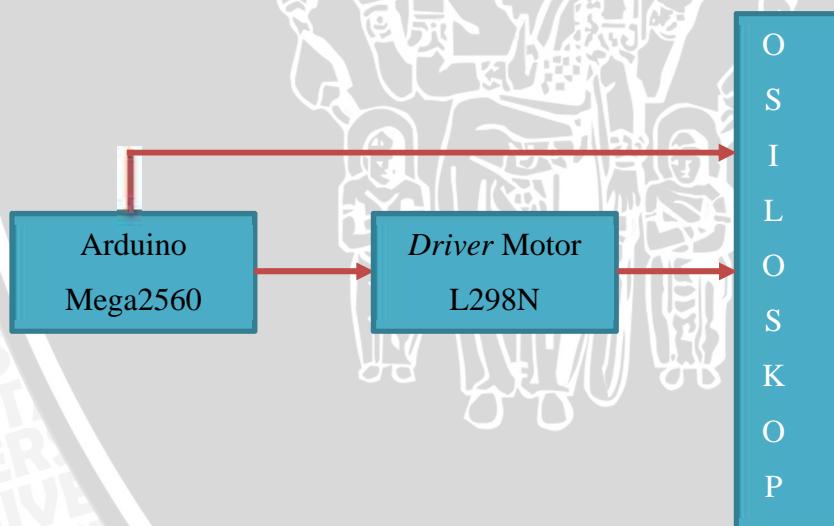
#### 2) Pengujian Driver

Pengujian ini bertujuan untuk mengetahui mikrokontroler pengendali motor DC sebagai antarmuka motor dengan mikrokontroler dapat bekerja dengan baik. Ada dua pengujian yang dilakukan yaitu pengujian pengendali arah putaran motor DC dan pengujian respon *Driver* motor

L298N terhadap masukan sinyal PWM. Pada Gambar 3.6 menunjukkan blok diagram pengujian pengendalian arah putaran motor DC dan Gambar 3.7 menunjukkan diagram blok pengujian respon *Driver* motor L298N terhadap masukan sinyal PWM.



Gambar 3.6 Blok Diagram Pengujian Pengendalian Arah Putaran Motor DC



Gambar 3.7 Blok Diagram Pengujian Respon *Driver* Motor L298N terhadap Masukan Sinyal PWM

3) Pengujian Sistem Secara Keseluruhan

Pengujian ini dilakukan dengan cara menggabungkan semua bagian yang dibuat dan melihat kinerja sistem. Pengujian ini bertujuan untuk mengetahui kinerja sistem yang dibuat dan memberikan analisis terhadap kinerja sistem.



## BAB IV

### HASIL DAN PEMBAHASAN

Setelah melakukan perancangan langkah selanjutnya adalah pembahasan hasil pengujian yang bertujuan untuk menganalisis alat yang telah dirancang agar sesuai dengan perancangan yang diharapkan. Pengujian dilakukan tiap-tiap blok dengan tujuan untuk mengamati apakah tiap blok sistem sudah sesuai dengan perancangan, kemudian diikuti dengan pengujian secara keseluruhan. Adapun pengujian yang perlu dilakukan yakni sebagai berikut:

1. Pengujian Sensor Ultrasonik
2. Pengujian Driver Motor L298N
3. Pengujian Sistem Secara Keseluruhan

#### 4.1 Pengujian Sensor Ultrasonik

Pengujian Sensor Ultrasonik dilakukan dengan tujuan mengetahui apakah pembacaan sensor ultrasonik PING))) sesuai dengan jarak sesungguhnya. Gambar 4.1 menunjukkan diagram blok pengujian sensor ultrasonik PING))).



Gambar 4.1 Blok Diagram Pengujian Sensor Ultrasonik PING))

Pengujian dilakukan dengan cara Robot diletakkan sejajar dengan objek dinding. Jarak objek dinding dengan sensor ultrasonik akan diubah-ubah setiap kelipatan 5 cm sebanyak 5 kali pengujian. Selanjutnya data berupa jarak terbaca sensor ultrasonik dalam satuan centimeter (cm) akan ditampilkan pada *Serial Monitor* sedangkan untuk mengukur jarak sebenarnya robot terhadap objek dinding akan menggunakan penggaris dengan satuan centimeter (cm). Hasil pengujian yang diperoleh ditunjukkan pada Tabel 4.1.

Tabel 4.1 Hasil Pengujian Sensor Ultrasonik PING)))

Pengujian	Jarak Sebenarnya	Jarak Terukur Ultrasonik	Eror
1	5,00 cm	5,74 cm	0,74 cm
2	10,00 cm	10,53 cm	0,53 cm
3	15,00 cm	15,65 cm	0,65 cm
4	20,00 cm	20,36 cm	0,36 cm
5	25,00 cm	25,56 cm	0,56 cm

Dari hasil pengujian diperoleh bahwa kesalahan rata-rata yang terjadi saat pembacaan sensor ultrasonik adalah 0,568 cm dan kesalahan pembacaan terbesar yaitu 0,74 cm. Pada pengujian, kesalahan pembacaan yang terjadi berupa hasil pengukuran yang lebih besar pada nilai desimal dibelakang koma, sedangkan pada nilai desimal di depan koma bernilai sama dengan jarak yang diuji. Kesalahan tersebut tidak memberikan pengaruh pada kinerja sistem yang dirancang karena pada sistem hanya digunakan data jarak dengan nilai desimal di depan koma.





(a)

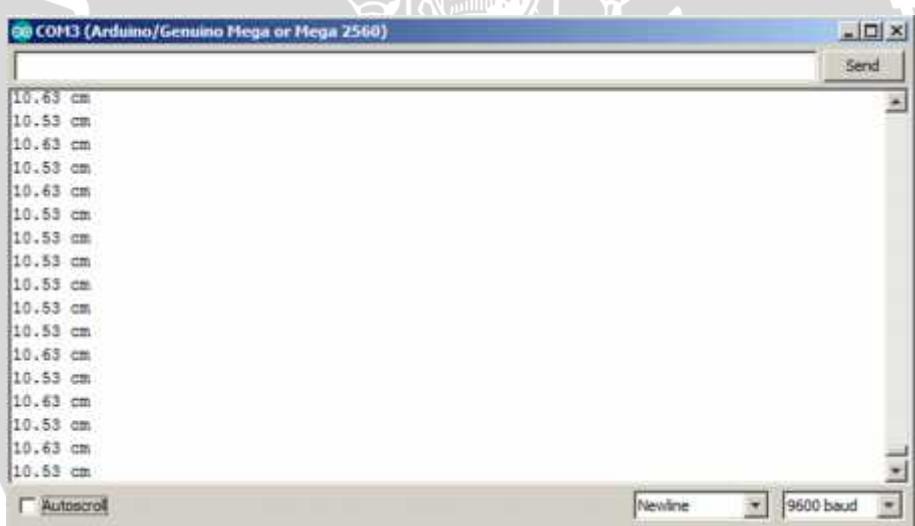
```
5.74 cm
5.74 cm
5.74 cm
5.74 cm
5.74 cm
5.64 cm
5.64 cm
5.74 cm
5.74 cm
5.74 cm
5.64 cm
5.64 cm
5.74 cm
5.74 cm
5.74 cm
5.74 cm
5.74 cm
5.74 cm
```

(b)

Gambar 4.2 (a) Tampilan Hasil Pengujian pada Jarak 5 cm, (b) Hasil Pengujian Menggunakan Serial Monitor



(a)

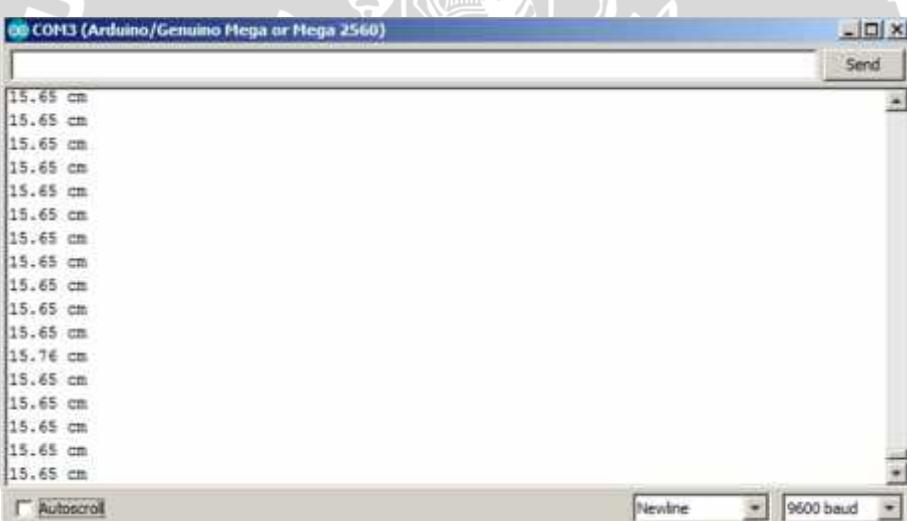


(b)

Gambar 4.3 (a) Tampilan Hasil Pengujian pada Jarak 10 cm Menggunakan Penggaris, (b) Hasil Pengujian Menggunakan *Serial Monitor*

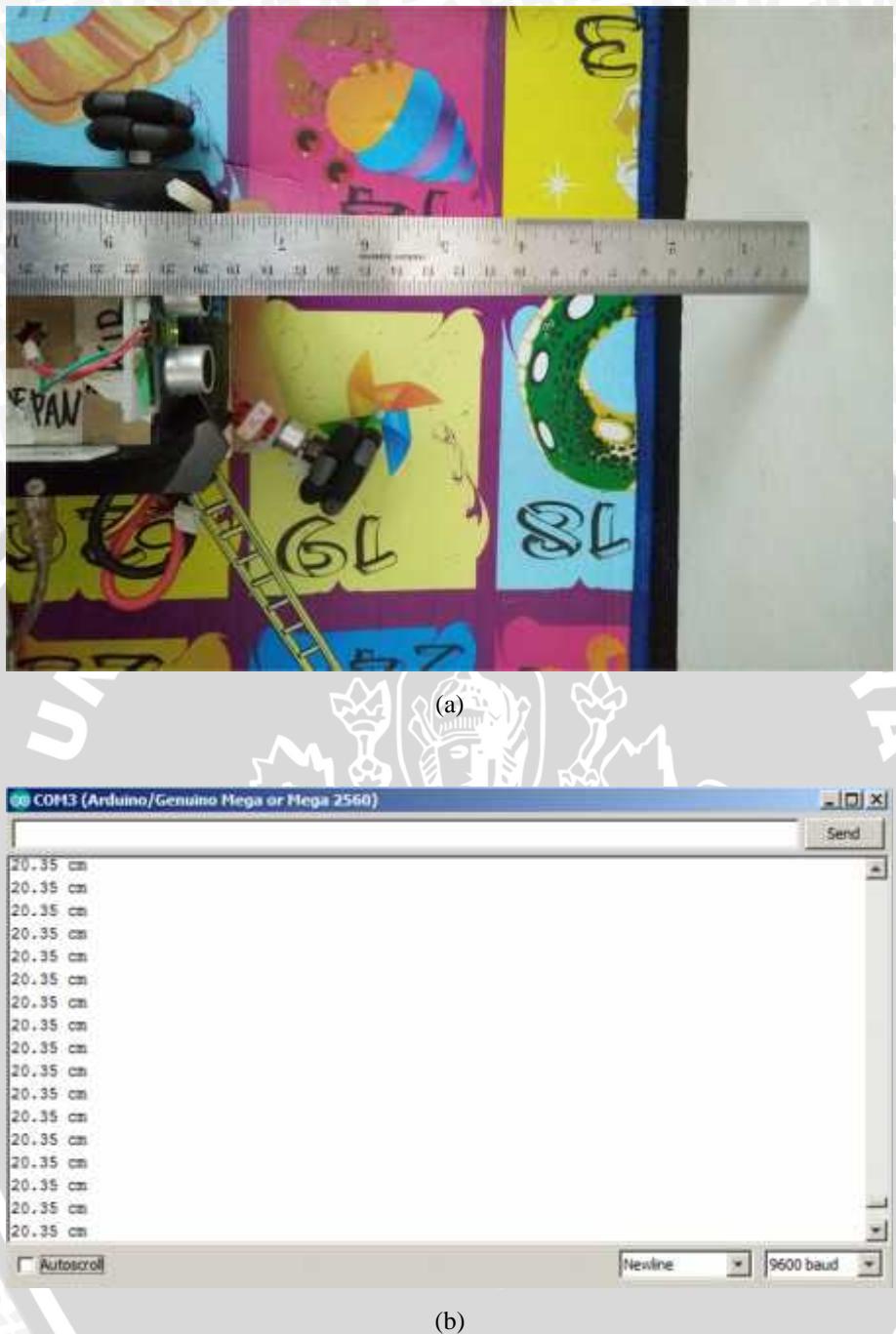


(a)

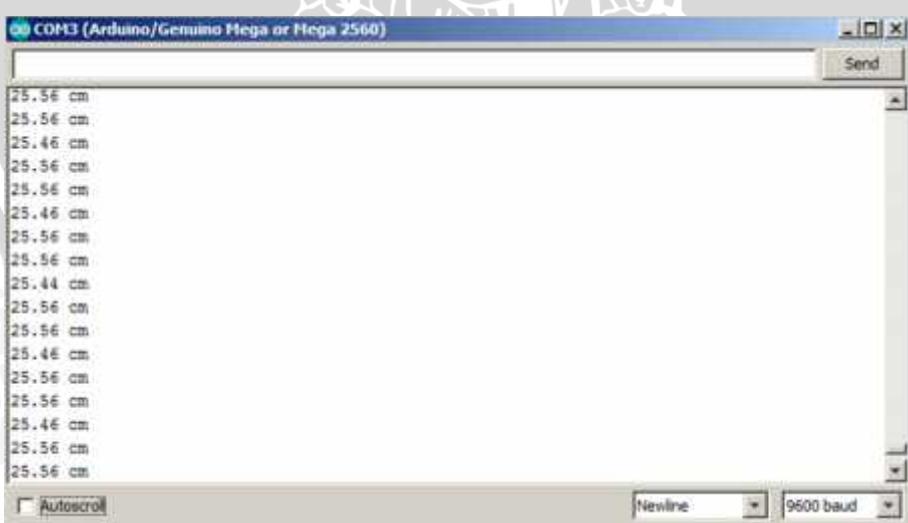


(b)

Gambar 4.4(a) Tampilan Hasil Pengujian pada Jarak 15 cm, (b) Hasil Pengujian Menggunakan Serial Monitor



Gambar 4.5 (a) Tampilan Hasil Pengujian pada Jarak 20 cm Menggunakan Penggaris, (b) Hasil Pengujian Menggunakan *Serial Monitor*



The screenshot shows the Arduino Serial Monitor window titled "COM3 (Arduino/Genuino Mega or Mega 2560)". The text area displays a series of identical measurements: "25.56 cm". At the bottom, there are controls for "Autoscroll", "Newline", and "9600 baud".

```
25.56 cm
25.56 cm
25.46 cm
25.56 cm
25.56 cm
25.46 cm
25.56 cm
25.44 cm
25.56 cm
25.56 cm
25.46 cm
25.56 cm
25.56 cm
25.56 cm
25.46 cm
25.56 cm
25.56 cm
25.56 cm
```

Gambar 4.6 (a) Tampilan Hasil Pengujian pada Jarak 25 cm Menggunakan Penggaris, (b) Hasil Pengujian Menggunakan *Serial Monitor*

## 4.2 Pengujian *Driver* Pengendali Motor DC

### 4.2.1 Pengujian Pengendalian Arah Putaran Motor DC

Pengujian pengendalian arah putar motor DC dilakukan dengan cara memberikan keluaran pin mikrokontroler berupa logika *high* dan *low* kemudian dihubungkan ke modul *driver L298N*. Pengujian Pengendalian arah putar motor DC dilakukan menurut jumlah motor DC yang digunakan yakni 3 buah. Pada Tabel 4.2 menunjukkan hasil pengujian pengendalian arah putar motor DC 1, Pada Tabel 4.3 menunjukkan hasil pengujian pengendalian arah putar motor DC 2, dan Pada Tabel 4.4 menunjukkan hasil pengujian pengendalian arah putar motor DC 3.

Tabel 4.2 Hasil Pengujian Pengendalian Arah Putar Motor DC 1

Pin 32	Pin 33	Respon Motor DC 1	
		Pergerakan Yang Diharapkan	Hasil Pengujian
1	0	Berputar Searah Jarum Jam	Berputar Searah Jarum Jam
0	1	Berputar Melawan Arah Jarum Jam	Berputar Melawan Arah Jarum Jam
0	0	Tidak Berputar	Tidak Berputar

Tabel 4.3 Hasil Pengujian Pengendalian Arah Putar Motor DC 2

Pin 34	Pin 35	Respon Motor DC 2	
		Pergerakan Yang Diharapkan	Hasil Pengujian
1	0	Berputar Searah Jarum Jam	Berputar Searah Jarum Jam
0	1	Berputar Melawan Arah Jarum Jam	Berputar Melawan Arah Jarum Jam
0	0	Tidak Berputar	Tidak Berputar

Tabel 4.4 Hasil Pengujian Pengendalian Arah Putar Motor DC 3

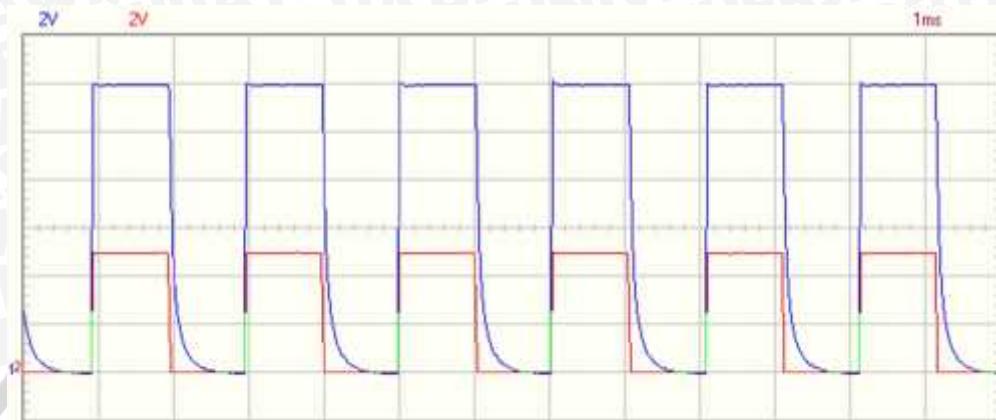
Pin 36	Pin 37	Respon Motor DC 3	
		Pergerakan Yang Diharapkan	Hasil Pengujian
1	0	Berputar Searah Jarum Jam	Berputar Searah Jarum Jam
0	1	Berputar Melawan Arah Jarum Jam	Berputar Melawan Arah Jarum Jam
0	0	Tidak Berputar	Tidak Berputar

Berdasarkan Tabel 4.2, Tabel 4.3, dan Tabel 4.4 dapat diketahui bahwa pengendalian arah putar motor dapat bekerja dengan baik.

