



**RANCANG BANGUN SISTEM NAVIGASI ROBOT BERODA PEMANDU  
ORANG DENGAN PENYANDANG TUNA NETRA MENGGUNAKAN  
METODE WAYPOINT.**

**Skripsi**

Oleh

**Ahmad Rausan Fikri**

**NIM 141910201063**

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2019**



**RANCANG BANGUN SISTEM NAVIGASI ROBOT BERODA PEMANDU  
ORANG DENGAN PENYANDANG TUNA NETRA MENGGUNAKAN  
METODE WAYPOINT.**

**Skripsi**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat  
untuk menyelesaikan Program Studi Elektro (S1)  
dan mencapai gelar Sarjana Teknik

Oleh

**Ahmad Rausan Fikri**

**NIM 141910201063**

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2019**

## PERSEMBAHAN

Alhamdulillah, puji syukur kehadirat Allah SWT. Yang selalu memberikan nikmat rizki yang tiada banding untuk terus berupaya dalam menjalankan perjuangan di dunia. Shalawat serta salam tetap tucurahkan kepada baginda besar Nabi Muhammad SAW. Dengan perjuangan beliau islam bisa sampai dalam hati sanubari hambah. Dan selalu kuharap syafaatmu di *yaumul qiyamah*. Dengan sadar dan ikhlas kupersembahkan laporan skripsi ini kepada orang-orang yang ikut serta dalam membimbing dan membentuk pribadiku hingga sampai di jalan yang penuh dengan pencerahan ilmu pengetahuan ini.

### *Yang Pertama*

Bapak dan Ibuk tersayang yang telah mencurahkan seluruh kasih dan sayangnya dalam berbagai bentuk pengajaran kehidupan sejak masih dalam kandungan hingga kini tanpa rasa lelah. Perjuangan lahir dan batin di sepanjang siang dan malam mereka yang mampu membuka ridha ilahi aku mengucapkan Terima kasih yang dalam serta tulus ikhlas.

### *Yang Kedua*

Ketiga adik-adikku yang selalu menjadi motivasi tersendiri supaya aku mampu menjadi suri tauladan dalam mengarungi pengabdian di dunia ini.

### *Yang Ketiga*

Semua dosen-dosen dan civitas akademika yang telah memberikan bimbingan dan ilmu yang tidak bisa terhitung berapa banyaknya jasa dan barakahnya.

### *Yang Terakhir*

Semua saudara seperjuangan KETEK UJ 14 dan khususnya seluruh anggota keluarga besar TEKNOKES yang selalu memberikan seruan moral untuk terus maju melewati rintangan yang ada.

**MOTTO**

*“Dan aku tidak menciptakan jin dan manusia melainkan supaya mereka mengabdikan kepada-Ku”  
(QS. Adz Dzariyat : 56)*

*“Sebaik-baik manusia adalah manusia yang bermanfaat bagi manusia lainnya”  
(Muhammad SAW.)*

*“Sesungguhnya Allah berfirman: "Aku sebagaimana prasangka hambaku kepada-Ku. Aku bersamanya ketika dia mengingat-Ku”  
(HR. Bukhori Muslim)*

*“Bukan mencoba untuk menjadi orang yang sukses, tapi lebih pada menjadi orang yang memiliki nilai”  
(Albert Einstein)*

*“Siapa dapat ramalkan bagaimana bakal jadinya bayi? Jadi nabi atau bajingan, atau sekedar jadi tambahan isi dunia, polos, tanpa apa-apa”  
(Minke, hal. 296 – Jejak Langkah)*

*“Tidak ada hal yang lebih indah di dunia ini selain ikhlas berbagi”  
(Penulis)*

**PERNYATAAN**

Saya yang bertanda tangan dibawah ini:

Nama : Ahmad Rausan Fikri

NIM : 141910201063

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Rancang Bangun Sistem Navigasi Robot Beroda Pemandu Orang Dengan Penyandang Tuna Netra Menggunakan Metode *Waypoint*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun dan bukan karya jiplakan. Saya bertanggung jawab penuh atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 26 Desember 2019

Yang menyatakan,

Ahmad Rausan Fikri

NIM 141910201063

**SKRIPSI**

**RANCANG BANGUN SISTEM NAVIGASI ROBOT BERODA PEMANDU  
ORANG DENGAN PENYANDANG TUNA NETRA MENGGUNAKAN  
METODE *WAYPOINT*.**

Oleh

Ahmad Rausan Fikri

NIM 141910201063

Pembimbing :

Dosen Pembimbing Utama : Khairul Anam, S.T., M.T., Ph.D

Dosen Pembimbing Anggota : Widya Cahyadi, S.T., M.T.

**PENGESAHAN**

Skripsi berjudul “**Rancang Bangun Sistem Navigasi Robot Beroda Pemandu Orang Dengan Penyandang Tuna Netra Menggunakan Metode Waypoint**”

karya Ahmad Rausan Fikri telah diuji dan disahkan pada :

Hari : Kamis

Tanggal : 26 Desember 2019

Tempat : Fakultas Teknik Universitas Jember

Tim penguji,

Ketua,

Sekretaris,

**Khairul Anam, S.T., M.T., Ph.D**

NIP 197804052005011002

**Widva Cahyadi, S.T., M.T.**

NIP 198511102014041001

Anggota I,

Anggota II,

**Sumardi, S.T., M.T.**

NIP 196701131998021001

**Dodi Setiabudi S.T., M.T.**

NIP 198405312008121004

Mengesahkan  
Dekan,

**Dr. Ir. Entin Hidayah, M.U.M.**

NIP 196612151995032001

Rancang Bangun Sistem Navigasi Robot Beroda Pemandu Orang Dengan  
Penyanggah Tuna Netra Menggunakan Metode *Waypoint*

**AHMAD RAUSAN FIKRI**

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

**ABSTRAK**

*Di zaman modern perkembangan teknologi semakin hari semakin pesat. Berawal dari tujuannya yaitu untuk membantu serta memudahkan manusia dalam mengerjakan tugasnya hingga kini menjadi sebuah ketergantungan. Salah satu bidang teknologi yang sangat membantu kehidupan manusia dan juga perkembangan teknologi itu sendiri adalah bidang robotika. Melihat penderita disabilitas di Indonesia yang tidak sedikit, di tahun 2016 penderita disabilitas di Indonesia sebesar 12,15 persen (BPS, 2016). Sehingga penulis melakukan penelitian terkait system navigasi robot beroda yang melakukan proses navigasi pada lingkungan terbuka atau outdoor dengan lintasan berhalang dan berlubang sebagai pemandu orang dengan penyandang tuna netra menggunakan sistem navigasi berbasis Global Positioning System dengan metode waypoint.*

**Kata kunci:** *Robot navigation , GPS, Waypoint, Disabilities.*



*(Design Of Mobile Robot Navigation System for People's Guide with the Visually Impaired Using Waypoint Method)*

**AHMAD RAUSAN FIKRI**

*Department of Electrical Engineering, Faculty of Engineering, University of Jember*

## **ABSTRACT**

*In modern times the development of technology is increasingly rapid. Starting from the goal that is to help and facilitate humans in carrying out their duties until now it has become a dependency. One area of technology that really helps human life and also the development of technology itself is the field of robotics. Looking at the number of people with disabilities in Indonesia, in 2016 there were 12.15 percent of people with disabilities in Indonesia (BPS, 2016). So the authors conduct research related to wheeled robot navigation systems that do the navigation process in an open or outdoor environment with obstructed and hollow paths as a guide for people with blind people using a Global Positioning System based navigation system with the waypoint method.*

**Keywords:** *Robot navigation , GPS, Waypoint, Disabilities.*

## RINGKASAN

**Rancang Bangun Sistem Navigasi Robot Beroda Pemandu Orang Dengan Penyandang Tuna Netra Menggunakan Metode *Waypoint***; Ahmad Rausan Fikri; 141910201063; 2019; 49 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Di zaman modern perkembangan teknologi semakin hari semakin pesat. Berawal dari tujuannya yaitu untuk membantu serta memudahkan manusia dalam mengerjakan tugasnya hingga kini menjadi sebuah ketergantungan. Sehingga bisa penulis asumsikan bahwa teknologi menjadi sangat penting bagi berjalannya kehidupan manusia bahkan untuk keberlangsungan manusia itu sendiri. Salah satu bidang teknologi yang sangat membantu kehidupan manusia dan juga perkembangan teknologi itu sendiri adalah bidang robotika. *Mobile robot* menjadi jenis robot yang paling sering dijadikan bahan penelitian. Penelitian robot yang ditujukan untuk membantu orang yang mengalami disabilitas menjadi salah satu objek penelitian yang menarik. Melihat penderita disabilitas di Indonesia yang tidak sedikit, di tahun 2016 penderita disabilitas di Indonesia sebesar 12,15 persen (BPS, 2016) hal ini menjadi salah satu faktor dari banyaknya peneliti yang memilih robot pembantu penyandang disabilitas. Dari latar belakang diatas dalam skripsi ini penulis memilih robot beroda pemandu orang dengan penyandang tuna netra sebagai objek penelitian.

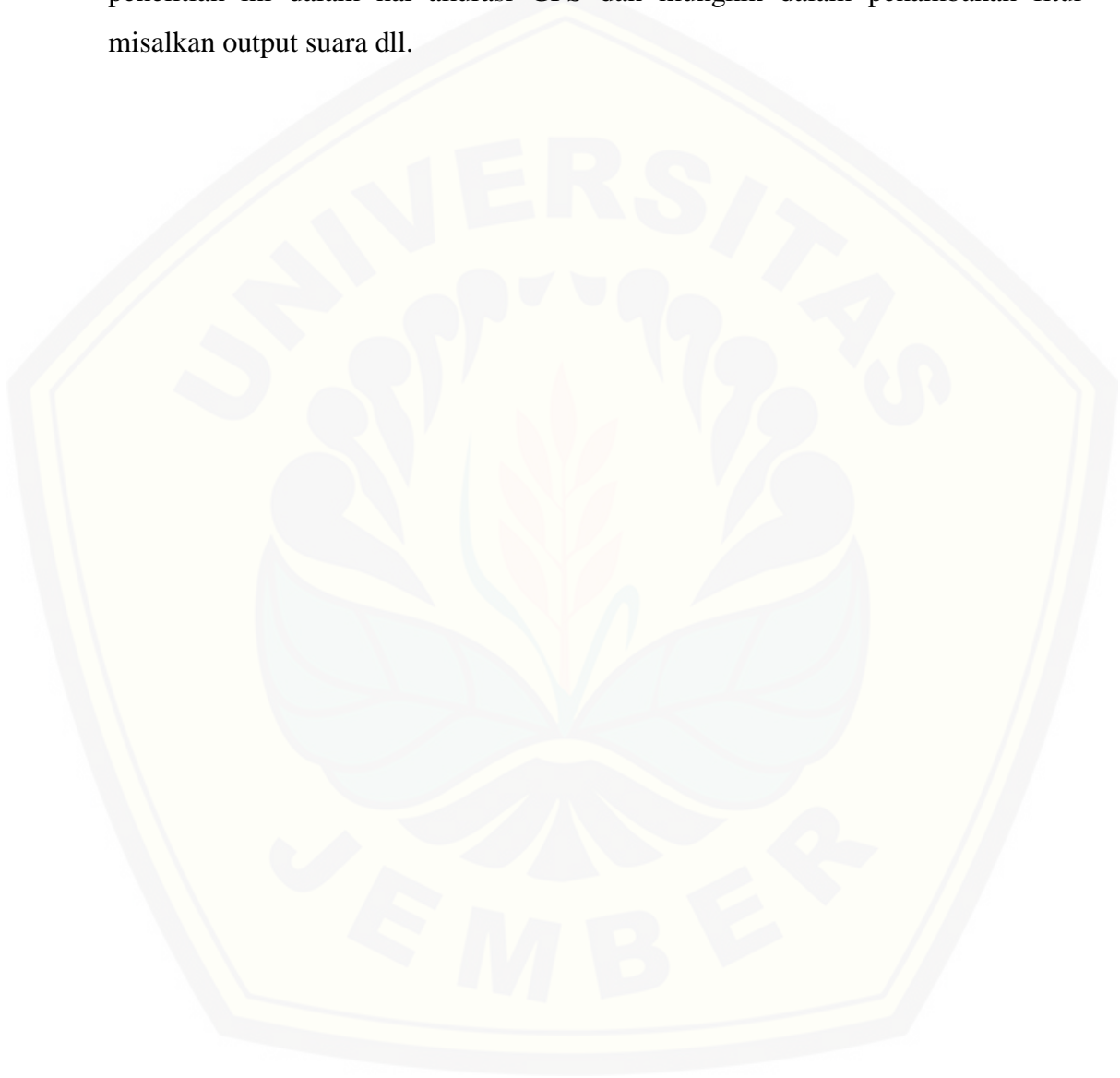
Sebenarnya dalam dunia internasional, robot yang dibuat untuk membantu orang penyandang tuna netra bukanlah objek penelitian yang baru seperti disebutkan dalam jurnal Marion A. Hersh yang berjudul "*A robotic guide for blind people. part 1. A multi nasional survey of the attitudes, requirements and preferences of potential end-users*" pembuatan robot pemandu orang dengan penyandang tuna netra sudah dimulai dengan MELDOG proyek sejak tahun 1977. Sehingga dari uraian di atas pada penelitian kali ini penulis mencoba mengambil