#### 4.2.2 Pengujian *Duty Cycle Driver L298N Terhadap Sinyal Masukan PWM*

Pengujian ini dilakukan dengan tujuan mengetahui kinerja dan respon dari *driver* L298N dengan membandingkan sinyal *duty cycle* pada rangkaian *driver* terhadap sinyal keluaran *duty cycle* PWM dari mikrokontroler. Pengujian dilakukan dengan menghubungkan *driver* motor L298N, mikrokontroler utama dan osiloskop digital Velleman PCLAB PCSU1000. Dengan memberikan sinyal

masukan *duty cycle* PWM dari *range* 0% sampai 100% pada pin masukan *driver* dengan kenaikan 10% maka sinyal persegi keluaran dari mikrokontroler dan *driver* L298 dapat diamati pada osiloskop. Adapun hasil pengujian *duty cycle* PWM sebesar 50% ditunjukkan pada Gambar 4.7.



Gambar 4.7 Sinyal Keluaran *Driver* L298N terhadap Sinyal Masukan PWM MK Sebesar 50%  
(Warna Merah: Sinyal Masukan PWM MK, Warna Biru: Sinyal Keluaran *Driver* L298N)

Hasil pengujian keseluruhan data selisih *duty cycle* antara sinyal PWM mikrokontroler dengan keluaran *driver* motor L298N dapat dilihat pada Tabel 4.5

Tabel 4.5 Hasil Pengujian *Duty Cycle* *Driver* L298N terhadap Sinyal Masukan PWM

<b>Duty Cycle</b> <b>PWM</b> <b>MK</b>	<b>Duty Cycle</b> <b>Keluaran</b> <b><i>driver</i></b>	<b>Selisih</b> <b><i>Duty Cycle</i></b>
0%	0%	0%
10,0 %	19,3 %	9,3 %
20,0 %	29,1 %	9,1 %
30,0 %	40,1 %	10,1 %
40,0 %	50,1 %	10,1 %
50,0 %	57,3 %	7,3 %
60,0 %	66,0 %	6 %
70,0 %	75,2 %	5,2 %
80,0 %	87,3 %	7,3 %
90,0 %	94,6 %	4,6 %
100 %	100 %	0 %

Berdasarkan Tabel 4.5 dapat diketahui bahwa pada pengujian terdapat selisih *duty cycle* keluaran rata-rata sebesar 6,9%. Selisih rata-rata 6,9% cukup baik karena tidak memberikan pengaruh yang besar pada kinerja sistem yang dirancang sehingga dapat disimpulkan bahwa *driver* L298N dapat bekerja dengan baik saat mendapatkan sinyal PWM dari mikrokontroler.

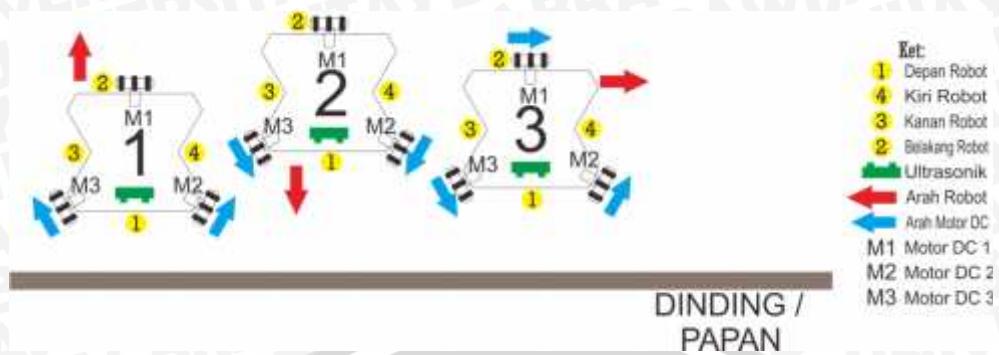
### 4.3 Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem ditinjau beberapa aspek yakni Osilasi, Standar Deviasi, dan Benturan terhadap dinding dengan tujuan untuk mengetahui kerja sistem secara keseluruhan apakah sesuai dengan perancangan atau tidak. Proses pengujian dilakukan menggunakan objek dinding dimana robot akan menggunakan navigasi *wall following* kanan dan *wall following* kiri secara bergantian dengan pemanfaatan arah badan robot secara bergantian yakni bagian depan, kanan, kiri, dan belakang sepanjangnya 7 meter.

Terdapat 8 bagian pengujian yang nantinya akan dicoba sebanyak 5 kali tiap bagian. Prosedurnya adalah robot akan menggunakan navigasi *wall following* kanan dan *wall following* kiri secara bergantian dengan arah badan robot bagian depan, kanan, kiri, belakang secara bergantian. Robot akan diletakkan diluar nilai jarak yang ditentukan sehingga robot akan mendekati objek dinding. Bila nilai jarak pembacaan sama dengan nilai jarak yang ditentukan maka robot akan bergerak lurus namun apabila nilai jarak pembacaan lebih kecil daripada nilai jarak yang ditentukan maka robot akan bergerak menjauhi objek dinding.

#### 4.3.1 Pengujian Robot Menggunakan Ultrasonik Depan Dengan Gerak Ke Kanan

Pengujian ini dilakukan untuk mengetahui apakah robot dapat menelusuri objek dinding menggunakan ultrasonik depan dengan gerak ke kanan. Pada Gambar 4.8 menunjukkan robot menelusuri objek dinding menggunakan ultrasonik depan dengan gerak ke kanan.



Gambar 4.8. Robot Menelusuri Objek Dinding Menggunakan Ultrasonik Depan dengan Gerak ke Kanan

Kondisi *follow* sensor ultrasonik depan dengan gerak ke kanan sebagai berikut.

1. Pada Robot 1, M1 tidak berputar, M2 dan M3 berputar ke belakang (kondisi : robot menjauhi dinding/papan karena jarak pembacaan sensor ultrasonik lebih kecil daripada nilai batas yang ditentukan).
2. Pada Robot 2, M1 tidak berputar, M2 dan M3 berputar ke depan (kondisi : robot mendekati dinding/papan karena pembacaan jarak sensor ultrasonik lebih besar daripada nilai batas jarak yang ditetapkan).
3. Pada Robot 3, M1 berputar ke kanan, M2 berputar ke depan, M3 berputar ke belakang (kondisi : robot berjalan lurus karena nilai jarak dan batas yang ditentukan sama besarnya).

Hasil pengujian Osilasi Robot dan Standar Deviasi ditunjukkan pada Tabel 4.6. Pada Tabel 4.7 menunjukkan hasil pengujian robot menggunakan ultrasonik depan dengan gerak ke kanan.



Tabel 4.6 Hasil Pengujian Osilasi dan Standar Deviasi Robot Menggunaan Ultrasonik Depan dengan Gerak ke Kanan

Pengujian ke- (cm)				
1	2	3	4	5
15	14	15	15	15
15	14,7	14	16	15
16	15	14	16	15,4
16,4	16	15	16	16
17	16	15	15	16,4
15	15	16,7	15	17
15	16	17	15	16,8
15	15	16,7	14	16
14,5	15	17,3	15	15
14,6	16	17	14	15,3
14	15	16	15	15
15	15	16	14	15
15,2	15	16	15	16,1
15,6	15	15	15	16
$\bar{x} = 15,23571$	$\bar{x} = 15,19286$	$\bar{x} = 15,76429$	$\bar{x} = 15$	$\bar{x} = 15,71429$
SD=0,784079	SD=0,59286	SD=1,105655	SD=0,679366	SD=0,708039

Tabel 4.7 Hasil Pengujian Robot Menggunakan Ultrasonik Depan dengan Gerak ke Kanan

Pengujian ke-	Jumlah Benturan	Jarak Rata-Rata Osilasi	Lama Waktu (detik)	Hasil Pengujian
1	-	15,23571	13,2	Berhasil
2	-	15,19286	12,9	Berhasil
3	-	15,76429	14	Berhasil
4	-	15	12,7	Berhasil
5	-	15,71429	14	Berhasil

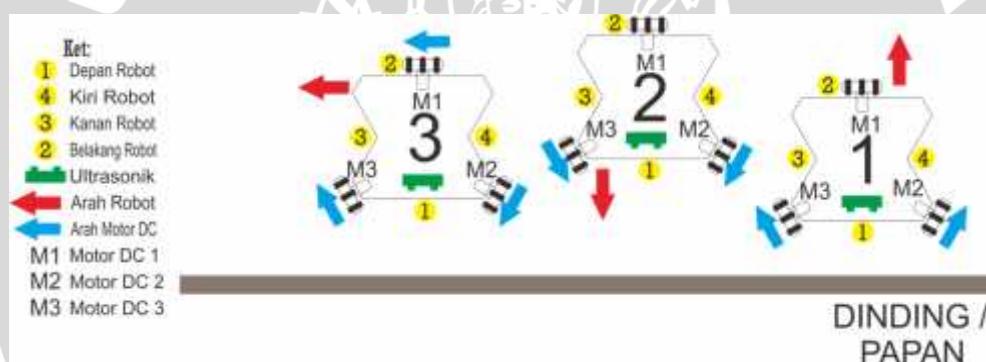
Berdasarkan hasil pengujian dari Tabel 4.6 Pengambilan data dilaksanakan setiap jarak 50 cm dari 0 cm sampai 700 cm sehingga dapat diketahui bahwa jarak rata-



rata keseluruhan robot berisolasি menggunakan ultrasonik depan dengan gerak ke kanan adalah sebesar 15,38143 cm dan Standar Deviasi rata-rata adalah 0,774. Pada Hasil Pengujian Tabel 4.7 dapat diketahui dari total 5 kali pengujian, robot dapat menyelesaikan tugasnya bernavigasi tanpa berbenturan dengan objek dinding. Untuk rata-rata waktu bernavigasi yakni 13,36 detik dengan jarak navigasi sepanjang 7 meter. Sehingga dapat dikatakan bahwa robot mampu bernavigasi dengan *wall following* menggunakan ultrasonik depan dengan gerak ke kanan.

#### 4.3.2 Pengujian Robot Menggunakan Ultrasonik Depan Dengan Gerak Ke Kiri

Pengujian ini dilakukan untuk mengetahui apakah robot dapat menelusuri objek dinding menggunakan ultrasonik depan dengan gerak ke kiri. Pada Gambar 4.9 menunjukkan robot menelusuri objek dinding menggunakan ultrasonik depan dengan gerak ke kiri.



Pada Gambar 4.9 Robot Menelusuri Objek Dinding Menggunakan Ultrasonik Depan dengan Gerak ke Kiri

Kondisi *follow* sensor ultrasonik depan dengan gerak ke kiri sebagai berikut.

1. Pada Robot 1, M1 tidak berputar, M2 dan M3 berputar ke belakang (kondisi : robot menjauhi dinding/papan karena jarak pembacaan sensor ultrasonik lebih kecil daripada nilai batas yang ditentukan).
2. Pada Robot 2, M1 tidak berputar, M2 dan M3 berputar ke depan (kondisi : robot mendekati dinding/papan karena pembacaan jarak sensor ultrasonik lebih besar daripada nilai batas jarak yang ditetapkan).
3. Pada Robot 3, M1 berputar ke kanan, M2 berputar ke depan, M3 berputar ke belakang (kondisi : robot berjalan lurus karena nilai jarak dan batas yang ditentukan sama besarnya).

Hasil pengujian Osilasi Robot dan Standar Deviasi ditunjukkan pada Tabel 4.8. Pada Tabel 4.9 menunjukkan hasil pengujian robot menggunakan ultrasonik depan dengan gerak ke kiri.

Tabel 4.8 Hasil Pengujian Osilasi Robot dan Standar Deviasi Menggunakan Ultrasonik Depan dengan Gerak ke Kiri

Pengujian ke- (cm)				
1	2	3	4	5
15	16	16	15	15
15	16	15	15	15
16,2	15	15	15	16
16,3	16	17	17	16,3
15,7	16	17,2	17,2	16
16,2	16,2	17	16	16
16	16,5	17,2	16	15
15,6	16	16,5	17	16
16	15	15,4	17,3	16,1
16,5	15,2	16	17,2	16,1
17	16,7	16	16	16
16	17	17,3	15,5	15
17,3	16	17	15	16,5
16	15,3	15	15	15,5
$\bar{x}=16,05714$	$\bar{x}=15,92143$	$\bar{x}=16,25714$	$\bar{x}=16,01429$	$\bar{x}=15,75$
SD=0,639368	SD=0,609134	SD=0,888139	SD=0,953421	SD=0,53601

Tabel 4.9 Hasil Pengujian Robot Menggunakan Ultrasonik Depan dengan Gerak ke Kiri

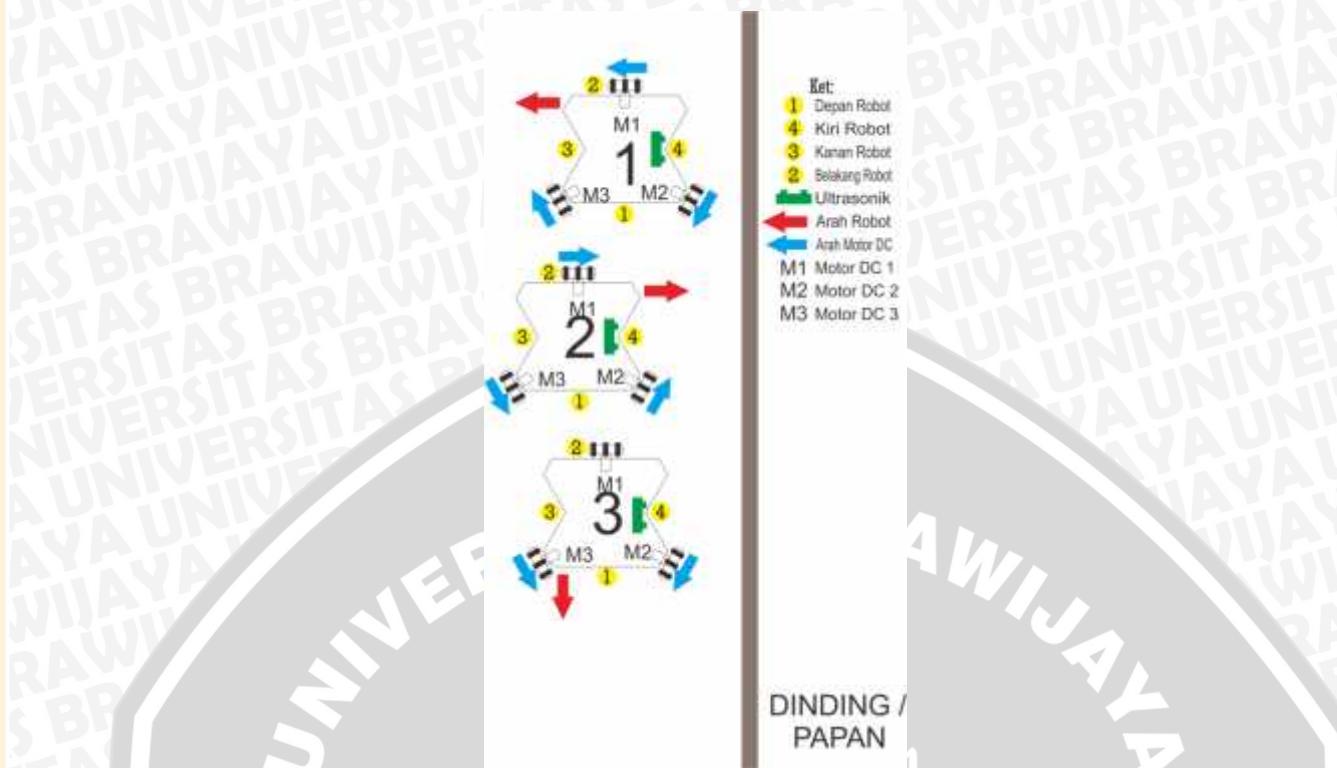
Pengujian ke-	Jumlah Benturan	Jarak Rata-Rata Osilasi	Lama Waktu (detik)	Hasil Pengujian
1	-	16,05714	14,6	Berhasil
2	-	15,92143	14,4	Berhasil
3	-	16,25714	15	Berhasil
4	-	16,01429	14,9	Berhasil
5	-	15,75	14	Berhasil

Berdasarkan hasil pengujian dari Tabel 4.8 Pengambilan data dilaksanakan setiap jarak 50 cm dari 0 cm sampai 700 cm sehingga dapat diketahui jarak rata-rata keseluruhan robot berisolasikan adalah sebesar 16 cm dan Standar Deviasi rata-rata sebesar 0,725214 cm. Pada Hasil pengujian Tabel 4.9 dapat diketahui dari total 5 kali pengujian, robot dapat menyelesaikan tugasnya bernavigasi tanpa berbenturan dengan objek dinding. Untuk rata-rata waktu bernavigasi yakni 14,58 detik dengan jarak navigasi sepanjang 7 meter. Sehingga dapat dikatakan bahwa robot mampu bernavigasi dengan *wall following* menggunakan ultrasonik depan dengan gerak ke kiri.