beberapa aspek yang akan penulis kembangkan yaitu dalam hal otomasi robot dalam berjalan menentukan tujuan serta menghindari halangan dan lubang dengan menggunakan *Global Positioning System* dengan metode *Waypoint*. Manfaat penelitian adalah dapat membuat suatu system navigasi robot beroda yang melakukan proses navigasi pada lingkungan terbuka atau *outdoor* dengan lintasan berhalang dan berlubang sebagai pemandu orang dengan penyandang tuna netra menggunakan sistem navigasi berbasis *Global Positioning System* dengan metode *waypoint*.

Dalam pembuatan rancang bangun robot pemandu penyandang tuna netra ini penulis menggunakan bahan dasar akrilik sebagai kerangka dasar robot dengan bentuk bundar supaya lebih muda dalam melakukan sebuah pergerakan. Kemudian dalam hal mikokontroler penulis menggunakan perangkat Linkit One yang sudah memiliki *Global Positioning System* yang sudah terhubung langsung, serta beberapa komponen sensor seperti ultrasonic, kompas, odometer dan komponen dasar robot lainnya berupa motor, batrei dan driver motor.

Setelah selesai dalam pembuatan mekanik, elektrikal dan kemudian penulis rangkai semua hingga robot sudah siap untuk deprogram dan melanjutkan pada proses pengambilan data. Proses pengambilan data dilakukan dengan melakukan kalibrasi untuk mengetahui akurasi dari masing-masing sensor, kemudian dilanjutkan dengan pengujian beberapa system seperti system menghindari halangan dan lubang, system Bearing target, system odometer dan yang terakhir pengujian keseluruhan yaitu metode *waypoint* dengan tanpa halangan dan dengan adanya halangan. Setelah melakukan pengujian kami mendapatkan hasil di mana beberapa system bisa berjalan dengan akurasi yang cukup baik seperti system menghindari halangan dan lubang dan system odometer serta ada beberapa system yang berjalan dengan kurang baik diantaranya system Bearing target dan system GPS di mana robot memiliki eror yang relatif besar dalam mencari titik koordinat target hingga lebih dari 4 meter ketika robot berpindah-pindah tempat. Namun dalam pengujian ternyata kekurangan GPS tersebut bisa diminimalisir dengan akurasi yang dimiliki oleh odometer dalam menghitung jarak tempuh robot.

Sehingga dalam penelitian ini masih memiliki banyak kekurangan yaitu dalam menentukan koordinat robot masih memiliki akurasi yang kurang baik sehingga akan memungkinkan robot untuk berjalan pada track yang tidak berada pada jalan yang sesungguhnya. Harapannya akan ada pengembangan dalam penelitian ini dalam hal akurasi GPS dan mungkin dalam penambahan fitur misalkan output suara dll.



## PRAKATA

Puji syukur kehadirat Allah SWT yang maha kuasa atas segalanya, karena dengan ridho, hidayah dan petunjukNya, penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Sistem Navigasi Robot Beroda Pemandu Orang Dengan Penyandang Tuna Netra Menggunakan Metode *Waypoint*”. Selama penyusunan skripsi ini penulis mendapat bantuan berbagai pihak yang turut memberikan bantuan berupa motivasi, inspirasi, bimbingan, doa, fasilitas dan dukungan lainnya yang membantu memperlancar pengerjaan skripsi ini.

Untuk itu penulis mengucapkan terimakasih kepada.

1. Allah SWT yang telah melimpahkan rezeki, rahmat, hidayah dan karunia serta kasih sayang-Nya sehingga penulis mampu menyelesaikan skripsi ini.
2. Nabi besar Muhammad SAW, yang telah menjadi suri tauladan bagi seluruh umat.
3. Ibu Dr. Ir. Entin Hidayah, M.U.M., selaku Dekan Fakultas Teknik Universitas Jember
4. Bapak Dr. Bambang Sri Kaloko, S.T., M.T., Selaku Ketua Jurusan Teknik Elektro Universitas Jember;
5. Bapak Khairul Anam, S.T., M.T., Ph.D dan Bapak Widya Cahyadi, S.T., M.T. selaku dosen pembimbing yang telah membimbing menyelesaikan tugas akhir ini;
6. Bapak Bapak Sumardi, S.T., M.T. dan Bapak Dodi Setiabudi S.T., M.T. selaku dosen penguji yang sudah memberikan saran untuk memperbaiki tugas akhir ini;
7. Bapak Majlis Noer dan Ibuk Siti Badriyah yang sejak masa kecil penulis menjadi sosok motivator dalam keilmuan, serta perjuangan dalam mendidik, menafkahi dan selalu memberi doa dan support moral yang terukur besarnya;
8. Ketiga adek yang selalu menjadi pengingat untuk terus semangat dalam menjadi sosok panutan dan referensi dalam menjadi seorang anak;
9. Sahabat-sahabat seperjuangan Teknik Elektro 2014 yang telah memberikan motivasi dan semangat dalam perjuangan di bangku kuliah khususnya

Elektronika angkatan 2014 yang selalu memberikan bantuan dan support dalam mengerjakan tugas akhir ini;

10. Ucapan terimakasih saya sampaikan kepada seluruh Sahabat-sahabati PMII di Jember, Komisariat Universitas Jember dan terkhusus keluarga besar Teknokes yang sudah memberikan ruang proses dialektis yang memberikan penulis pemahaman akan arti menjadi manusia.
11. Semua pihak yang tidak dapat disebutkan satu persatu.

Semoga skripsi ini dapat bermanfaat dalam mengembangkan ilmu pengetahuan khususnya untuk disiplin ilmu teknik elektro. Kritik dan saran yang mambangun diharapkan terus mengalir untuk lebih menyempurnakan skripsi ini dan dapat dikembangkan untuk penelitian selanjutnya;

Jember, 26 Desember 2019

Penulis

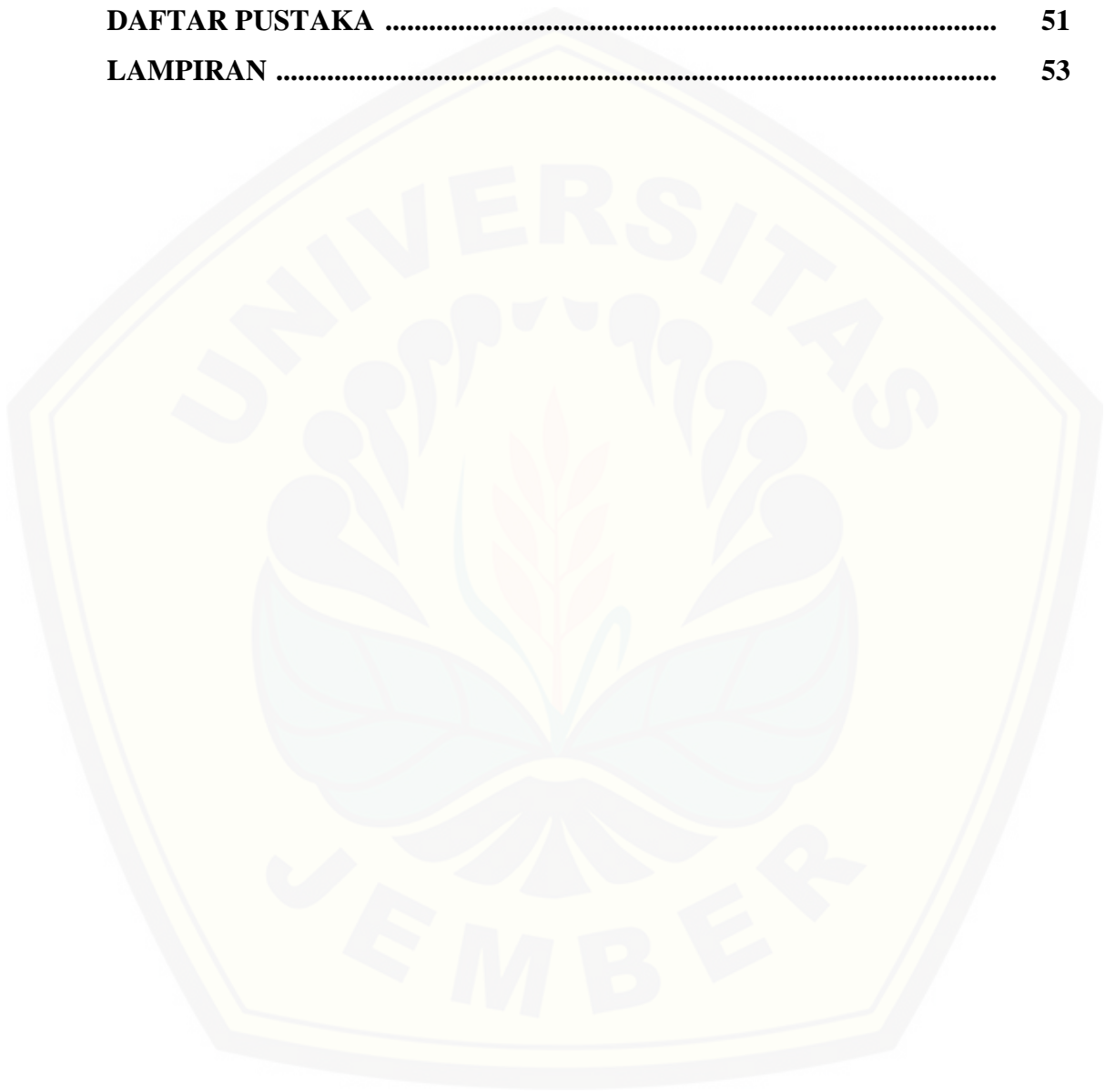
DAFTAR ISI

HALAMAN JUDUL .....	i
PERSEMBAHAN .....	ii
MOTTO .....	iii
PERNYATAN.....	iv
HALAMAN PEMBIMBING.....	v
PENGESAHAN .....	vi
ABSTRAK .....	vii
RINGKASAN .....	ix
PRAKATA .....	xii
DAFTAR ISI.....	xiv
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xix
DAFTAR LAMPIRAN .....	xx
<b>BAB 1. PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian .....	3
<b>BAB 2. TINJAUAN PUSTAKA .....</b>	<b>5</b>
2.1 Penelitian Terkait .....	5
2.2 <i>Global Positioning System</i> .....	7
2.3 <i>Metode Waypoint</i> .....	8
2.4 Linkit ONE .....	11
2.5 <i>Magnetic Compas HMC5883L</i> .....	13
2.6 LCD 16 x 2 .....	14
2.7 Driver Motor .....	15
2.8 Motor DC.....	16
2.9 <i>Sensor Rotary Encoder</i> .....	18