#### 4.3.3 Pengujian Robot Menggunakan Ultrasonik Kanan Dengan Gerak Maju

Pengujian ini dilakukan untuk mengetahui apakah robot dapat menelusuri objek dinding menggunakan ultrasonik kanan dengan *wall following* kanan. Pada Gambar 4.10 menunjukkan robot menelusuri objek dinding menggunakan ultrasonik kanan dengan gerak maju.





Pada Gambar 4.10 Robot Menelusuri Objek Dinding Menggunakan Ultrasonik Kanan dengan Gerak Maju

Kondisi *follow* sensor ultrasonik kanan dengan gerak maju sebagai berikut.

1. Pada Robot 1, M1 berputar ke kiri, M2 berputar ke belakang, M3 berputar ke depan (kondisi : robot menjauhi dinding/papan karena jarak pembacaan sensor ultrasonik lebih kecil daripada nilai batas yang ditentukan).
2. Pada Robot 2, M1 berputar ke kanan, M2 berputar ke depan, M3 berputar ke belakang (kondisi : robot mendekati dinding/papan karena pembacaan jarak sensor ultrasomik lebih besar daripada nilai batas jarak yang ditetapkan).
3. Pada Robot 3, M1 tidak berputar, M2 dan M3 berputar ke depan. (kondisi : kondisi : robot berjalan lurus karena nilai jarak dan batas yang ditentukan sama besarnya).

Hasil pengujian Osilasi Robot dan Standar Deviasi ditunjukkan pada Tabel 4.10. Pada Tabel 4.11 menunjukkan hasil pengujian robot menggunakan ultrasonik kanan dengan gerak maju.

Tabel 4.10 Hasil Pengujian Osilasi Robot dan Standar Deviasi Robot Menggunakan Ultrasonik Kanan dengan Gerak Maju

Pengujian ke- (cm)				
1	2	3	4	5
16	16	15	15	15,4
16	15	15	15	16,1
15	15	15,4	16	16
15,4	15	16	16	16
15,7	15,6	17	16	16
16	16	17	15	15,6
14	17	16,7	15	14
15	17	16	14,6	14,7
14	17	16	14	15
15	15	14	14	16
15,4	15	14,7	15,2	16
16	15,2	14,7	15,3	16
15	15	15	15	15,5
15	16	16	15	15
$\bar{x}=15,25$	$\bar{x}=15,7$	$\bar{x}=15,60714$	$\bar{x}=15,07857$	$\bar{x}=15,52143$
SD=0,675676	SD=0,810508	SD=0,922759	SD=0,632673	SD=0,63631

Tabel 4.11 Hasil Pengujian Robot Menggunakan Ultrasonik Kanan dengan Gerak Maju

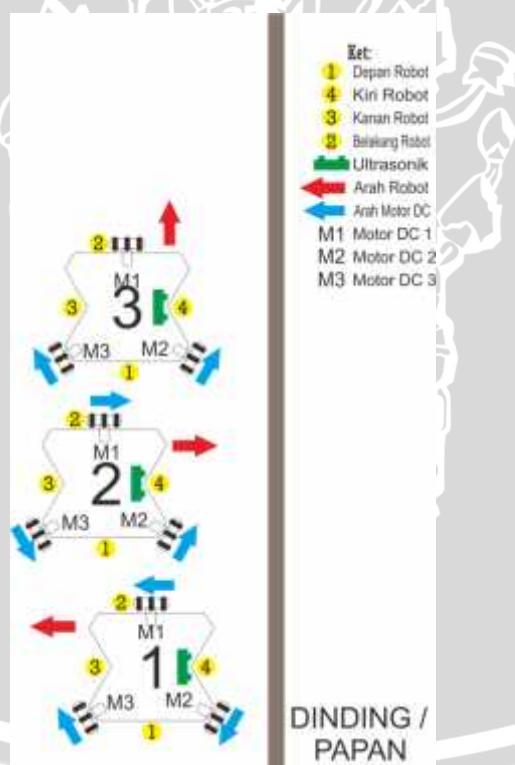
Pengujian ke-	Jumlah Benturan	Jarak Rata-Rata Osilasi	Lama Waktu (detik)	Hasil Pengujian
1	-	15,25	13,2	Berhasil
2	-	15,7	13,9	Berhasil
3	-	15,60714	13,7	Berhasil
4	-	15,07857	12,7	Berhasil
5	-	15,52143	13,7	Berhasil

Berdasarkan hasil pengujian dari Tabel 4.10 Pengambilan data dilaksanakan setiap jarak 50 cm dari 0 cm sampai 700 cm sehingga dapat diketahui bahwa jarak

rata-rata keseluruhan robot berisolasikan adalah sebesar 15,43143 dan Standar Deviasi rata-rata sebesar 0,735585 cm. Pada Hasil pengujian Tabel 4.11 dapat diketahui dari total 5 kali pengujian, robot dapat menyelesaikan tugasnya bernavigasi tanpa berbenturan dengan objek dinding. Untuk rata-rata waktu bernavigasi yakni 13,4 detik dengan jarak navigasi sepanjang 7 meter. Sehingga dapat dikatakan bahwa robot mampu bernavigasi dengan *wall following* menggunakan ultrasonik kanan dengan gerak maju.

#### 4.3.4 Pengujian Robot Menggunakan Ultrasonik Kanan Dengan Gerak Mundur

Pengujian ini dilakukan untuk mengetahui apakah robot dapat menelusuri objek dinding menggunakan ultrasonik kanan dengan gerak mundur. Pada Gambar 4.11 menunjukkan robot menelusuri objek dinding menggunakan ultrasonik kanan dengan gerak mundur.



Pada Gambar 4.11 Robot Menelusuri Objek Dinding Menggunakan Ultrasonik Kanan dengan Gerak Mundur

Kondisi *follow* sensor ultrasonik kanan dengan gerak mundur sebagai berikut.

- Pada Robot 1, M1 berputar ke kiri, M2 berputar ke belakang, M3 berputar ke depan (kondisi : robot menjauhi dinding/papan karena jarak pembacaan sensor ultrasonik lebih kecil daripada nilai batas yang ditentukan).
- Pada Robot 2, M1 berputar ke kanan, M2 berputar ke depan, M3 berputar ke belakang (kondisi : robot mendekati dinding/papan karena pembacaan jarak sensor ultrasonik lebih besar daripada nilai batas jarak yang ditetapkan).
- Pada Robot 3, M1 tidak berputar, M2 dan M3 berputar ke belakang (kondisi : kondisi : robot berjalan lurus karena nilai jarak dan batas yang ditentukan sama besarnya )

Hasil pengujian Osilasi Robot dan Standar Deviasi ditunjukkan pada Tabel 4.12. Pada Tabel 4.13 menunjukkan hasil pengujian robot menggunakan ultrasonik kanan dengan gerak mundur.

Tabel 4.12 Hasil Pengujian Osilasi Robot dan Standar Deviasi Robot Menggunakan Ultrasonik Kanan dengan Gerak Mundur

Pengujian ke- (cm)				
1	2	3	4	5
15	15,3	16	15	15
15	16	15,5	15,1	14
14,2	16	15	14	14
15	15,4	17	14	16
16	17	16,6	14	14
15,6	17	16	16	14
15	16	15	16	16
14,5	15	17	15	17
16,4	15	14	15	15
17	14	15	17	15
15	14	15	17	16
16	15	16	16	16
15	17	17	14	15
15	17	15	14	17
$\bar{x}=15,33571$	$\bar{x}=15,6926$	$\bar{x}=15,72143$	$\bar{x}=15,15$	$\bar{x}=15,28571$
SD=0,766217	SD=1,057184	SD=0,941503	SD=1,098776	SD=1,069045



Tabel 4.13 Hasil Pengujian Robot Menggunakan Ultrasonik Kanan dengan Gerak Mundur

Pengujian ke-	Jumlah Benturan	Jarak Rata-Rata Osilasi	Lama Waktu (detik)	Hasil Pengujian
1	-	15,33571	13,2	Berhasil
2	-	15,69286	13,9	Berhasil
3	-	15,72143	14	Berhasil
4	-	15,15	12,9	Berhasil
5	-	15,28571	13,1	Berhasil

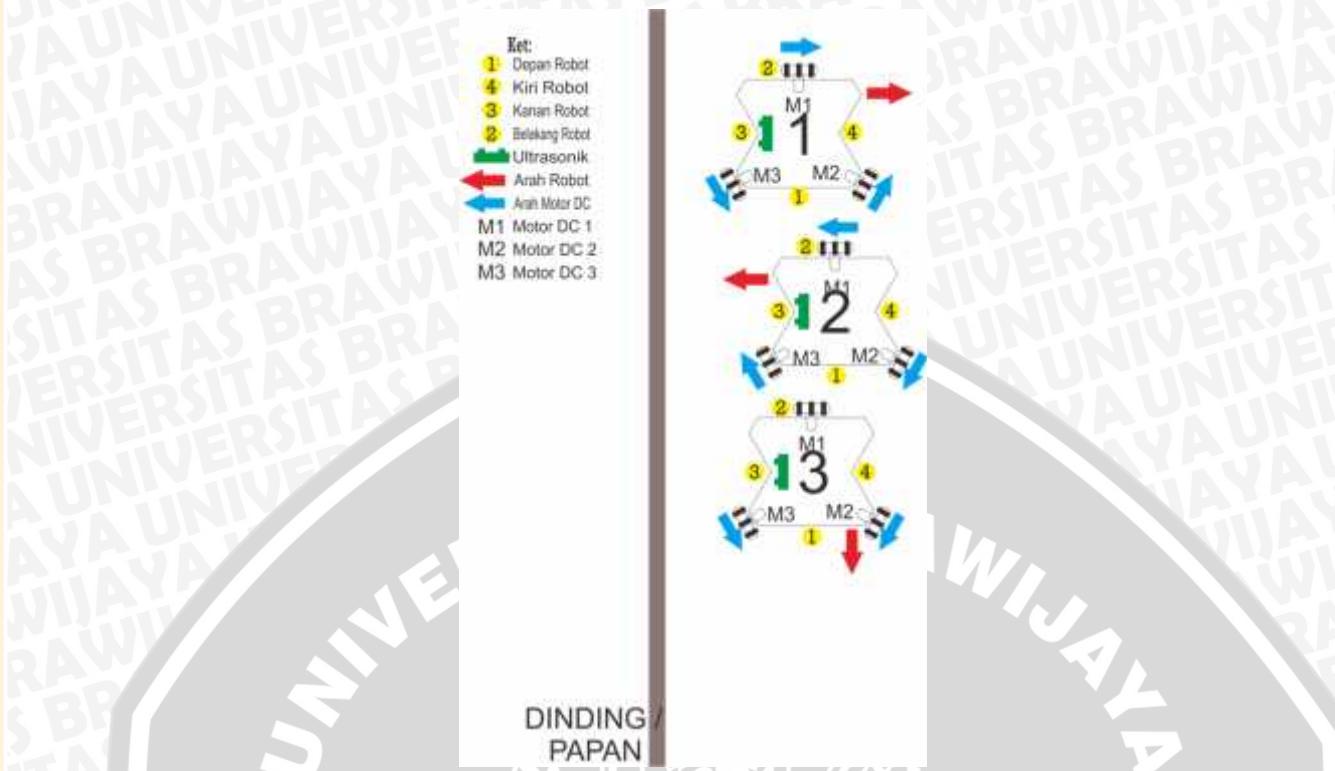
Berdasarkan hasil pengujian Tabel 4.12 Pengambilan data dilaksanakan setiap jarak 50 cm dari 0 cm sampai 700 cm sehingga dapat diketahui bahwa jarak rata-rata keseluruhan robot berisolasikan adalah sebesar 15,43714 dan Standar Deviasi rata-rata sebesar 0,986545 cm. Pada hasil pengujian Tabel 4.13 dapat diketahui dari total 5 kali pengujian, robot dapat menyelesaikan tugasnya bernavigasi tanpa berbenturan dengan objek dinding. Untuk rata-rata waktu bernavigasi yakni 13,42 detik dengan jarak navigasi sepanjang 7 meter. Sehingga dapat dikatakan bahwa robot mampu bernavigasi dengan *wall following* menggunakan ultrasonik kanan dengan gerak mundur.

#### 4.3.5 Pengujian Robot Menggunakan Ultrasonik Kiri Dengan Gerak Maju

Pengujian ini dilakukan untuk mengetahui apakah robot dapat menelusuri objek dinding menggunakan ultrasonik kiri dengan gerak maju.

Pada Gambar 4.12 menunjukkan robot menelusuri objek dinding menggunakan ultrasonik kiri dengan gerak maju.





Gambar 4.12 Robot Menelusuri Objek Dinding Menggunakan Ultrasonik Kiri dengan Gerak Maju.

Kondisi *follow* sensor ultrasonik kiri dengan gerak maju sebagai berikut.

1. Pada Robot 1, M1 berputar ke kanan, M2 berputar ke depan, M3 berputar ke belakang (kondisi : robot menjauhi dinding/papan karena jarak pembacaan sensor ultrasonik lebih kecil daripada nilai batas yang ditentukan).
2. Pada Robot 2, M1 berputar ke kiri, M2 berputar ke belakang, M3 berputar ke depan (kondisi : robot mendekati dinding/papan karena pembacaan jarak sensor ultrasonik lebih besar daripada nilai batas jarak yang ditetapkan).
3. Pada Robot 3, M1 tidak berputar, M2 dan M3 berputar ke depan. (kondisi : kondisi : robot berjalan lurus karena nilai jarak dan batas yang ditentukan sama besarnya )

Hasil pengujian Osilasi Robot dan Standar Deviasi ditunjukkan pada Tabel 4.14. Pada Tabel 4.15 menunjukkan hasil pengujian robot menggunakan ultrasonik kiri dengan gerak maju.

Tabel 4.14 Hasil Pengujian Osilasi Robot dan Standar Deviasi Robot Menggunakan Ultrasonik

Kiri dengan Gerak Maju

Pengujian ke- (cm)				
1	2	3	4	5
15	15	17	15	15,4
15,2	15	17,1	15	15
15,2	14,7	16,6	16	15
16	14	15	16,2	14
16	15	15	16	14
17	15	15	15	16
16	16	15	15,4	16
16	17	14	15	15
16	15	14	14	16,3
15,7	15,7	15	15	17
15	15	14	16	16
15	14,7	15	15	15
14	15	15,3	15	15
15,2	15	15	15	15
$\bar{x} = 15,52143$	$\bar{x} = 15,15$	$\bar{x} = 15,21429$	$\bar{x} = 15,25714$	$\bar{x} = 15,33571$
SD=0,72343	SD=0,699175	SD=1,015917	SD=0,599634	SD=0,776539

Tabel 4.15 Hasil Pengujian Robot Menggunakan Ultrasonik Kiri dengan Gerak Maju

Pengujian ke-	Jumlah Benturan	Jarak Rata-Rata Osilasi	Lama Waktu (detik)	Hasil Pengujian
1	-	15,52143	13,7	Berhasil
2	-	15,15	12,9	Berhasil
3	-	15,21429	13,1	Berhasil
4	-	15,25714	13,1	Berhasil
5	-	15,33571	13,2	Berhasil

Berdasarkan hasil pengujian Tabel 4.14 Pengambilan data dilaksanakan setiap jarak 50 cm dari 0 cm sampai 700 cm sehingga dapat diketahui bahwa jarak rata-

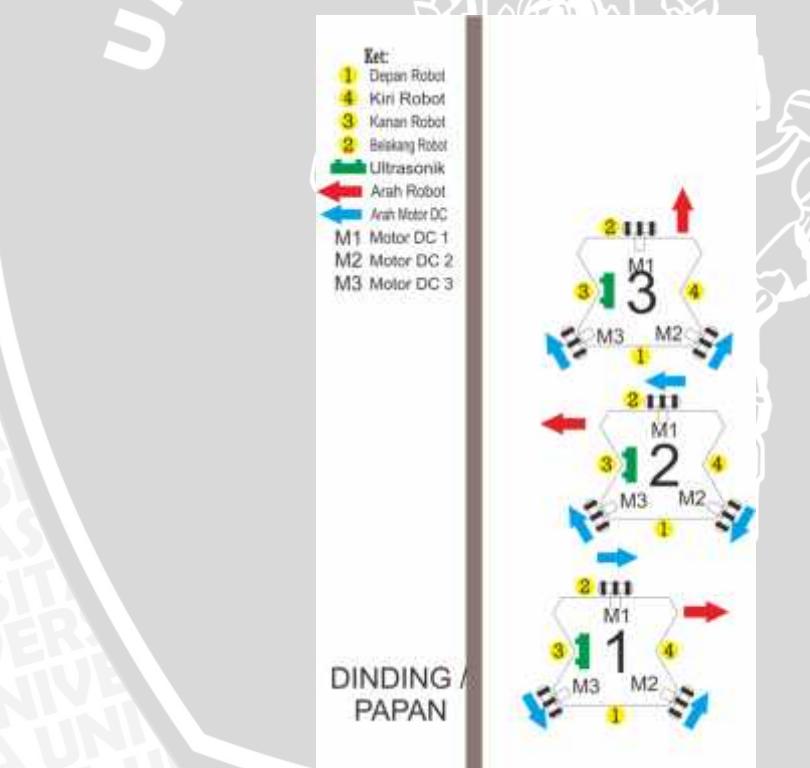


rata keseluruhan robot berisolasikan adalah sebesar 15,29571 cm dan Standar Deviasi rata-rata sebesar 0,776539 cm. Pada hasil pengujian Tabel 4.15 dari total 5 kali pengujian, robot dapat menyelesaikan tugasnya bernavigasi tanpa berbenturan dengan objek dinding. Untuk rata-rata waktu bernavigasi yakni 13,2 detik dengan jarak navigasi sepanjang 7 meter. Sehingga dapat dikatakan bahwa robot mampu bernavigasi dengan *wall following* menggunakan ultrasonik kiri dengan gerak maju.