<b>BAB 3. METODOLOGI PENELITIAN .....</b>	<b>19</b>
<b>3.1 Waktu dan Tempat Kegiatan .....</b>	<b>19</b>
<b>3.2 Ruang Lingkup Kegiatan .....</b>	<b>20</b>
<b>3.3 Jenis dan Sumber Data .....</b>	<b>20</b>
<b>3.4 Metode Pengumpulan Data.....</b>	<b>21</b>
<b>3.5 Perancangan Alat .....</b>	<b>21</b>
3.5.1 Perancangan Desain Alat .....	21
3.5.2 Perancangan Perangkat Keras.....	22
3.5.2.1 Blok Diagram.....	23
3.5.2.2 Rangkaian Sensor Kompas dengan Linkit ONE.....	24
3.5.2.3 Pemasangan <i>Rotary Encoder</i> atau <i>Optocoupler</i> Pada Linkit ONE.....	24
3.5.2.4 Rangkaian <i>Driver Motor</i> dengan Linkit ONE .....	25
3.5.2.5 Pemasangan sensor <i>Ultrasonic</i> Pada Linkit ONE ..	26
3.5.2.6 Pemasangan LCD 16 x 2 Pada Linkit ONE.....	26
3.5.3 Perancangan Sistem Kontrol.....	27
3.5.4 Perancangan Perangkat Lunak.....	28
3.5.4.1 <i>Flowchart</i> Jalannya Robot Secara Keseluruhan .....	28
3.5.4.2 <i>Flowchart</i> Sensor Ultrasonik .....	30
<b>BAB 4. HASIL DAN PEMBAHASAN .....</b>	<b>31</b>
<b>4.1 Sensor Jarak HCSR-04 .....</b>	<b>31</b>
4.1.1 Kalibrasi Sensor Jarak HCSR-04 .....	31
4.1.2 Pengujian Sensor Jarak HCSR-04 .....	33
<b>4.2 Sensor <i>Rotary Encoder</i> .....</b>	<b>35</b>
4.2.1 Kalibrasi Sensor <i>Rotary Encoder</i> .....	35
4.2.2 Pengujian Sensor <i>Rotary Encoder</i> .....	36
<b>4.3 Sensor Kompas .....</b>	<b>38</b>
4.3.1 Kalibrasi Sensor Kompas.....	38
4.3.2 Pengujian Sensor Kompas .....	39
<b>4.4 Pengujian GPS Linkit ONE .....</b>	<b>41</b>



4.5 Pengujian Waypoint .....	43
<b>BAB 5. KESIMPULAN DAN SARAN .....</b>	<b>48</b>
5.1 Kesimpulan .....	48
5.2 Saran .....	50
<b>DAFTAR PUSTAKA .....</b>	<b>51</b>
<b>LAMPIRAN .....</b>	<b>53</b>



DAFTAR GAMBAR

Gambar 2.1 Robot Pemandu Orang Dengan Penyandang Tuna Netra. ....	6
Gambar 2.2 Perancangan Bentuk dan Pemasangan Sensor .....	6
Gambar 2.3 Segmen Angkasa GPS.....	8
Gambar 2.4 Bentuk Fisik Linkit ONE .....	11
Gambar 2.5 Sensor <i>Magnetic Compas</i> HMC5883L .....	13
Gambar 2.6 LCD .....	14
Gambar 2.7 I2C LCD .....	15
Gambar 2.8 Driver motor DC dengan L298 .....	16
Gambar 2.9 Motor DC <i>Power Window</i> .....	17
Gambar 2.10 <i>Rotary Encoder</i> . ....	18
Gambar 3.1 Perancangan Desain Alat .....	22
Gambar 3.2 Alat Tampak Samping .....	22
Gambar 3.3 Blok Diagram .....	23
Gambar 3.4 Rangkaian Sensor kompas dengan Linkit ONE. ....	24
Gambar 3.5 Rangkaian <i>Rotary Encoder</i> dengan Linkit ONE.....	25
Gambar 3.6 Rangkaian <i>Driver Motor</i> dengan Linkit ONE.....	25
Gambar 3.7 Rangkaian sensor <i>Ultrasonic</i> dengan Linkit ONE.....	26
Gambar 3.8 Rangkaian LCD 16 x 2 I2C dengan Linkit ONE .....	27
Gambar 3.9 Bagan Sistem Kontrol Robot.....	27
Gambar 3.10 Diagram Alir Perangkat Lunak Sistem Navigasi <i>Waypoint</i> .....	28
Gambar 3.11 Diagram Alir Perangkat Lunak Sub Fungsi <i>Waypoint</i> .....	29
Gambar 3.12 Diagram Alir Sensor Halangan .....	30
Gambar 3.13 Diagram Alir Sensor Berlubang.....	30
Gambar 4.1 Grafik kalibrasi sensor HCSR-04.....	33
Gambar 4.2 Pengujian Sensor Jarak Untuk Halang-Rintang .....	34
Gambar 4.3 Pengujian Sensor Jarak Untuk Deteksi Lubang .....	35
Gambar 4.4 Grafik kalibrasi sensor <i>rotary encoder</i> .....	36
Gambar 4.5 Pengujian Sensor <i>Rotary Encoder</i> Untuk Mengukur Jarak	

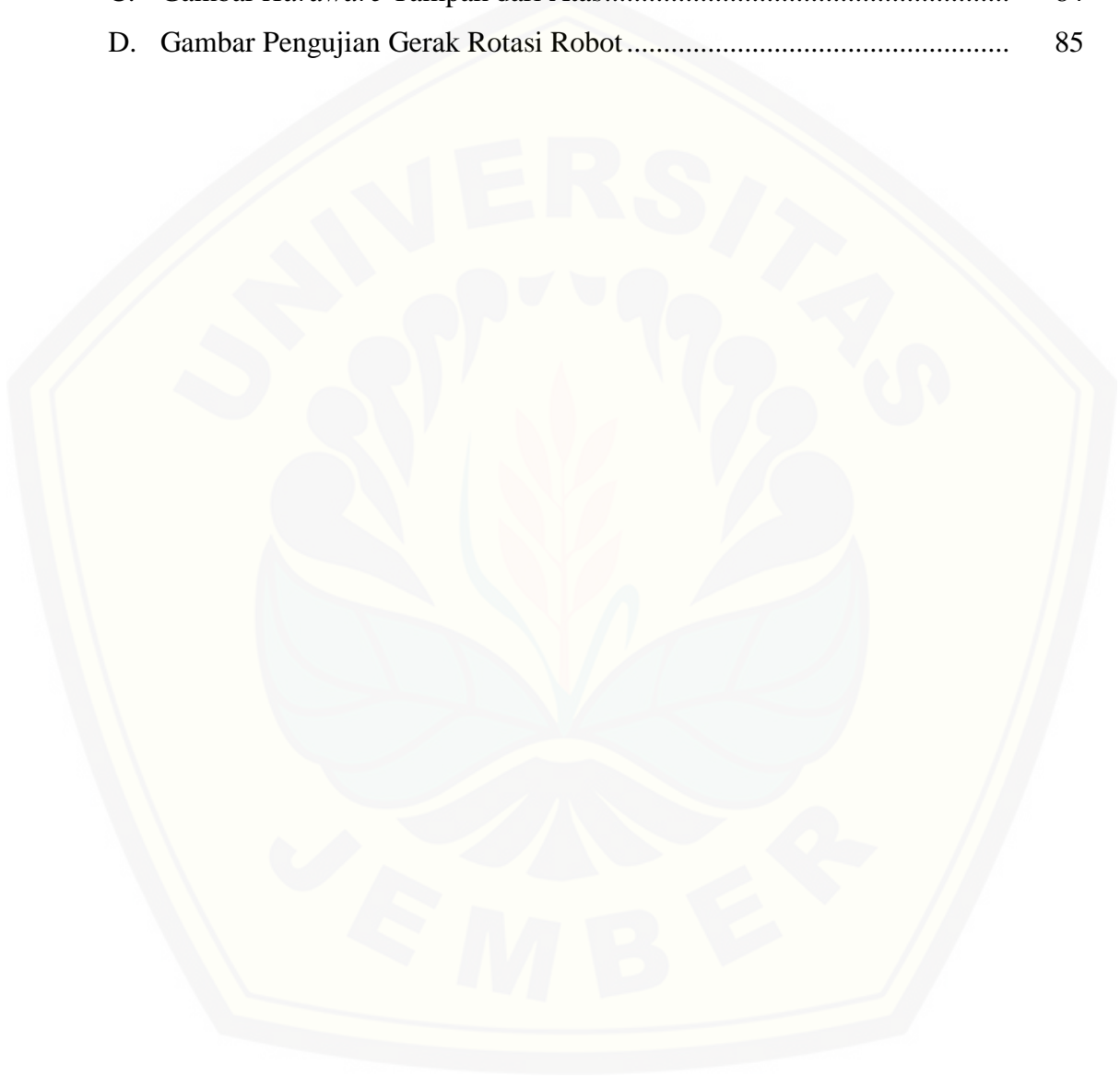
Tempuh.....	37
Gambar 4.6 Grafik Kalibrasi Sensor <i>Kompas HMC5883L</i> .....	39
Gambar 4.7 Pengujian Sensor <i>Kompas HMC5883L</i> Ketika Selisih Bearing < -10 .....	40
Gambar 4.8 Pengujian Sensor <i>Kompas HMC5883L</i> Ketika Selisih Bearing > 10 .....	40
Gambar 4.9 Perbandingan Data Posisi GPS Linkit ONE .....	42
Gambar 4.10 Pengambilan Latitude dan Longitude GPS Linkit ONE .....	42
Gambar 4.11 Pengujian <i>Waypoint</i> di gedung KAUJE Universitas Jember .....	44
Gambar 4.12 Pengujian <i>Waypoint</i> Saat Tanpa Halangan Atau Lubang .....	45
Gambar 4.13 Pengujian <i>Waypoint</i> Saat Ada Halangan Atau Lubang.....	46

**DAFTAR TABEL**

Tabel 3.1 Rencana Kegiatan Tugas Akhir .....	19
Tabel 4.1 Kalibrasi Sensor HCSR-04 (Cm).....	32
Tabel 4.2 Pengujian Sensor Jarak Untuk Halang-Rintang.....	33
Tabel 4.3 Pengujian Sensor Jarak Untuk Deteksi Lubang.....	34
Tabel 4.4 Kalibrasi Sensor <i>Rotary Encoder</i> .....	35
Tabel 4.5 Pengujian Sensor <i>Rotary Encoder</i> .....	37
Tabel 4.6 Kalibrasi Sensor Kompas HMC5883L .....	38
Tabel 4.7 Pengujian Gerak Rotasi Robot.....	41
Tabel 4.8 <i>Waypoint</i> Dan Jarak Tempuh Robot .....	43
Tabel 4.9 Error Lokasi Robot di Setiap <i>Waypoint</i> Saat Tanpa Ada Halangan Atau Lubang .....	44
Tabel 4.10 Error Lokasi Robot di Setiap <i>Waypoint</i> Saat Ada Halangan Atau Lubang.....	45

**DAFTAR LAMPIRAN**

A. <i>Listing</i> Program Linkit ONE.....	53
B. Gambar <i>Hardware</i> Tampak dari Samping.....	84
C. Gambar <i>Hardware</i> Tampak dari Atas.....	84
D. Gambar Pengujian Gerak Rotasi Robot.....	85



## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Di zaman modern perkembangan teknologi semakin hari semakin pesat. Berawal dari tujuannya yaitu untuk membantu serta memudahkan manusia dalam mengerjakan tugasnya hingga kini menjadi sebuah ketergantungan. Ketergantungan manusia terhadap teknologi yang kini semakin besar, dilihat dari aktivitas manusia mulai dari bangun tidur hingga tidur lagi selalu bersanding dengan teknologi. Sehingga bisa penulis asumsikan bahwa teknologi menjadi sangat penting bagi berjalannya kehidupan manusia bahkan untuk keberlangsungan manusia itu sendiri. Salah satu bidang teknologi yang sangat membantu kehidupan manusia dan juga perkembangan teknologi itu sendiri adalah bidang robotika. Bidang robotika memberikan dampak besar dalam dunia industri melalui sistem otomasi yang membuat pekerjaan pabrik bisa dikerjakan dengan bantuan robot. Tidak hanya itu robotika mulai menjadi disiplin ilmu yang banyak diperhatikan di negara-negara besar sehingga mulai bermunculan jenis-jenis robot lainnya seperti *mobile robot* yang sangat populer saat ini (Supriyanto, 2010).