#### 4.3.6 Pengujian Robot Menggunakan Ultrasonik Kiri Dengan Gerak Mundur

Pengujian ini dilakukan untuk mengetahui apakah robot dapat menelusuri objek dinding menggunakan ultrasonik kiri dengan gerak mundur.

Pada Gambar 4.13 menunjukkan robot menelusuri objek dinding menggunakan ultrasonik kiri dengan gerak mundur.



Gambar 4.13 Robot Menelusuri Objek Dinding Menggunakan Ultrasonik Kiri dengan Gerak Mundur.

Kondisi *follow* sensor ultrasonik kiri dengan gerak mundur sebagai berikut.

1. Pada Robot 1, M1 berputar ke kanan, M2 berputar ke depan, M3 berputar ke belakang (kondisi : robot menjauhi dinding/papan karena jarak

pembacaan sensor ultrasonik lebih kecil daripada nilai batas yang ditentukan).

2. Pada Robot 2, M1 berputar ke kiri, M2 berputar ke belakang, M3 berputar ke depan (kondisi : robot mendekati dinding/papan karena pembacaan jarak sensor ultrasonik lebih besar daripada nilai batas jarak yang ditetapkan).
3. Pada Robot 3, M1 tidak berputar, M2 berputar ke depan, M3 berputar ke depan (kondisi : robot berjalan lurus karena nilai jarak pembacaan sensor ultrasonik dan batas yang ditentukan sama besarnya)

Hasil pengujian Osilasi Robot dan Standar Deviasi ditunjukkan pada Tabel 4.16. Pada Tabel 4.17 menunjukkan hasil pengujian robot menggunakan ultrasonik kiri dengan gerak mundur.



Tabel 4.16 Hasil Pengujian Osilasi Robot dan Standar Deviasi Robot Menggunakan Ultrasonik

Kiri dengan Gerak Mundur

Pengujian ke- (cm)				
1	2	3	4	5
15,4	16	15	15	16
16	16	14	15	16
16,1	16	14	15	15
15	15	15	16	15
14	14	16,1	16	16
16	15	17	16	17
15	16	16	15	17
15	16,2	15	14	16
16	15,6	15	15	15
16	15	16	17	15
15	16,6	16	16	14
14	15	17	15	15
15	15	15	16	15
15	16	15	16	15
$\bar{x}=15,25$	$\bar{x}=15,52857$	$\bar{x}=15,43571$	$\bar{x}=15,5$	$\bar{x}=15,5$
SD=0,706835	SD=0,704304	SD=0,94267	SD=0,759555	SD=0,85485

Tabel 4.17 Hasil Pengujian Robot Menggunakan Ultrasonik Kiri dengan Gerak Mundur

Pengujian ke-	Jumlah Benturan	Jarak Rata-Rata Osilasi	Lama Waktu (detik)	Hasil Pengujian
1	-	15,25	13,2	Berhasil
2	-	15,52857	13,7	Berhasil
3	-	15,43571	13,6	Berhasil
4	-	15,5	13,7	Berhasil
5	-	15,5	13,7	Berhasil

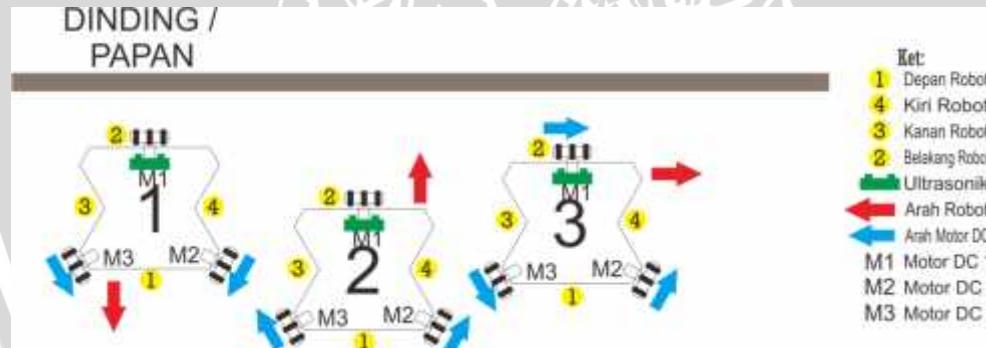
Berdasarkan hasil pengujian Tabel 4.16 Pengambilan data dilaksanakan setiap jarak 50 cm dari 0 cm sampai 700 cm sehingga dapat diketahui bahwa jarak rata-



rata robot berisolasikan adalah sebesar 15,44286 dan Standar Deviasi rata-rata sebesar 0,793643 cm. Pada hasil pengujian Tabel 4.17 dari total 5 kali pengujian, robot dapat menyelesaikan tugasnya bernavigasi tanpa berbenturan dengan objek dinding. Untuk rata-rata waktu bernavigasi yakni 13,58 detik dengan jarak navigasi sepanjang 7 meter. Sehingga dapat dikatakan bahwa robot mampu bernavigasi dengan *wall following* menggunakan ultrasonik kiri dengan gerak mundur.

#### 4.3.7 Pengujian Robot Menggunakan Ultrasonik Belakang Dengan Gerak Ke Kanan

Pengujian ini dilakukan untuk mengetahui apakah robot dapat menelusuri objek dinding menggunakan ultrasonik belakang dengan gerak ke kanan. Pada Gambar 4.14 menunjukkan robot menelusuri objek dinding menggunakan ultrasonik belakang dengan gerak ke kanan.



Gambar 4.14 Robot Menelusuri Objek Dinding Menggunakan Ultrasonik Belakang dengan Gerak ke Kanan

Kondisi *follow* sensor ultrasonik belakang dengan gerak ke kanan sebagai berikut.

1. Pada Robot 1, M1 tidak berputar, M2 dan M3 berputar ke belakang (kondisi : robot menjauhi dinding/papan karena jarak pembacaan sensor ultrasonik lebih kecil daripada nilai batas yang ditentukan).
2. Pada Robot 2, M1 tidak berputar, M2 dan M3 berputar ke depan (kondisi : robot mendekati dinding/papan karena pembacaan jarak sensor ultrasonik lebih besar daripada nilai batas jarak yang ditetapkan).

3. Pada Robot 3, M1 berputar ke kanan, M2 berputar ke depan, M3 berputar ke belakang (kondisi : robot berjalan lurus karena nilai jarak dan batas yang ditentukan sama besarnya).

Hasil pengujian Osilasi Robot dan Standar Deviasi ditunjukkan pada Tabel 4.18. Pada Tabel 4.19 menunjukkan hasil pengujian robot menggunakan ultrasonik belakang dengan gerak ke kanan.

Tabel 4.18 Hasil Pengujian Osilasi Robot dan Standar Deviasi Robot Menggunakan Ultrasonik Belakang dengan Gerak ke Kanan

Pengujian ke- (cm)				
1	2	3	4	5
16	15	16	16	17
16	15	16,2	17	16
15,5	16	16,4	16	16
16	16	15,6	16	16
15	17	15	15	15
15,2	17,6	16	15	15
15,2	18	15,6	15	15
16	17	17	16	15
17	16	17	16	14
16	15	16,7	16	15
15	16	15,5	15	15
14	15	14,6	15	16
15	16	15	16	15
15	15	15	16	15
$\bar{x}=15,4928$	$\bar{x}=16,0428$	$\bar{x}=15,8285$	$\bar{x}=15,7142$	$\bar{x}=15,3571$
SD=0,730084	SD=1,020343	SD=0,775022	SD=0,61125	SD=0,744946



Tabel 4.19 Hasil Pengujian Robot Menggunakan Ultrasonik Belakang dengan Gerak ke Kanan

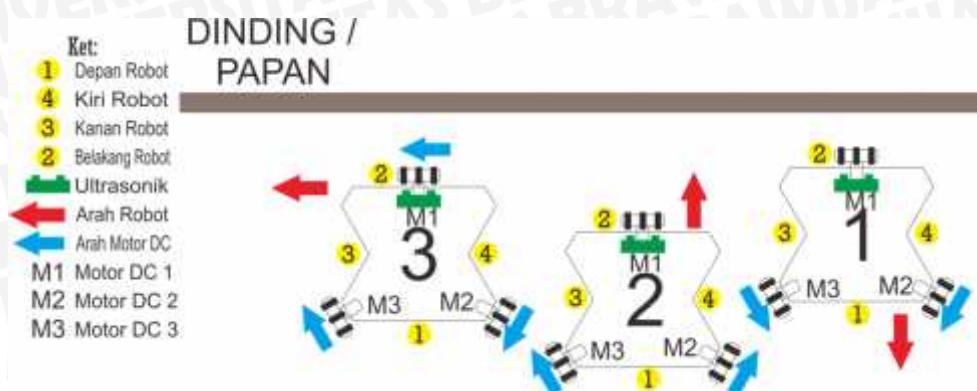
Pengujian ke-	Jumlah Benturan	Jarak Rata-Rata Osilasi	Lama Waktu (detik)	Hasil Pengujian
1	-	15,49286	13,5	Berhasil
2	-	16,04286	14,6	Berhasil
3	-	15,82857	14,2	Berhasil
4	-	15,71429	14,1	Berhasil
5	-	15,35714	13,3	Berhasil

Berdasarkan hasil pengujian dari Tabel 4.18 Pengambilan data dilaksanakan setiap jarak 50 cm dari 0 cm sampai 700 cm sehingga dapat diketahui bahwa jarak rata-rata keseluruhan robot berisolasikan adalah sebesar 15,68714 cm dan Standar Deviasi rata-rata sebesar 0,776329 cm. Pada hasil pengujian Tabel 4.19 diketahui bahwa dari total 5 kali pengujian, robot dapat menyelesaikan tugasnya bermavigasi tanpa berbenturan dengan objek dinding. Untuk rata-rata waktu bermavigasi yakni 13,94 detik dengan jarak navigasi sepanjang 7 meter. Sehingga dapat dikatakan bahwa robot mampu bermavigasi dengan *wall following* menggunakan ultrasonik belakang dengan gerak ke kanan.

#### 4.3.8 Pengujian Robot Menggunakan Ultrasonik Belakang Dengan Gerak Ke Kiri

Pengujian ini dilakukan untuk mengetahui apakah robot dapat menelusuri objek dinding menggunakan ultrasonik belakang dengan gerak ke kiri.

Pada Gambar 4.15 menunjukkan robot menelusuri objek dinding menggunakan ultrasonik belakang dengan gerak ke kiri.



Gambar 4.15 Robot Menelusuri Objek Dinding Menggunakan Ultrasonik Belakang dengan Gerak ke Kiri

Kondisi *follow* sensor ultrasonik belakang dengan gerak ke kiri sebagai berikut.

1. Pada Robot 1, M1 tidak berputar, M2 dan M3 berputar ke depan (kondisi : robot menjauhi dinding/papan karena jarak pembacaan sensor ultrasonik lebih kecil daripada nilai batas yang ditentukan).
2. Pada Robot 2, M1 tidak berputar, M2 dan M3 berputar ke depan (kondisi : robot mendekati dinding/papan karena pembacaan jarak sensor ultrasonik lebih besar daripada nilai batas jarak yang ditetapkan).
3. Pada Robot 3, M1 berputar ke kiri, M2 berputar ke belakang, M3 berputar ke depan (kondisi : robot berjalan lurus karena nilai jarak dan batas yang ditentukan sama besarnya).

Hasil pengujian Osilasi Robot dan Standar Deviasi ditunjukkan pada Tabel 4.20. Pada Tabel 4.21 menunjukkan hasil pengujian robot menggunakan ultrasonik belakang dengan gerak ke kiri.

Tabel 4.20 Hasil Pengujian Osilasi Robot dan Standar Deviasi Robot Menggunakan Ultrasonik

Belakang dengan Gerak ke Kiri

Pengujian ke- (cm)				
1	2	3	4	5
15	16	16	17	19
16	16	16	17	18,3
16,3	16	15,4	18	18
15	16	15,4	18	18
15	16	15	17,6	17,9
14	16	16	16,8	17,6
15	17	16,7	15,6	16
16	17	16	15	16
16	16	16	14	15,2
16,4	16	15	15	15
17,1	15	15	15	14
17	14	14	14,5	15
17,3	15	15	15	14
17	15	15	15	15
$\bar{x}=15,9357$	$\bar{x}=15,7857$	$\bar{x}=15,4642$	$\bar{x}=15,9642$	$\bar{x}=16,3571$
SD=1,003538	SD=0,801784	SD=0,690158	SD=1,375392	SD=1,117526

Tabel 4.21 Hasil Pengujian Robot Menggunakan Ultrasonik Belakang dengan Gerak ke Kiri

Pengujian ke-	Jumlah Benturan	Jarak Rata-Rata Osilasi	Lama Waktu (detik)	Hasil Pengujian
1	-	15,93571	14,4	Berhasil
2	-	15,78571	14	Berhasil
3	-	15,46429	13,5	Berhasil
4	-	15,96429	14,4	Berhasil
5	-	16,35714	15,2	Berhasil

Berdasarkan hasil pengujian dari Tabel 4.20 Pengambilan data dilaksanakan setiap jarak 50 cm dari 0 cm sampai 700 cm sehingga dapat diketahui bahwa jarak



rata-rata keseluruhan robot berisolasikan adalah sebesar 15,90143 cm dan Standar Deviasi rata-rata sebesar 1,117526 cm. Pada hasil pengujian Tabel 4.21 diketahui bahwa dari total 5 kali pengujian, robot dapat menyelesaikan tugasnya bernalavigasi tanpa berbenturan dengan objek dinding. Untuk rata-rata waktu bernalavigasi yakni 14,3 detik dengan jarak navigasi sepanjang 7 meter. Sehingga dapat dikatakan bahwa robot mampu bernalavigasi dengan *wall following* menggunakan ultrasonik belakang dengan gerak ke kiri.



## 5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis tiap bagian dan keseluruhan sistem yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut :

1. Akses sensor ultrasonik PING))) memiliki kesalahan rata-rata pengukuran sebesar 0,568 cm. Kesalahan pengukuran ini tidak berpengaruh pada keseluruhan sistem karena yang digunakan dalam pembacaan adalah nilai desimal di depan koma.
2. Respon sinyal keluaran *duty cycle* driver motor L298N terhadap masukan pwm mikrokontroler memiliki kesalahan rata-rata 6,9%. Nilai kesalahan ini tidak memberikan pengaruh yang besar pada kinerja sistem yang dirancang.
3. Robot KRPAI beroda dibentuk dengan *platform* segitiga dengan pemasangan tiga buah *omni wheel* yang terpisah sejauh  $120^\circ$ .
4. Dengan menggunakan *Omni Wheel*, robot mampu mengikuti dinding dengan waktu rata-rata 13,7225 detik sejauh 7 meter dengan berisolasikan rata-rata keseluruhan sebesar 15,572 cm dan tidak mengalami benturan terhadap dinding.

## 5.2 Saran

1. Pengimplementasian sistem navigasi *wall following* robot pada arena Kontes Robot Pemadam Api Indonesia (KRPAI).
2. Penerapan kontroler PID pada robot agar membuat pergerakan robot menjadi stabil.



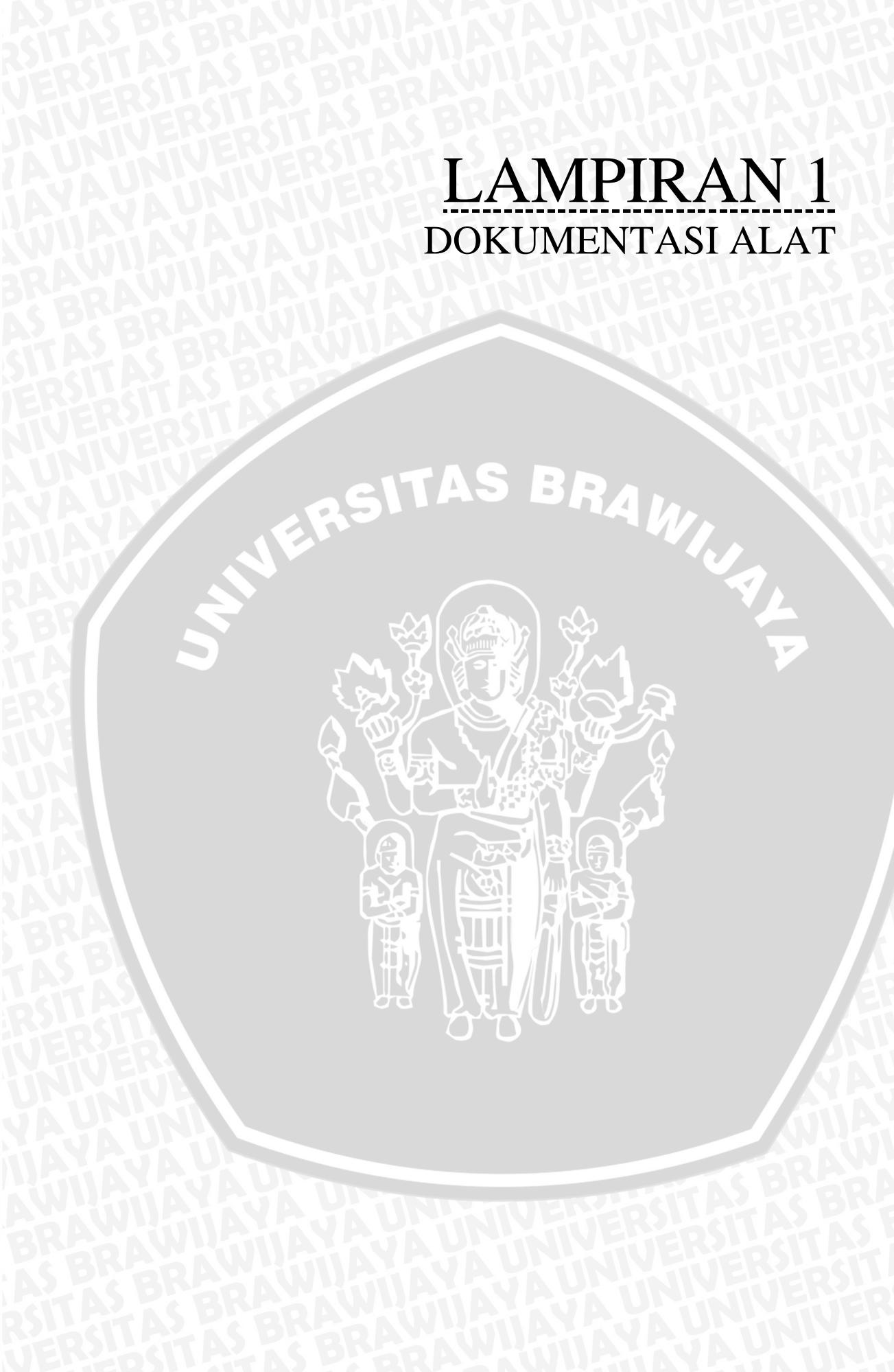
## DAFTAR PUSTAKA

- Akbar, Arnas Elmiawan. 2013. *Implementasi Sistem Navigasi Wall Following Menggunakan Kontroler PID Dengan Metode Tuning Pada Robot Kontes Robot Cerdas Indonesia (KRCI) Divisi Senior Beroda*. Malang: Skripsi Jurusan Teknik Elektro FT-UB.
- Allied Electronics. *Arduino Mega 2560*. <http://www.alliedelec.com/arduino-a000067/70415155/>, diakses pada 1 Oktober 2016.
- Instructables. *Arduino Modules – L298N Dual H-Bridge Motor Controller*. <http://www.instructables.com/id/Arduino-Modules-L298N-Dual-H-Bridge-Motor-Controll/>, diakses tanggal 10 Oktober 2016.
- Nexus Robot. *3WD 48mm Omni Wheel Arduino Mobile Robot Kit 10019*.
- Rafiuddin Syam, Irham & Widhi Erlangga. 2012. Rancang Bangun *Omni Wheels Robot* Dengan Roda Penggerak Independent. *Jurnal Mekanikal*. Vol. 3 No.1: 213-220.
- RISTEKDIKTI. 2015. *Panduan Kontes Robot Pemadam Api Indonesia 2015*. Jakarta: RISTEKDIKTI.
- STMicroelectronics. 2010. *L298, DUAL FULL-BRIDGE DRIVER*. [https://www.sparkfun.com/datasheets/Robotics/L298\\_H\\_Bridge.pdf](https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf), diakses tanggal 7 Oktober 2016.

# LAMPIRAN 1

---

## DOKUMENTASI ALAT



UNIVERSITAS BRAWIJAYA



UNIV

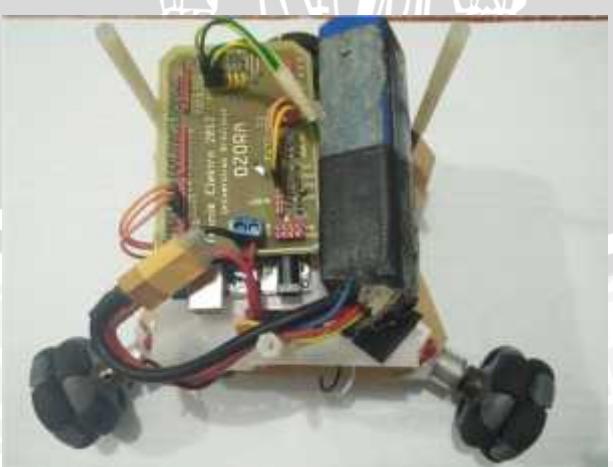
IJAYA



Gambar 1.1 *Shield Board* Arduino Mega 2560



Gambar 1.2 Body Bawah, Modul Driver L298N, Motor DC, dan *Omni Wheel*



Gambar 1.3 Keseluruhan Bodi Tengah



Gambar 1.4 Bodi Atas dan Posisi Ultrasonik



Gambar 1.5 Keseluruhan Robot

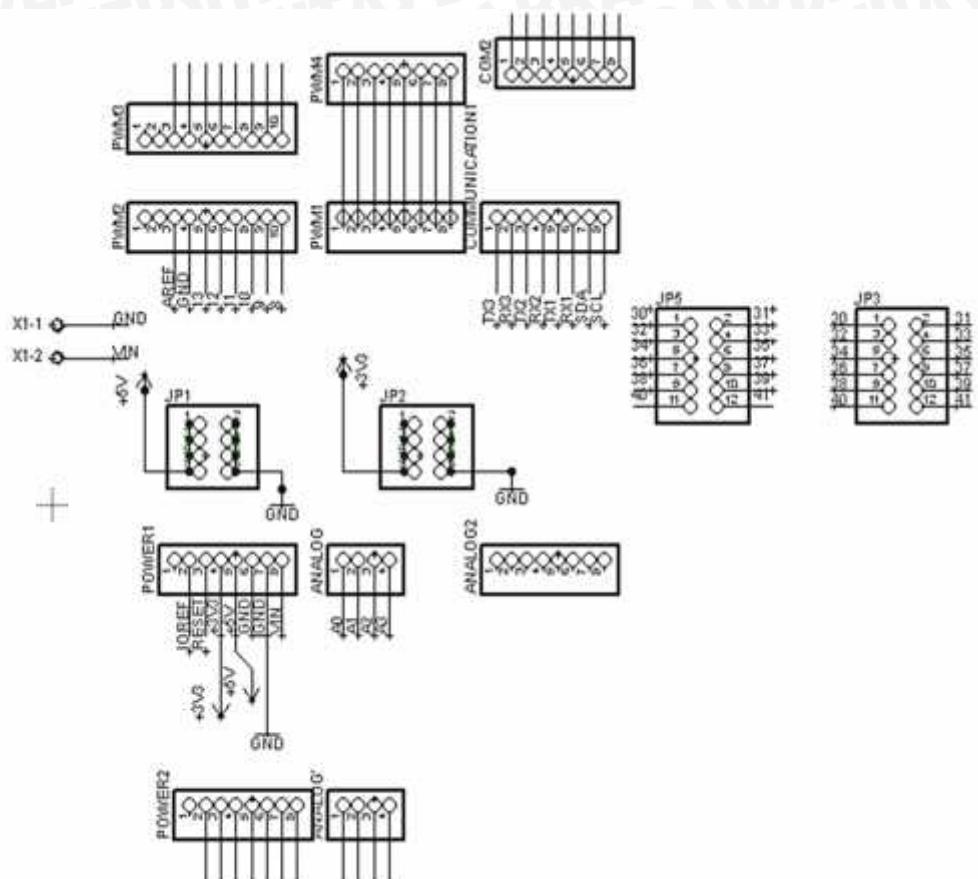
## LAMPIRAN 2

### RANGKAIAN SHIELD PCB

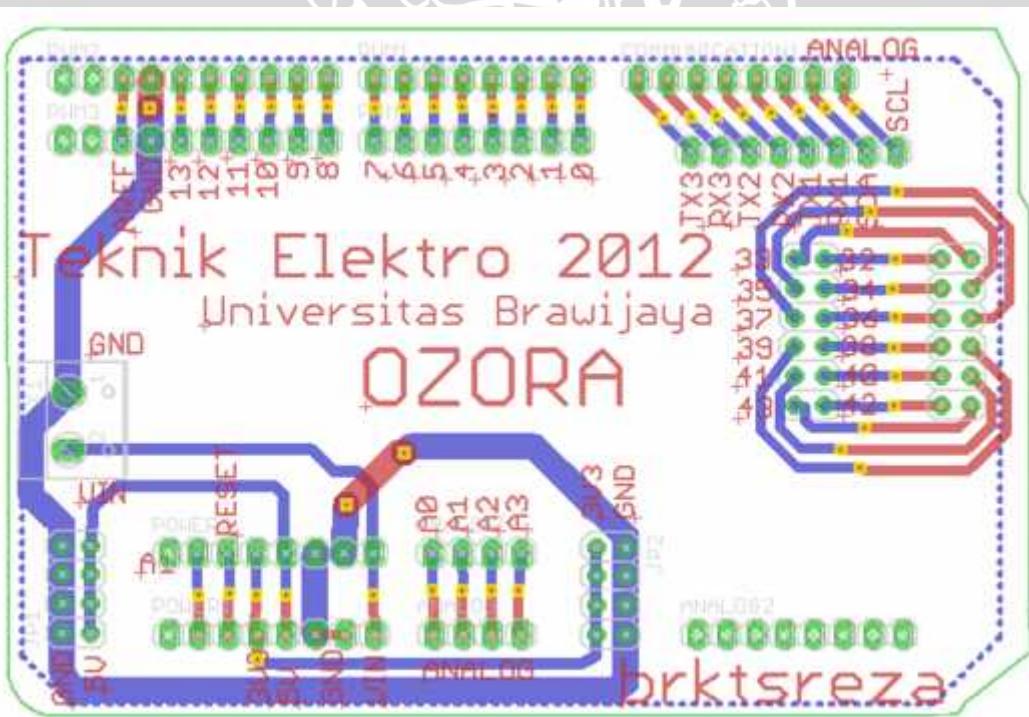


UNIVERSITAS BRAWIJAYA





Gambar 1.1 Jembatan Rangkaian *Shield* Arduino Mega 2560



Gambar 1.2 Board PCB Rangkaian Shield Arduino Mega 2560

# LAMPIRAN 3

## *LISTING PROGRAM*



UNIVERSITAS BRAWIJAYA



## 1. Listing Program Robot Menggunakan Ultrasonik Depan Dengan Gerak Ke Kanan

```
long duration, Depankanancm;
```

```
void setup() {  
Serial.begin (9600);
```

```
// put your setup code here, to run once:  
pinMode (32,OUTPUT);  
pinMode (33,OUTPUT);  
pinMode (34,OUTPUT);  
pinMode (35,OUTPUT);  
pinMode (36,OUTPUT);  
pinMode (37,OUTPUT);  
pinMode (13,OUTPUT);  
pinMode (12,OUTPUT);  
pinMode (11,OUTPUT);  
}
```

```
void loop(){
```

```
//pemangilan fungsi sensor ping  
sensor_ping(41);  
if (Depankanancm > 22)  
{  
motor1(0);  
motor2(-130);  
motor3(130);  
}  
else if (Depankanancm < 15)  
{  
motor1(0);  
motor2(130);  
motor3(-130);  
}  
else  
{  
motor1(-180);  
motor2(95);  
motor3(95);  
}  
}  
void sensor_ping(const int pingPin)
```

```
{
```

```
pinMode(pingPin, OUTPUT);  
digitalWrite(pingPin, LOW);
```

```
//memberikan trigger pulsa high  
digitalWrite(pingPin, HIGH);  
digitalWrite(pingPin, LOW);  
pinMode(pingPin, INPUT);
```

```
duration = pulseIn(pingPin, HIGH);  
Depankanancm = duration / 29 / 2;
```

```
//menampilkan data pembacaan sensor  
// Serial.print(duration);  
// Serial.print(" uS | ");  
Serial.print(Depankanancm);  
Serial.print(" cm");  
Serial.println();  
//delay(10);  
}
```

```
void motor1(int pwm)  
{  
if(pwm<0)  
{  
pwm=pwm*-1;  
digitalWrite(32,LOW);  
digitalWrite(33,HIGH);  
analogWrite(13,pwm);  
}  
else if (pwm>0)  
{  
digitalWrite(32,HIGH);  
digitalWrite(33,LOW);  
analogWrite(13,pwm);  
}  
else  
{  
digitalWrite(32,LOW);  
digitalWrite(33,LOW);  
}  
}
```

```
void motor2(int pwm)
```

```

{
  if(pwm<0)
  {
    pwm=pwm*-1;
    digitalWrite(34,LOW);
    digitalWrite(35,HIGH);
    analogWrite(12,pwm);
  }
  else if (pwm>0)
  {
    digitalWrite(34,HIGH);
    digitalWrite(35,LOW);
    analogWrite(12,pwm);
  }
  else
  {
    digitalWrite(34,LOW);
    digitalWrite(35,LOW);
  }
}

void motor3(int pwm)
{
  if(pwm<0)
  {
    pwm=pwm*-1;
    digitalWrite(36,LOW);
    digitalWrite(37,HIGH);
    analogWrite(11,pwm);
  }
  else if (pwm>0)
  {
    digitalWrite(36,HIGH);
    digitalWrite(37,LOW);
    analogWrite(11,pwm);
  }
  else
  {
    digitalWrite(36,LOW);
    digitalWrite(37,LOW);
  }
}
  
```

```

// put your setup code here, to run once:
pinMode (32,OUTPUT);
pinMode (33,OUTPUT);
pinMode (34,OUTPUT);
pinMode (35,OUTPUT);
pinMode (36,OUTPUT);
pinMode (37,OUTPUT);
pinMode (13,OUTPUT);
pinMode (12,OUTPUT);
pinMode (11,OUTPUT);

}

void loop(){
  //pemanggilan fungsi sensor ping
  sensor_ping(41);
  if (Depankiricm > 22)
  {
    motor1(0);
    motor2(-130);
    motor3(130);
  }
  else if (Depankiricm < 15)
  {
    motor1(0);
    motor2(130);
    motor3(-130);
  }
  else
  {
    motor1(180);
    motor2(-95);
    motor3(-95);
  }
}

void sensor_ping(const int pingPin)
{
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);

  //memberikan trigger pulsa high
  digitalWrite(pingPin, HIGH);
  digitalWrite(pingPin, LOW);
  pinMode(pingPin, INPUT);
}

duration = pulseIn(pingPin, HIGH);
Depankiricm = duration / 29 / 2;
  
```

```

duration = pulseIn(pingPin, HIGH);
Depankiricm = duration / 29 / 2;

//menampilkan data pembacaan sensor
// Serial.print(duration);
// Serial.print(" uS | ");
Serial.print(Depankiricm);
Serial.print(" cm");
Serial.println();
  
```

## 2. Listing Program Robot Menggunakan Ultrasonik Depan Dengan Gerak Ke Kiri

```

long duration, Depankiricm;

void setup() {
  Serial.begin (9600);
  
```

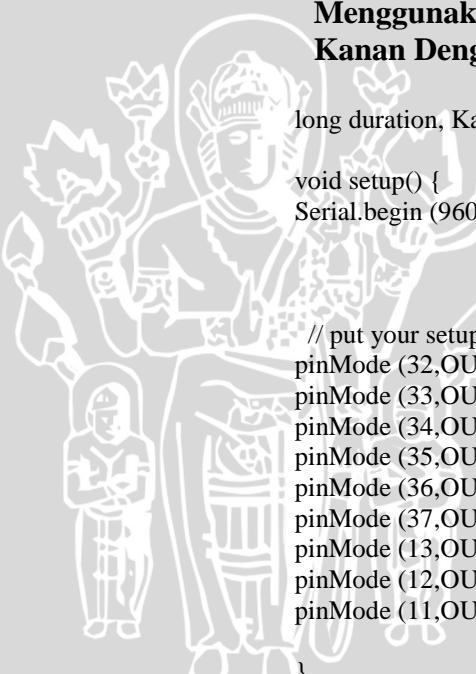
```
//delay(10);
}

void motor1(int pwm)
{
    if(pwm<0)
    {
        pwm=pwm*-1;
        digitalWrite(36,LOW);
        digitalWrite(37,HIGH);
        analogWrite(11,pwm);
    }
    else if (pwm>0)
    {
        digitalWrite(32,HIGH);
        digitalWrite(33,LOW);
        analogWrite(13,pwm);
    }
    else
    {
        digitalWrite(32,LOW);
        digitalWrite(33,LOW);
    }
}

void motor2(int pwm)
{
    if(pwm<0)
    {
        pwm=pwm*-1;
        digitalWrite(34,LOW);
        digitalWrite(35,HIGH);
        analogWrite(12,pwm);
    }
    else if (pwm>0)
    {
        digitalWrite(34,HIGH);
        digitalWrite(35,LOW);
        analogWrite(12,pwm);
    }
    else
    {
        digitalWrite(34,LOW);
        digitalWrite(35,LOW);
    }
}

void motor3(int pwm)
{
    if(pwm<0)
```