*Mobile robot* menjadi jenis robot yang paling sering dijadikan bahan penelitian. Penelitian mengenai *mobile robot* pada dewasa ini sering kali ditujukan dalam pengembangan teknologi otomasi dalam transportasi, *platform* bergerak untuk robot industri, eksplorasi tanpa awak hingga sesuatu yang bersinggungan langsung dengan kehidupan manusia seperti membantu melakukan pekerjaan rumah tangga, hingga membantu mereka yang mengalami disabilitas. Penelitian robot yang ditujukan untuk membantu orang yang mengalami disabilitas menjadi salah satu objek penelitian yang menarik. Melihat penderita disabilitas di Indonesia yang tidak sedikit, di tahun 2016 penderita disabilitas di Indonesia sebesar 12,15 persen (BPS, 2016) hal ini menjadi salah satu faktor dari banyaknya peneliti yang memilih robot pembantu penyandang disabilitas.

Dari latar belakang di atas dalam skripsi ini penulis memilih robot beroda pemandu orang dengan penyandang tuna netra sebagai objek penelitian. Jika

penelitian tentang robot yang membantu penderita disabilitas di Indonesia sudah tidak sulit untuk ditemukan, lain halnya dengan robot pemandu orang dengan disabilitas tertentu yaitu penyandang tuna netra. Penelitian tentang robot pemandu orang dengan penyandang tuna netra di Indonesia masih susah untuk kita temukan, sehingga referensi yang bisa kita gunakan mengenai penelitian serupa ini akan lebih mudah jika kita dapatkan dari penelitian di luar negeri. Seperti penelitian yang dilakukan oleh Irwan Ulrich dan Johann Borenstein dalam jurnalnya “*The GuideCane-Applying Mobile Robot Technologies to Assist the Visually Impaired*” (2001) mereka membuat sebuah *mobile robot* pemandu orang dengan penyandang tuna netra dengan menggunakan sensor ultrasonik untuk menghindari halangan. Dalam penelitiannya, robot mereka masih menggunakan manual dalam memandu yaitu *joy stick*. *Mobile robot* yang mereka buat masih fokus pada menghindari halangan dan membuat roda kembali pada arah awal setelah halangan sudah dihindari serta mereka menggunakan *joy stick* dalam membantu mengontrol robotnya. Dalam paragraf akhir di paper mereka, mereka berharap penelitian terkait ini akan dilanjutkan dengan beberapa improvisasi di antaranya menggunakan *Sonar, Break, Wheel Conviguration, Speech Output, Global Navigation* dan *Computer Vision*.

Sebenarnya robot yang dibuat untuk membantu orang penyandang tuna netra bukanlah objek penelitian yang baru seperti disebutkan dalam jurnal Marion A. Hersh yang berjudul “*A robotic guide for blind people. part 1. A multi nasional survey of the attitudes, requirements adn preferences of potential end-users*” pembuatan robot pemandu orang dengan penyandang tuna netra sudah dimulai dengan MELDOG projek sejak tahun 1977. Berbeda dengan robot yang dibuat oleh Ulrich, dalam jurnalnya Vladimir Kulyukin yang berjudul “*Human-Robot Interaction in a Robotic Guide for the Visually Impaired*” dia menjelaskan robot RG (Robot Guide) yang telah dia buat untuk memandu orang dengan penyandang tuna netra sudah dilengkapi dengan User Interface dan komunikasi melalui mikrofon, dan sonar sensornya lebih banyak yaitu 16 buah dengan 8 di depan dan 8 di belakang. Sehingga dari uraian di atas pada penelitian kali ini penulis mencoba mengambil beberapa aspek yang akan penulis kembangkan yaitu dalam hal otomasi

robot dalam berjalan menentukan tujuan serta menghindari halangan dan lubang. Sehingga penulis mengambil judul “Rancang Bangun Sistem Navigasi Robot Beroda Pemandu Orang Dengan Penyandang Tuna Netra Menggunakan Metode *Waypoint*” dengan maksud untuk memberikan beberapa inovasi lanjutan dari penelitian-penelitian terkait menggunakan *Global Positioning System*. Di mana dalam penelitian di atas sebagai respon dari *Global Navigation* yaitu sebagai input dari navigasi *mobile robot* pemandu orang dengan penyandang tuna netra.

## 1.2 Rumusan Masalah

Dari latar belakang di atas maka kami mencari suatu pemecahan masalah yaitu bagaimana membuat rancang bangun robot mobile navigasi *outdoor* dengan menggunakan sistem navigasi berbasis *Global Positioning System* dengan metode *waypoint* yang bisa memandu orang dengan penyandang tuna netra di lingkungan terbuka. Dengan mengasumsikan beberapa kejadian yang mungkin ditemukan ketika robot mulai memandu di lingkungan terbuka seperti adanya sebuah halang rintang atau lubang.

## 1.3 Tujuan Penelitian

Adapun tujuan penelitian adalah sebagai berikut:

- a. Merancang bangun system navigasi robot beroda di lingkungan terbuka sebagai pemandu orang penyandang tuna netra pada lintasan berhalang dan berlubang.
- b. Merancang sistem navigasi pada robot beroda berbasis *Global Positioning System* dengan metode *Waypoint*.
- c. Menguji robot beroda pemandu orang dengan penyandang tuna netra di lingkungan terbuka yang sebenarnya.