### 3. Listing Program Robot Menggunakan Ultrasonik Kanan Dengan Gerak Maju

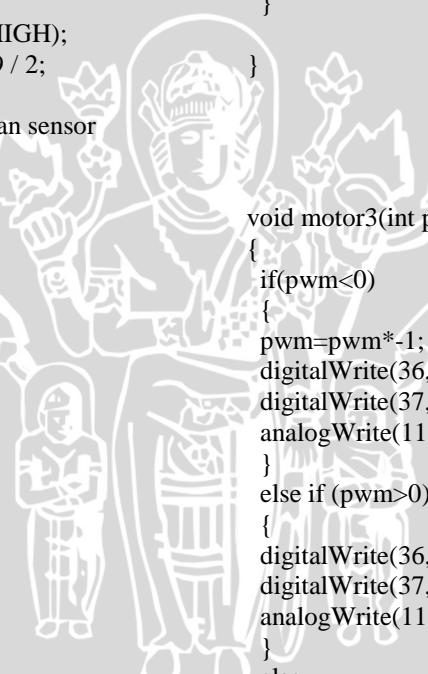


```
long duration, Kananmajucm;

void setup() {
    Serial.begin (9600);

    // put your setup code here, to run once:
    pinMode (32,OUTPUT);
    pinMode (33,OUTPUT);
    pinMode (34,OUTPUT);
    pinMode (35,OUTPUT);
    pinMode (36,OUTPUT);
    pinMode (37,OUTPUT);
    pinMode (13,OUTPUT);
    pinMode (12,OUTPUT);
    pinMode (11,OUTPUT);
}

void loop(){
    //pemangilan fungsi sensor ping
    sensor_ping(41);
    if (Kananmajucm > 22)
    {
        motor1(180);
        motor2(-95);
        motor3(-95);
    }
    else if (Kananmajucm < 15)
    {
        motor1(-180);
        motor2(95);}
```



```
motor3(95);
    }
else
{
    motor1(0);
    motor2(130);
    motor3(-130);
}
void sensor_ping(const int pingPin)
{
    pinMode(pingPin, OUTPUT);
    digitalWrite(pingPin, LOW);

    //memberikan trigger pulsa high
    digitalWrite(pingPin, HIGH);
    digitalWrite(pingPin, LOW);
    pinMode(pingPin, INPUT);

    duration = pulseIn(pingPin, HIGH);
    Kananmajucm = duration / 29 / 2;

    //menampilkan data pembacaan sensor
    // Serial.print(duration);
    // Serial.print(" uS | ");
    Serial.print(Kananmajucm);
    Serial.print(" cm");
    Serial.println();
    delay(10);
}

void motor1(int pwm)
{
    if(pwm<0)
    {
        pwm=pwm*-1;
        digitalWrite(32,LOW);
        digitalWrite(33,HIGH);
        analogWrite(13,pwm);
    }
    else if (pwm>0)
    {
        digitalWrite(32,HIGH);
        digitalWrite(33,LOW);
        analogWrite(13,pwm);
    }
    else
    {
        digitalWrite(32,LOW);
        digitalWrite(33,LOW);
    }
}

void motor2(int pwm)
{
    if(pwm<0)
    {
        pwm=pwm*-1;
        digitalWrite(34,LOW);
        digitalWrite(35,HIGH);
        analogWrite(12,pwm);
    }
    else if (pwm>0)
    {
        digitalWrite(34,HIGH);
        digitalWrite(35,LOW);
        analogWrite(12,pwm);
    }
    else
    {
        digitalWrite(34,LOW);
        digitalWrite(35,LOW);
    }
}

void motor3(int pwm)
{
    if(pwm<0)
    {
        pwm=pwm*-1;
        digitalWrite(36,LOW);
        digitalWrite(37,HIGH);
        analogWrite(11,pwm);
    }
    else if (pwm>0)
    {
        digitalWrite(36,HIGH);
        digitalWrite(37,LOW);
        analogWrite(11,pwm);
    }
    else
    {
        digitalWrite(36,LOW);
        digitalWrite(37,LOW);
    }
}
```

#### 4. Listing Program Robot Menggunakan Ultrasonik Kanan Dengan Gerak Mundur

long duration, Kananmundurcm;

```
void setup() {
Serial.begin (9600);
```

```
// put your setup code here, to run once:  
pinMode (32,OUTPUT);  
pinMode (33,OUTPUT);  
pinMode (34,OUTPUT);  
pinMode (35,OUTPUT);  
pinMode (36,OUTPUT);  
pinMode (37,OUTPUT);  
pinMode (13,OUTPUT);  
pinMode (12,OUTPUT);  
pinMode (11,OUTPUT);  
  
}  
  
void loop(){  
  
//pemangilan fungsi sensor ping  
sensor_ping(41);  
if (Kananmundurcm > 22)  
{  
motor1(180);  
motor2(-95);  
motor3(-95);  
}  
else if (Kananmundurcm < 15)  
{  
motor1(-180);  
motor2(95);  
motor3(95);  
}  
else  
{  
motor1(0);  
motor2(-130);  
motor3(130);  
}  
}  
  
void sensor_ping(const int pingPin)  
{  
  
pinMode(pingPin, OUTPUT);  
digitalWrite(pingPin, LOW);  
  
//memberikan trigger pulsa high  
digitalWrite(pingPin, HIGH);  
digitalWrite(pingPin, LOW);  
pinMode(pingPin, INPUT);  
  
duration = pulseIn(pingPin, HIGH);  
Kananmundurcm = duration / 29 / 2;  
  
//menampilkan data pembacaan sensor  
// Serial.print(duration);  
// Serial.print(" uS | ");  
Serial.print(Kananmundurcm);  
  
Serial.print(" cm");  
// Serial.println();  
// delay(10);  
}  
  
void motor1(int pwm)  
{  
if(pwm<0)  
{  
pwm=pwm*-1;  
digitalWrite(32,LOW);  
digitalWrite(33,HIGH);  
analogWrite(13,pwm);  
}  
else if (pwm>0)  
{  
digitalWrite(32,HIGH);  
digitalWrite(33,LOW);  
analogWrite(13,pwm);  
}  
else  
{  
digitalWrite(32,LOW);  
digitalWrite(33,LOW);  
}  
}  
  
void motor2(int pwm)  
{  
if(pwm<0)  
{  
pwm=pwm*-1;  
digitalWrite(34,LOW);  
digitalWrite(35,HIGH);  
analogWrite(12,pwm);  
}  
else if (pwm>0)  
{  
digitalWrite(34,HIGH);  
digitalWrite(35,LOW);  
analogWrite(12,pwm);  
}  
else  
{  
digitalWrite(34,LOW);  
digitalWrite(35,LOW);  
}  
}  
  
void motor3(int pwm)
```

```
{  
    if(pwm<0)  
    {  
        pwm=pwm*-1;  
        digitalWrite(36,LOW);  
        digitalWrite(37,HIGH);  
        analogWrite(11,pwm);  
    }  
    else if (pwm>0)  
    {  
        digitalWrite(36,HIGH);  
        digitalWrite(37,LOW);  
        analogWrite(11,pwm);  
    }  
    else  
    {  
        digitalWrite(36,LOW);  
        digitalWrite(37,LOW);  
    }  
}  
  
5. Listing Program Robot Menggunakan Ultrasonik Kiri Dengan Gerak Maju  
  
long duration, Kirimajucm;  
  
void setup() {  
Serial.begin (9600);  
  
    // put your setup code here, to run once:  
pinMode (32,OUTPUT);  
pinMode (33,OUTPUT);  
pinMode (34,OUTPUT);  
pinMode (35,OUTPUT);  
pinMode (36,OUTPUT);  
pinMode (37,OUTPUT);  
pinMode (13,OUTPUT);  
pinMode (12,OUTPUT);  
pinMode (11,OUTPUT);  
}  
  
void loop(){  
  
//pemangilan fungsi sensor ping  
sensor_ping(41);  
if (Kirimajucm > 22)  
{  
    motor1(-180);  
    motor2(95);  
    motor3(95);  
}  
else if (Kirimajucm < 15)  
{  
    if(pwm<0)  
    {  
        pwm=pwm*-1;  
        digitalWrite(36,LOW);  
        digitalWrite(37,HIGH);  
        analogWrite(11,pwm);  
    }  
    else if (pwm>0)  
    {  
        digitalWrite(36,HIGH);  
        digitalWrite(37,LOW);  
        analogWrite(11,pwm);  
    }  
    else  
    {  
        digitalWrite(36,LOW);  
        digitalWrite(37,LOW);  
    }  
}  
motor1(180);  
motor2(-95);  
motor3(-95);  
}  
else  
{  
    motor1(0);  
    motor2(-130);  
    motor3(130);  
}  
}  
}  
void sensor_ping(const int pingPin)  
{  
pinMode(pingPin, OUTPUT);  
digitalWrite(pingPin, LOW);  
// delayMicroseconds(2);  
  
//memberikan trigger pulsa high selama  
2-5 us  
digitalWrite(pingPin, HIGH);  
// delayMicroseconds(5);  
digitalWrite(pingPin, LOW);  
pinMode(pingPin, INPUT);  
  
duration = pulseIn(pingPin, HIGH);  
Kirimajucm = duration / 29 / 2;  
  
//menampilkan data pembacaan sensor  
// Serial.print(duration);  
// Serial.print(" uS | ");  
Serial.print(Kirimajucm);  
Serial.print(" cm");  
Serial.println();  
delay(10);  
}  
  
void motor1(int pwm)  
{  
    if(pwm<0)  
    {  
        pwm=pwm*-1;  
        digitalWrite(32,LOW);  
        digitalWrite(33,HIGH);  
        analogWrite(13,pwm);  
    }  
    else if (pwm>0)  
    {  
        digitalWrite(32,HIGH);  
        digitalWrite(33,LOW);  
        analogWrite(13,pwm);  
    }  
    else  
    {  
        digitalWrite(32,LOW);  
        digitalWrite(33,LOW);  
    }  
}
```

```

        long duration, Kirimundurcm;
    }

    void motor2(int pwm)
    {
        if(pwm<0)
        {
            pwm=pwm*-1;
            digitalWrite(34,LOW);
            digitalWrite(35,HIGH);
            analogWrite(12,pwm);
        }
        else if (pwm>0)
        {
            digitalWrite(34,HIGH);
            digitalWrite(35,LOW);
            analogWrite(12,pwm);
        }
        else
        {
            digitalWrite(34,LOW);
            digitalWrite(35,LOW);
        }
    }

    void motor3(int pwm)
    {
        if(pwm<0)
        {
            pwm=pwm*-1;
            digitalWrite(36,LOW);
            digitalWrite(37,HIGH);
            analogWrite(11,pwm);
        }
        else if (pwm>0)
        {
            digitalWrite(36,HIGH);
            digitalWrite(37,LOW);
            analogWrite(11,pwm);
        }
        else
        {
            digitalWrite(36,LOW);
            digitalWrite(37,LOW);
        }
    }

    void setup()
    {
        Serial.begin (9600);

        // put your setup code here, to run once:
        pinMode (32,OUTPUT);
        pinMode (33,OUTPUT);
        pinMode (34,OUTPUT);
        pinMode (35,OUTPUT);
        pinMode (36,OUTPUT);
        pinMode (37,OUTPUT);
        pinMode (13,OUTPUT);
        pinMode (12,OUTPUT);
        pinMode (11,OUTPUT);
    }

    void loop()
    {
        //pemanggilan fungsi sensor ping
        sensor_ping(41);
        if (Kirimundurcm > 22)
        {
            motor1(-180);
            motor2(95);
            motor3(95);
        }
        else if (Kirimundurcm < 15)
        {
            motor1(180);
            motor2(-95);
            motor3(-95);
        }
        else
        {
            motor1(0);
            motor2(130);
            motor3(-130);
        }
    }

    void sensor_ping(const int pingPin)
    {
        pinMode(pingPin, OUTPUT);
        digitalWrite(pingPin, LOW);
        // delayMicroseconds(2);

        //memberikan trigger pulsa high selama
        2-5 us
        digitalWrite(pingPin, HIGH);
        // delayMicroseconds(5);
        digitalWrite(pingPin, LOW);
        pinMode(pingPin, INPUT);

        duration = pulseIn(pingPin, HIGH);
    }
}

```

## 6. Listing Program Robot Menggunakan Ultrasonik Kiri Dengan Gerak Mundur

```
Kirimundurcm = duration / 29 / 2;
```

```
//menampilkan data pembacaan sensor  
// Serial.print(duration);  
// Serial.print(" uS | ");  
Serial.print(Kirimundurcm);  
Serial.print(" cm");  
Serial.println();  
delay(10);  
}
```

```
void motor1(int pwm)  
{  
    if(pwm<0)  
    {  
        pwm=pwm*-1;  
        digitalWrite(32,LOW);  
        digitalWrite(33,HIGH);  
        analogWrite(13,pwm);  
    }  
    else if (pwm>0)  
    {  
        digitalWrite(32,HIGH);  
        digitalWrite(33,LOW);  
        analogWrite(13,pwm);  
    }  
    else  
    {  
        digitalWrite(32,LOW);  
        digitalWrite(33,LOW);  
    }  
}
```

```
void motor2(int pwm)  
{  
    if(pwm<0)  
    {  
        pwm=pwm*-1;  
        digitalWrite(34,LOW);  
        digitalWrite(35,HIGH);  
        analogWrite(12,pwm);  
    }  
    else if (pwm>0)  
    {  
        digitalWrite(34,HIGH);  
        digitalWrite(35,LOW);  
        analogWrite(12,pwm);  
    }  
    else  
    {  
        digitalWrite(34,LOW);  
        digitalWrite(35,LOW);  
    }  
}
```

```
void motor3(int pwm)  
{  
    if(pwm<0)  
    {  
        pwm=pwm*-1;  
        digitalWrite(36,LOW);  
        digitalWrite(37,HIGH);  
        analogWrite(11,pwm);  
    }  
    else if (pwm>0)  
    {  
        digitalWrite(36,HIGH);  
        digitalWrite(37,LOW);  
        analogWrite(11,pwm);  
    }  
    else  
    {  
        digitalWrite(36,LOW);  
        digitalWrite(37,LOW);  
    }  
}
```

## 7. Listing Program Robot Menggunakan Ultrasonik Belakang Dengan Gerak Ke Kanan

```
long duration, Belakangkanancm;
```

```
void setup() {  
Serial.begin (9600);
```

```
// put your setup code here, to run once:  
pinMode (32,OUTPUT);  
pinMode (33,OUTPUT);  
pinMode (34,OUTPUT);  
pinMode (35,OUTPUT);  
pinMode (36,OUTPUT);  
pinMode (37,OUTPUT);  
pinMode (13,OUTPUT);  
pinMode (12,OUTPUT);  
pinMode (11,OUTPUT);
```

```
}
```

```
void loop(){
```

```
//pemanggilan fungsi sensor ping  
sensor_ping(41);  
if (Belakangkanancm > 22)
```

```
{  
    motor1(0);  
    motor2(-130);  
    motor3(130);  
}  
else if (Belakangkanancm < 15)  
{  
    motor1(0);  
    motor2(130);  
    motor3(-130);  
}  
else  
{  
    motor1(-180);  
    motor2(95);  
    motor3(95);  
}  
}  
void sensor_ping(const int pingPin)  
{  
  
    pinMode(pingPin, OUTPUT);  
    digitalWrite(pingPin, LOW);  
  
    //memberikan trigger pulsa high  
    digitalWrite(pingPin, HIGH);  
    digitalWrite(pingPin, LOW);  
    pinMode(pingPin, INPUT);  
  
    duration = pulseIn(pingPin, HIGH);  
    Belakangkanancm = duration / 29 / 2;  
  
    //menampilkan data pembacaan sensor  
    // Serial.print(duration);  
    // Serial.print(" uS | ");  
    Serial.print(Belakangkanancm);  
    Serial.print(" cm");  
    Serial.println();  
    delay(10);  
}  
  
void motor1(int pwm)  
{  
    if(pwm<0)  
    {  
        pwm=pwm*-1;  
        digitalWrite(32,LOW);  
        digitalWrite(33,HIGH);  
        analogWrite(13,pwm);  
    }  
    else if (pwm>0)  
    {  
        digitalWrite(32,HIGH);  
        digitalWrite(33,LOW);  
        analogWrite(13,pwm);  
    }  
}  
  
else  
{  
    digitalWrite(32,LOW);  
    digitalWrite(33,LOW);  
}  
}  
  
void motor2(int pwm)  
{  
    if(pwm<0)  
    {  
        pwm=pwm*-1;  
        digitalWrite(34,LOW);  
        digitalWrite(35,HIGH);  
        analogWrite(12,pwm);  
    }  
    else if (pwm>0)  
    {  
        digitalWrite(34,HIGH);  
        digitalWrite(35,LOW);  
        analogWrite(12,pwm);  
    }  
    else  
{  
        digitalWrite(34,LOW);  
        digitalWrite(35,LOW);  
    }  
}  
  
void motor3(int pwm)  
{  
    if(pwm<0)  
    {  
        pwm=pwm*-1;  
        digitalWrite(36,LOW);  
        digitalWrite(37,HIGH);  
        analogWrite(11,pwm);  
    }  
    else if (pwm>0)  
    {  
        digitalWrite(36,HIGH);  
        digitalWrite(37,LOW);  
        analogWrite(11,pwm);  
    }  
    else  
{  
        digitalWrite(36,LOW);  
        digitalWrite(37,LOW);  
    }  
}
```

## 8. Listing Program Robot Menggunakan Ultrasonik Belakang Dengan Gerak Ke Kiri

```
long duration, Belakangkircm;  
  
void setup() {  
Serial.begin (9600);  
  
// put your setup code here, to run  
once:  
pinMode (32,OUTPUT);  
pinMode (33,OUTPUT);  
pinMode (34,OUTPUT);  
pinMode (35,OUTPUT);  
pinMode (36,OUTPUT);  
pinMode (37,OUTPUT);  
pinMode (13,OUTPUT);  
pinMode (12,OUTPUT);  
pinMode (11,OUTPUT);  
  
}  
  
void loop()  
  
//pemangilan fungsi sensor ping  
sensor_ping(41);  
if (Belakangkircm > 22)  
{  
motor1(0);  
motor2(130);  
motor3(-130);  
}  
else if (Belakangkircm < 15)  
{  
motor1(0);  
motor2(-130);  
motor3(135);  
}  
else  
{  
motor1(180);  
motor2(-95);  
motor3(-95);  
}  
}  
  
void sensor_ping(const int  
pingPin)  
{  
  
pinMode(pingPin, OUTPUT);  
digitalWrite(pingPin, LOW);  
}
```

```
//memberikan trigger pulsa high  
digitalWrite(pingPin, HIGH);  
digitalWrite(pingPin, LOW);  
pinMode(pingPin, INPUT);
```

```
duration = pulseIn(pingPin,  
HIGH);  
Belakangkircm = duration / 29 /  
2;
```

```
//menampilkan data pembacaan  
sensor  
// Serial.print(duration);  
// Serial.print(" uS | ");  
Serial.print(Belakangkircm);  
Serial.print(" cm");  
Serial.println();  
delay(10);  
}
```

```
void motor1(int pwm)  
{  
if(pwm<0)  
{  
pwm=pwm*-1;  
digitalWrite(32,LOW);  
digitalWrite(33,HIGH);  
analogWrite(13,pwm);  
}  
else if (pwm>0)  
{  
digitalWrite(32,HIGH);  
digitalWrite(33,LOW);  
analogWrite(13,pwm);  
}  
else  
{  
digitalWrite(32,LOW);  
digitalWrite(33,LOW);  
}  
}
```

```
void motor2(int pwm)  
{  
if(pwm<0)  
{  
pwm=pwm*-1;  
digitalWrite(34,LOW);  
digitalWrite(35,HIGH);  
analogWrite(12,pwm);  
}  
else if (pwm>0)  
{  
digitalWrite(34,HIGH);  
}
```

```
digitalWrite(35,LOW);
analogWrite(12,pwm);
}
else
{
digitalWrite(34,LOW);
digitalWrite(35,LOW);
}

}

void motor3(int pwm)
{
if(pwm<0)
{



pwm=pwm*-1;
digitalWrite(36,LOW);
digitalWrite(37,HIGH);
analogWrite(11,pwm);
}
else if (pwm>0)
{
digitalWrite(36,HIGH);
digitalWrite(37,LOW);
analogWrite(11,pwm);
}
else
{
digitalWrite(36,LOW);
digitalWrite(37,LOW);
}
}
```



# LAMPIRAN 4

## DATASHEET



UNIVERSITAS BRAWIJAYA



## 1. Datasheet Driver L298N



L298

### DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

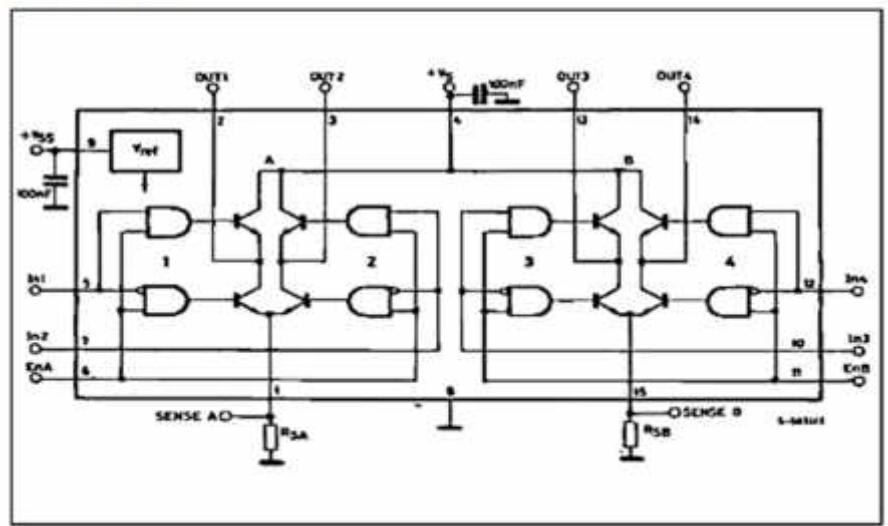
#### DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.



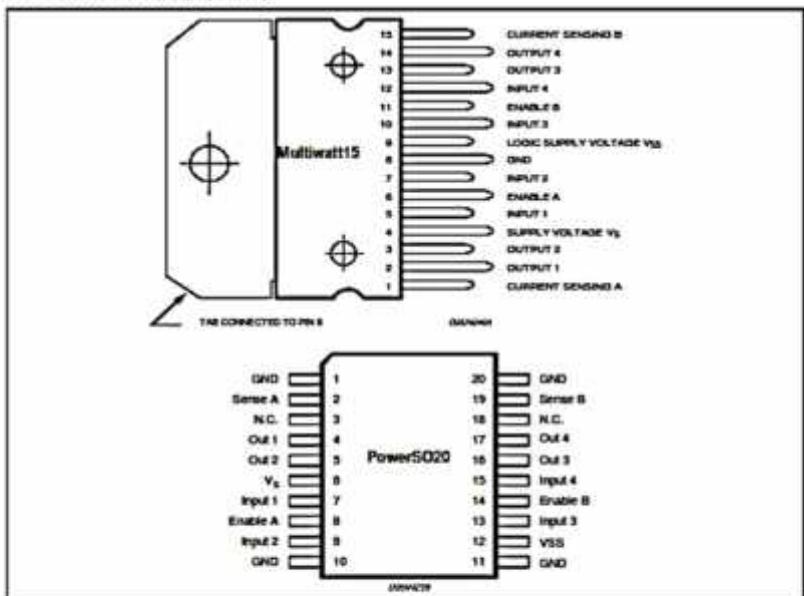
ORDERING NUMBERS : L298N (Multiwatt Vert.)  
L298HN (Multiwatt Horiz.)  
L298P (PowerSO20)

#### BLOCK DIAGRAM



**L298****ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	Value	Unit
$V_S$	Power Supply	30	v
$V_{SS}$	Logic Supply Voltage	7	v
$V_i, V_m$	Input and Enable Voltage	-0.3 to 7	v
$I_O$	Peak Output Current (each Channel)		
	- Non Repetitive ( $t = 100\mu s$ )	3	A
	- Repetitive (80% on -20% off; $t_m = 10ms$ )	2.5	A
	- DC Operation	2	A
$V_{sense}$	Sensing Voltage	-1 to 2.3	v
$P_{diss}$	Total Power Dissipation ( $T_{case} = 75^\circ C$ )	25	w
$T_{j}$	Junction Operating Temperature	-25 to 130	°C
$T_{stg}, T_j$	Storage and Junction Temperature	-40 to 150	°C

**PIN CONNECTIONS (top view)****THERMAL DATA**

Symbol	Parameter	PowerSO20	Multiwatt15	Unit
$R_{j-case}$	Thermal Resistance Junction-case	Max.	—	°C/W
$R_{j-amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	°C/W

(\*) Mounted on aluminum substrate





## PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1:15	2:19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2:3	4:5	Out 1; Out 2	Outputs of the Bridge A: the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V <sub>S</sub>	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5:7	7:9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6:11	8:14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	11,10,11,20	GND	Ground.
9	12	V <sub>SS</sub>	Supply Voltage for the Logic Blocks. A 100nF capacitor must be connected between this pin and ground.
10:12	13:15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13:14	16:17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3:18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS ( $V_S = 42V$ ;  $V_{SS} = 5V$ ,  $T_j = 25^\circ C$ ; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$V_S$	Supply Voltage (pin 4)	Operative Condition	$V_S = +2.5$		45	V
$V_{SS}$	Logic Supply Voltage (pin 9)			5	7	V
$I_S$	Quiescent Supply Current (pin 4)	$V_{SS} = H$ ; $I_L = 0$ $V_1 = L$ $V_3 = H$	4.5	5	22	mA
		$V_{SS} = L$ $V_1 = X$		50	70	mA
$I_{SS}$	Quiescent Current from $V_{SS}$ (pin 9)	$V_{SS} = H$ ; $I_L = 0$ $V_1 = L$ $V_3 = H$	24	35	mA	
		$V_{SS} = L$ $V_1 = X$		7	12	mA
$V_L$	Input Low Voltage (pins 3, 7, 10, 12)		-0.3		1.5	V
$V_H$	Input High Voltage (pins 3, 7, 10, 12)		2.3		$V_{SS}$	V
$I_L$	Low Voltage Input Current (pins 3, 7, 10, 12)	$V_1 = L$			-10	µA
$I_H$	High Voltage Input Current (pins 3, 7, 10, 12)	$V_1 = H \leq V_{SS} - 0.6V$		30	100	µA
$V_{EL}$ = L	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
$V_{EH}$ = H	Enable High Voltage (pins 6, 11)		2.3		$V_{SS}$	V
$I_{EL}$ = L	Low Voltage Enable Current (pins 6, 11)	$V_{EH} = L$			-10	µA
$I_{EH}$ = H	High Voltage Enable Current (pins 6, 11)	$V_{EH} = H \leq V_{SS} - 0.6V$		30	100	µA
$V_{CE(sat)}$	Source Saturation Voltage	$I_L = 1A$ $I_L = 2A$	0.85	1.35	1.7	V
$V_{CE(sat)}$	Sink Saturation Voltage	$I_L = 1A$ (D) $I_L = 2A$ (D)	0.85	1.2	1.6	V
$V_{CE}$	Total Drop	$I_L = 1A$ (D) $I_L = 2A$ (D)	1.80		3.2	V
$V_{SEN}$	Sensing Voltage (pins 1, 15)		-4 (1)		2	V

L298

## ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$T_1$ (V)	Source Current Turn-off Delay	0.5 V <sub>in</sub> to 0.9 I <sub>L</sub> (2): (4)		1.5		μs
$T_2$ (V)	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2): (4)		0.2		μs
$T_3$ (V)	Source Current Turn-on Delay	0.5 V <sub>in</sub> to 0.1 I <sub>L</sub> (2): (4)		2		μs
$T_4$ (V)	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2): (4)		0.7		μs
$T_5$ (V)	Sink Current Turn-off Delay	0.5 V <sub>in</sub> to 0.9 I <sub>L</sub> (3): (4)		0.7		μs
$T_6$ (V)	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3): (4)		0.25		μs
$T_7$ (V)	Sink Current Turn-on Delay	0.5 V <sub>in</sub> to 0.9 I <sub>L</sub> (3): (4)		1.6		μs
$T_8$ (V)	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3): (4)		0.2		μs
$f_C$ (V)	Commutation Frequency	I <sub>L</sub> = 2A	25	40	40	KHz
$T_1$ (V <sub>in</sub> )	Source Current Turn-off Delay	0.5 V <sub>in</sub> to 0.9 I <sub>L</sub> (2): (4)		3		μs
$T_2$ (V <sub>in</sub> )	Source Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (2): (4)		1		μs
$T_3$ (V <sub>in</sub> )	Source Current Turn-on Delay	0.5 V <sub>in</sub> to 0.1 I <sub>L</sub> (2): (4)		0.3		μs
$T_4$ (V <sub>in</sub> )	Source Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (2): (4)		0.4		μs
$T_5$ (V <sub>in</sub> )	Sink Current Turn-off Delay	0.5 V <sub>in</sub> to 0.9 I <sub>L</sub> (3): (4)		2.2		μs
$T_6$ (V <sub>in</sub> )	Sink Current Fall Time	0.9 I <sub>L</sub> to 0.1 I <sub>L</sub> (3): (4)		0.35		μs
$T_7$ (V <sub>in</sub> )	Sink Current Turn-on Delay	0.5 V <sub>in</sub> to 0.9 I <sub>L</sub> (3): (4)		0.25		μs
$T_8$ (V <sub>in</sub> )	Sink Current Rise Time	0.1 I <sub>L</sub> to 0.9 I <sub>L</sub> (3): (4)		0.1		μs

1) t<sub>1</sub> Sensing voltage can be -1 V for t < 50 μsec; in steady state V<sub>sense</sub> min. is -0.5 V.

2) See fig. 2.

3) See fig. 4.

4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

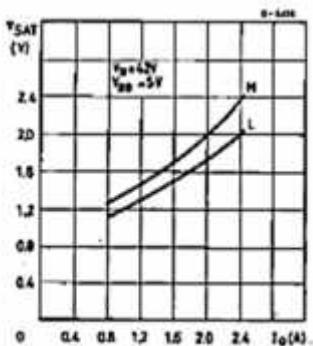


Figure 2 : Switching Times Test Circuits.

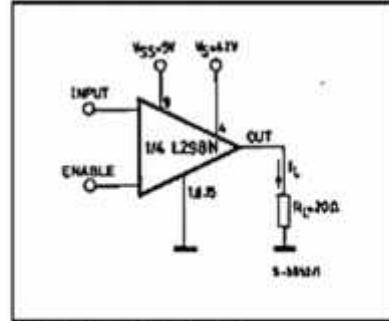


Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

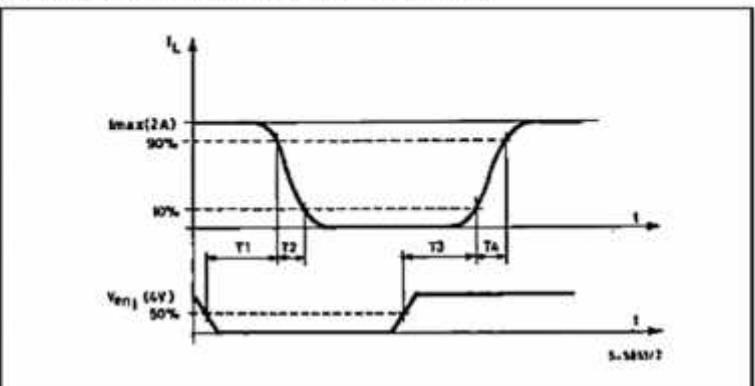
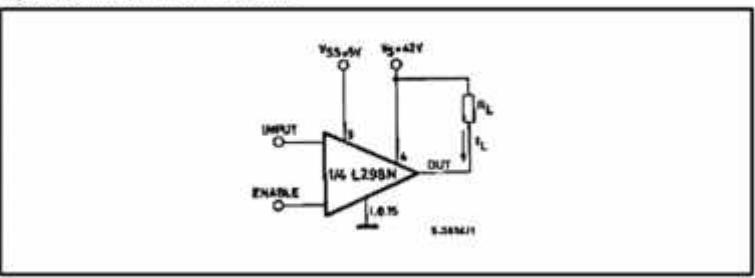


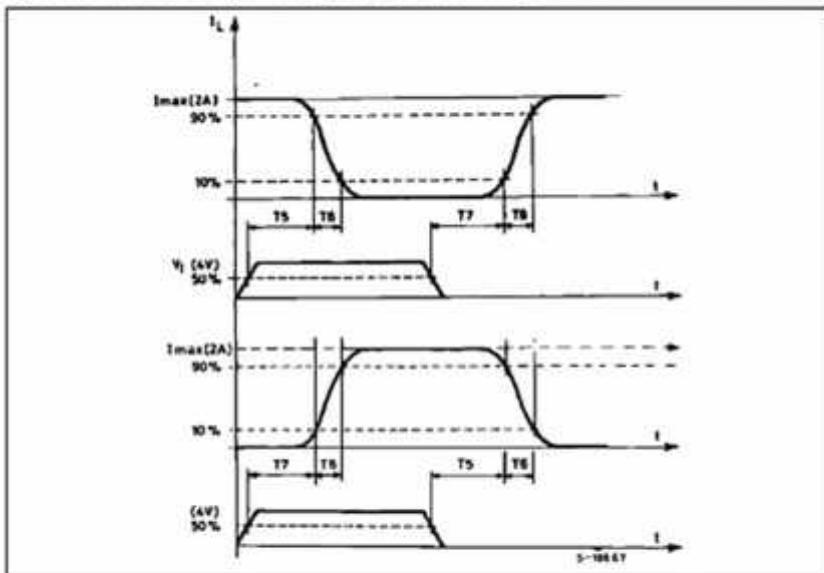
Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H  
For ENABLE Switching, set IN = L



**Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.**



**Figure 6 : Bidirectional DC Motor Control.**

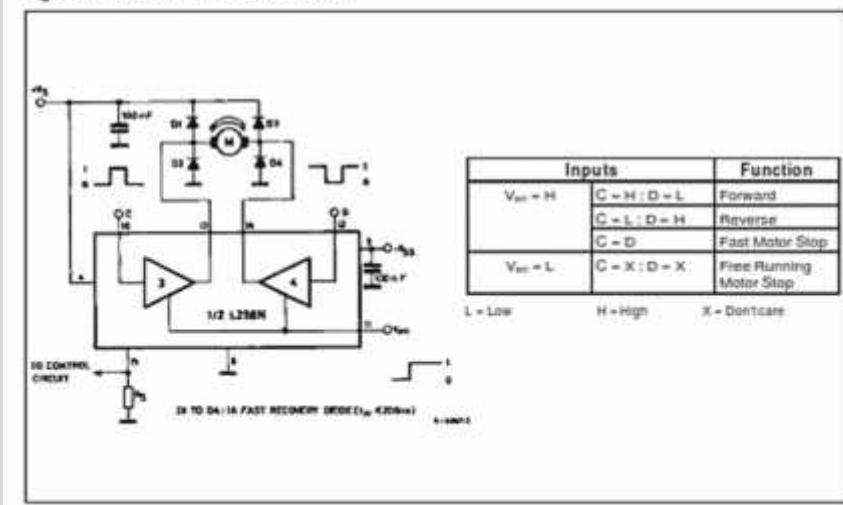
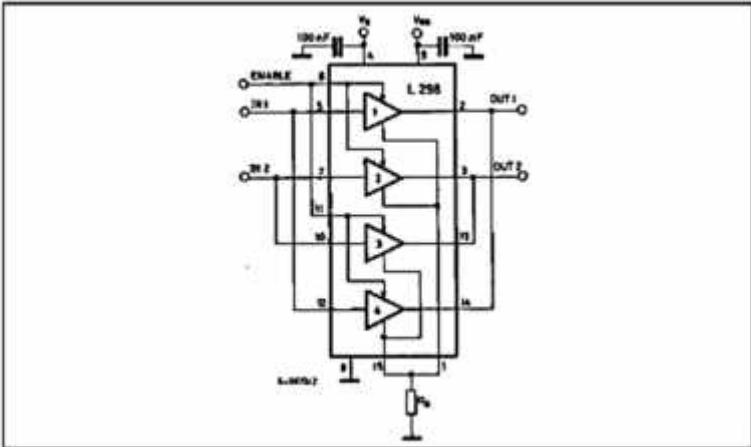




Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel4 and channel2 with channel3.



#### APPLICATION INFORMATION (Refer to the block diagram)

##### 1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differential mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output: an external resistor ( $R_{SA}$ ;  $R_{SB}$ ) allows to detect the intensity of this current.

##### 1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are  $In_1$ ;  $In_2$ ;  $En_A$  and  $In_3$ ;  $In_4$ ;  $En_B$ . The  $In$  inputs set the bridge state when the  $En$  input is high; a low state of the  $En$  input inhibits the bridge. All the inputs are TTL compatible.

##### 2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both  $V_s$  and  $V_{ss}$ , to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of  $V_s$  that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off: Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

##### 3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements ( $t_{rr} \leq 200$  nsec) that must be chosen of a VF as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped; Shottkydiodes would be preferred.





## L298

This solution can drive until 3 Amps in DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

**Figure 8 : Two Phase Bipolar Stepper Motor Circuit.**

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

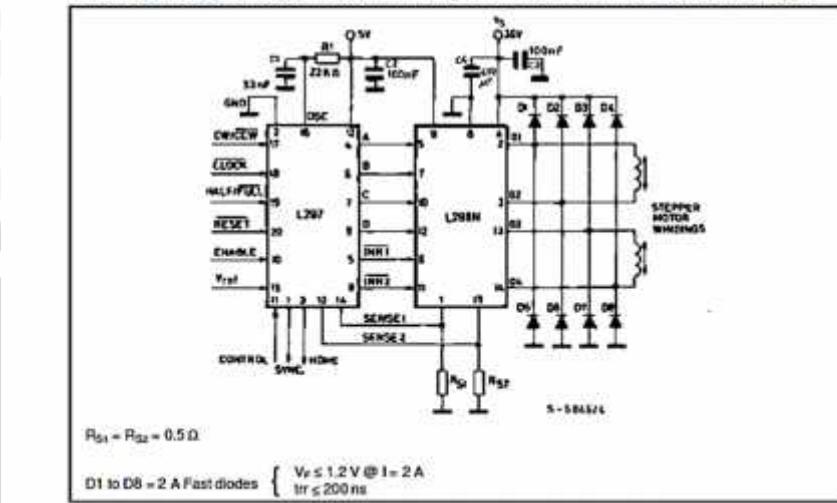


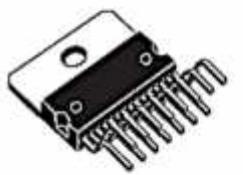
Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.



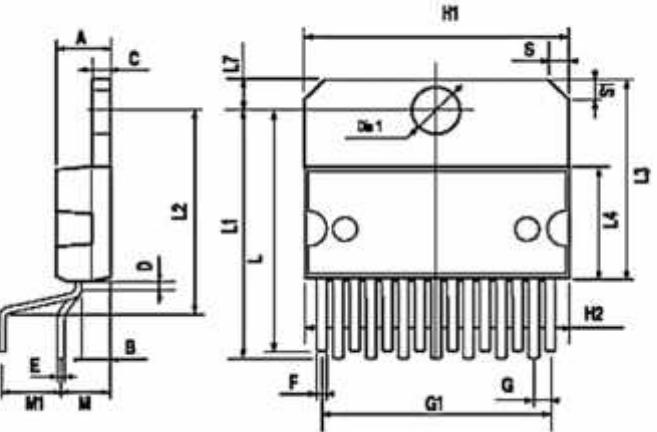


L298

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.5			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Diag	3.65		3.85	0.144		0.152

OUTLINE AND  
MECHANICAL DATA

Multiwatt15 V



## 2. *Omni Wheel*



### 3WD 48mm Omni Wheel Arduino Mobile Robot Kit 10019

This is unconventional mobile robot kit, it has 3 omni wheels that allow it to move and turn simultaneously, and accelerate in every direction at once without changing the direction it is facing.

Besides the 3 motors for driving, the robot has 3 Ultrasonic sensors to scan the environment. To program the mobile robot kit, we chose Arduino to maintain the software easily and to exploit the excellent motor control algorithms provided.



#### Features:

- 3 wheel drive
- Omni wheel
- Aluminum alloy fram
- Capable of rotation and side way
- Includes Ultrasonic sensors
- With Arduino microcontroller and IO expansion board
- Programmable with C, C++

#### Parts included

- 48mm Omni-wheel X 3
- Namiki DC Motor with encoder X 3
- Arduino 328 Controller
- Arduino IO Expansion
- Ultrasonic Range Finders X 3
- 12V NI-MH Battery X 1
- 12V Charger

#### Parts Description



Arduino IO Expansion is a sensor expansion board. The controller board is programmed from the PC in C, C++ programming language.



Arduino 328 is a microcontroller board. The USB port can be used for download of own programs and for transmitting data to the PC.

Nexus Automation Limited • [www.nexusrobot.com](http://www.nexusrobot.com)



The Ultrasonic Range Finder is the sensor that can give your robot a picture from his surroundings.



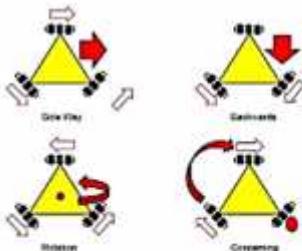
With the help of omni wheels, robot kit can move to all direction.

#### Specifications:

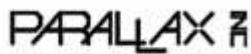
Product Name	48mm Omni wheel mobile robot kit	Speed	0.3m/s
Part Number	10019	Power	6w
Appearance	Circle	Drive Mode	3 Wheels Drive
Max Width	180mm	Color	Yellow
Height	120mm	Net Weight	1.5kg
Body Material	Aluminum Alloy	Load Capacity	1.5kg



#### Co-effect of Omni Wheels:



### 3. Ultrasonik PING)))



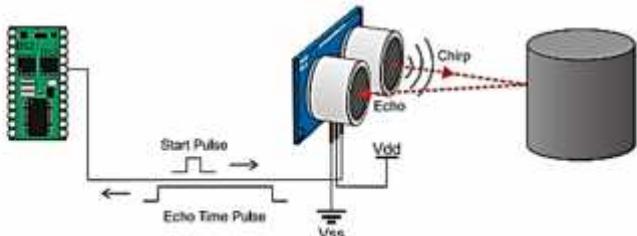
Web Site: [www.parallax.com](http://www.parallax.com)  
 Forums: [forums.parallax.com](http://forums.parallax.com)  
 Sales: [sales@parallax.com](mailto:sales@parallax.com)  
 Technical: [support@parallax.com](mailto:support@parallax.com)

Office: (916) 624-8333  
 Fax: (916) 624-8903  
 Sales: (888) 512-1024  
 Tech Support: (888) 997-8277

#### PING))) Ultrasonic Distance Sensor (#28015)

The Parallax PING)))™ ultrasonic distance sensor provides precise, non-contact distance measurements from about 2 cm (0.8 inches) to 3 meters (3.3 yards). It is very easy to connect to microcontrollers such as the BASIC Stamp®, Propeller chip, or Arduino, requiring only one I/O pin.

The PING))) sensor works by transmitting an ultrasonic (well above human hearing range) burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width, the distance to target can easily be calculated.



#### Features

- Range: 2 cm to 3 m (0.8 in to 3.3 yd)
- Burst indicator LED shows sensor activity
- Bidirectional TTL pulse interface on a single I/O pin can communicate with 5 V TTL or 3.3 V CMOS microcontrollers
- Input trigger: positive TTL pulse, 2 µs min, 5 µs typ.
- Echo pulse: positive TTL pulse, 115 µs minimum to 18.5 ms maximum.
- RoHS Compliant

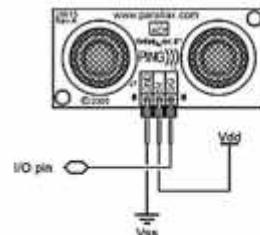
#### Key Specifications

- Supply voltage: +5 VDC
- Supply current: 30 mA typ; 35 mA max
- Communication: Positive TTL pulse
- Package: 3-pin SIP, 0.1" spacing (ground, power, signal)
- Operating temperature: 0 – 70° C.
- Size: 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)
- Weight: 9 g (0.32 oz)

#### Pin Definitions

GND	Ground (Vss)
5 V	5 VDC (Vdd)
SIG	Signal (I/O pin)

The PING))) sensor has a male 3-pin header used to supply ground, power (+5 VDC) and signal. The header may be plugged into a directly into solderless breadboard, or into a standard 3-wire extension cable (Parallax part #800-00120).

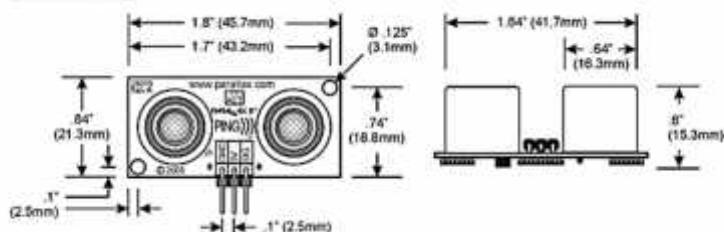


Copyright © Parallax Inc.

PING))) Ultrasonic Distance Sensor (#28015)

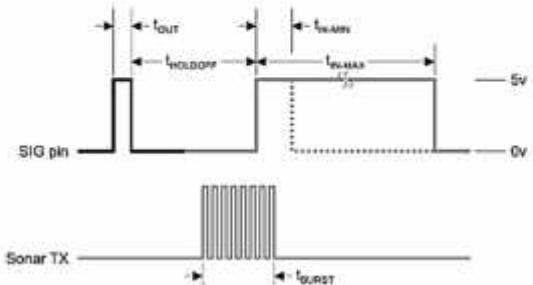
v2.0 2/4/2013 Page 1 of 9

#### Dimensions



## Communication Protocol

The PING))) sensor detects objects by emitting a short ultrasonic burst and then "listening" for the echo. Under control of a host microcontroller (trigger pulse), the sensor emits a short 40 kHz (ultrasonic) burst. This burst travels through the air, hits an object and then bounces back to the sensor. The PING))) sensor provides an output pulse to the host that will terminate when the echo is detected, hence the width of this pulse corresponds to the distance to the target.



	Host Device	Input Trigger Pulse	$t_{\text{TRIG}}$	2 $\mu\text{s}$ (min), 5 $\mu\text{s}$ typical
	PING))) Sensor	Echo Holdoff	$t_{\text{HOLDOFF}}$	750 $\mu\text{s}$
		Burst Frequency	$f_{\text{BURST}}$	200 $\mu\text{s}$ @ 40 kHz
		Echo Return Pulse Minimum	$t_{\text{IN-MIN}}$	115 $\mu\text{s}$
		Echo Return Pulse Maximum	$t_{\text{IN-MAX}}$	18.5 ms
		Delay before next measurement		200 $\mu\text{s}$

Copyright © Parallax Inc.

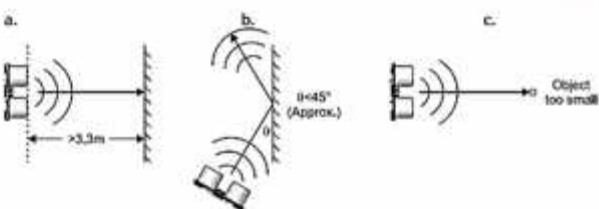
PING))) Ultrasonic Distance Sensor (#28015)

v2.0 2/4/2013 Page 2 of 9

## Practical Considerations for Use

### Object Positioning

The PING))) sensor cannot accurately measure the distance to an object that: a) is more than 3 meters away, b) that has its reflective surface at a shallow angle so that sound will not be reflected back towards the sensor, or c) is too small to reflect enough sound back to the sensor. In addition, if your PING))) sensor is mounted low on your device, you may detect sound reflecting off of the floor.



### Target Object Material

In addition, objects that absorb sound or have a soft or irregular surface, such as a stuffed animal, may not reflect enough sound to be detected accurately. The PING))) sensor will detect the surface of water, however it is not rated for outdoor use or continual use in a wet environment. Condensation on its transducers may affect performance and lifespan of the device.

### Air Temperature

Temperature has an effect on the speed of sound in air that is measurable by the PING))) sensor. If the temperature ( $^{\circ}\text{C}$ ) is known, the formula is:

$$C_{\text{air}} = 331.5 + [0.6 \times T_c] \text{ m/s}$$

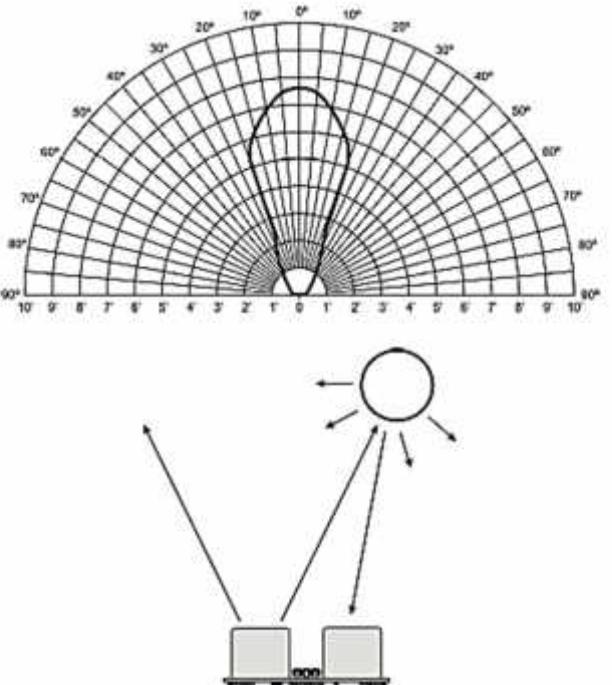
The percent error over the sensor's operating range of 0 to 70  $^{\circ}\text{C}$  is significant, in the magnitude of 11 to 12 percent. The use of conversion constants to account for air temperature may be incorporated into your program (as is the case in the example BS2 program given in the Example Programs section below). Percent error and conversion constant calculations are introduced in Chapter 2 of *Smart Sensors and Applications*, a Stamps in Class text available for download from the 28029 product page at [www.parallax.com](http://www.parallax.com).

### Test Data

The test data on the following pages is based on the PING))) sensor, tested in the Parallax lab, while connected to a BASIC Stamp microcontroller module. The test surface was a linoleum floor, so the sensor was elevated to minimize floor reflections in the data. All tests were conducted at room temperature, indoors, in a protected environment. The target was always centered at the same elevation as the PING))) sensor.

#### Test 1

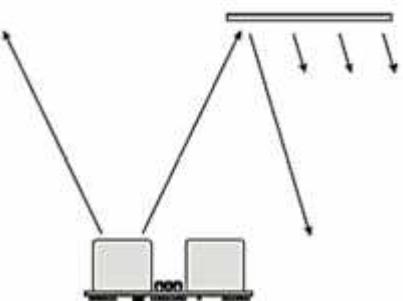
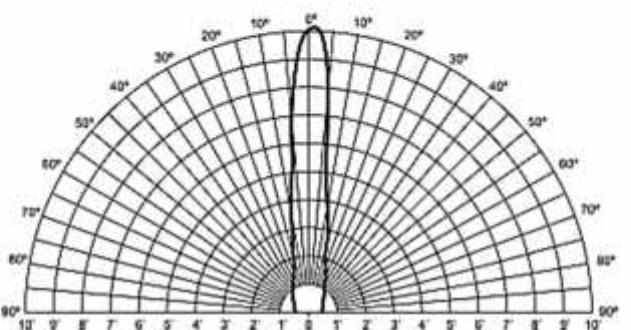
Sensor Elevation: 40 in. (101.6 cm)  
Target: 3.5 in. (8.9 cm) diameter cylinder, 4 ft. (121.9 cm) tall – vertical orientation



**Test 2**

Sensor Elevation:  
Target:

40 in. (101.6 cm)  
12 in. x 12 in. (30.5 cm x 30.5 cm) cardboard, mounted on 1 in. (2.5 cm) pole  
Target positioned parallel to backplane of sensor



#### 4. Arduino Mega 2560



##### Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Due and Arduino Uno.

##### Index

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



**radiospares**

**RADIONICS**



## Technical Specification

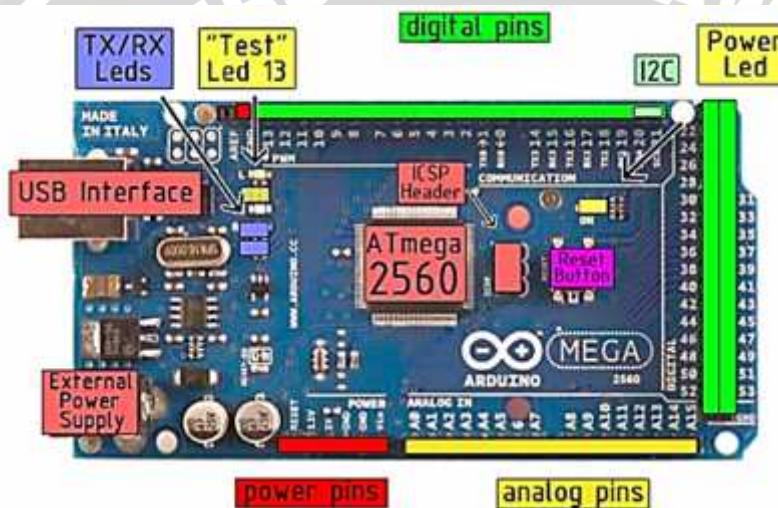


EAGLE files: [arduino-mega2560-reference-design.zip](#), Schematic: [arduino-mega2560-schematic.pdf](#)

### Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



Radiospares RADIONICS



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
  - **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
  - **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
  - **GND.** Ground pins.

## Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX); **Serial 1:** 19 (RX) and 18 (TX); **Serial 2:** 17 (RX) and 16 (TX); **Serial 3:** 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
  - **External Interrupts:** 2 (**interrupt 0**), 3 (**interrupt 1**), 18 (**interrupt 5**), 19 (**interrupt 4**), 20 (**interrupt 3**), and 21 (**interrupt 2**). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
  - **PWM:** 6 to 13. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
  - **SPI:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
  - **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
  - **I<sub>C</sub>:** 20 (SDA) and 21 (SCL). Support I<sub>C</sub> (TWI) communication using the [Wire library](#) (documentation on the [Wire website](#)). Note that these pins are not in the same location as the I<sub>C</sub> pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and `analogReference()` function.

There are a couple of other pins on the board:

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



## Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

**► SoftwareSerial library** allows for serial communication on any of the Mega's digital pins.

The ATmega2560 also supports 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 on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

## Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

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



**Radiospares**

**RADIONICS**



## Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmegaBU2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

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

## Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. Please note that I<sup>2</sup>C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).

## Dimensioned Drawing