## 1.4 Manfaat Penelitian

Manfaat penelitian adalah dapat membuat suatu system navigasi robot beroda yang dapat melakukan proses navigasi pada lingkungan terbuka atau *outdoor* dengan lintasan berhalang dan berlubang sebagai pemandu orang



penyandang tuna netra menggunakan sistem navigasi berbasis *Global Positioning System* dengan metode *waypoint*.



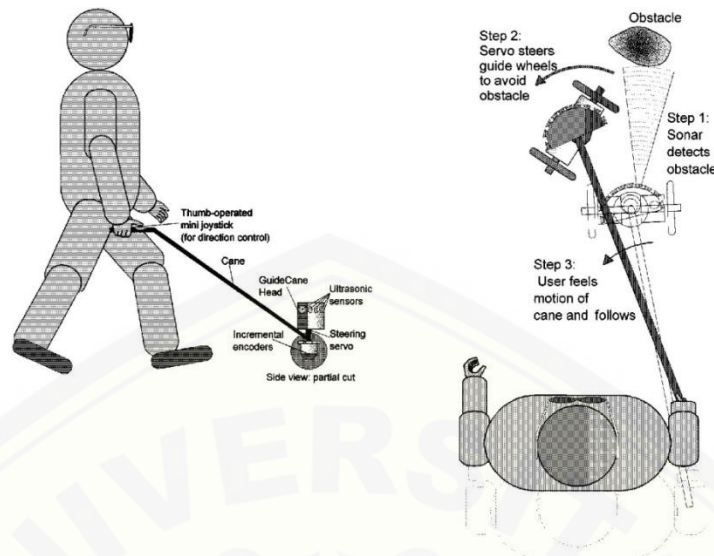
## BAB 2. TINJAUAN PUSTAKA

Pada bab ini dijelaskan materi tentang komponen yang digunakan beserta cara umum dari alat ini sendiri. Dan dijelaskan tentang pengumpulan pendapat atau teori yang telah ada yang berkaitan dengan masalah yang dibahas, membandingkan dan memilih teori yang paling relevan untuk memecahkan masalah, membahas atau menilai kelemahan dan keunggulan teori-teori, dan menentukan teori-teori sebagai dasar analisis selanjutnya, dan dapat dijelaskan di bawah ini antara lain:

### 2.1 Penelitian Terkait

- a. *The GuideCane—Applying Mobile Robot Technologies to Assist the Visually Impaired* (Tongkat Pandu-Penerapan teknologi robot beroda untuk membantu penderita gangguan Visual)

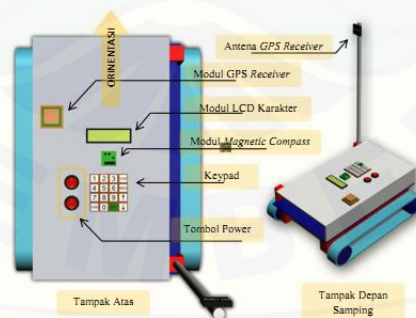
Penelitian terkait ini membahas tentang robot beroda yang dapat memandu penderita gangguan visual sebagai pengganti tongkat pandu. Robot tersebut menggunakan sonar sebagai sensor pendeteksi adanya halangan. Robot tersebut akan belok dengan bantuan servo ketika ada halangan yang terdeteksi dan roda kembali ke arah sebelumnya secara otomatis sesuai program yang sudah digunakan pada mikrokontrolernya. Robot tersebut masih menggunakan mikrokontroler MC86HC11 dan dilengkapi dengan 10 sensor ultrasonik, encoder, potensiometer, dua aktuator yaitu servo dan rem serta mini *joystick* sebagai bantu pengontrol langsung. Sehingga masih memiliki kekurangan terutama dalam hal kendali yang jika kita asumsikan akan membutuhkan orang lain untuk mengendalikan robot tersebut dengan menggunakan manual *joystick* (Ulrich dan Borenstein, 2001)



Gambar 2.1 Robot Pemandu Orang Dengan Penyandang Tuna Netra.  
(Ulrich dan Borenstein, 2001)

b. Sistem Navigasi *Waypoint* pada *Autonomous Mobile Robot*

Penelitian terkait ini membahas tentang system navigasi robot beroda agar robot bisa bergerak secara mandiri dengan menggunakan metode *waypoint* di mana menggunakan modul *GPS receiver* sebagai penentu posisi dan *magnetic compass* digunakan sebagai penentu arah dalam sistem navigasi. *Autonomous Mobile Robot* ini bergerak secara mandiri di darat di area luar ruangan.



Gambar 2.2 Perancangan Bentuk dan Pemasangan Sensor  
(Taufik, 2012).

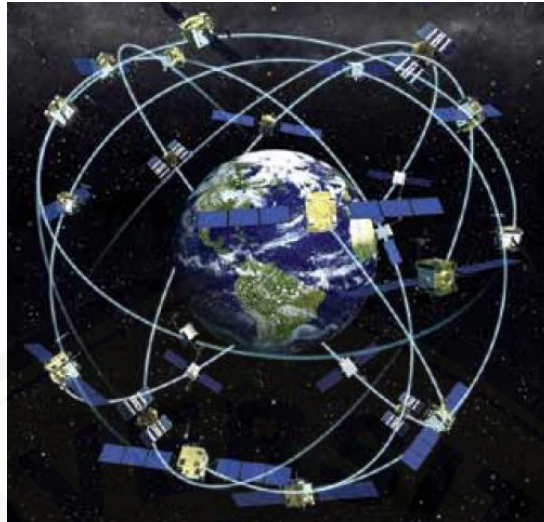
*Autonomous Mobile Robot* ini menggunakan *Atmega32* sebagai mikrokontroler utama yang memproses semua masukan mulai dari posisi koordinat bumi yang didapat dari modul *GPS*, arah robot dari *magnetic compass* dan beberapa

masukan lain serta menentukan keluaran atau tindakan apa yang akan diberikan oleh robot. Dibantu dengan mikrokontroler Atmega8 sebagai pengatur masukan *Driver* motor DC EMS 5A H-Bridge, sehingga kecepatan dan arah gerak motor DC dapat diatur. (Taufik, 2012)

## **2.2 Global Positioning System**

GPS atau nama formalnya NAVSTAR GPS (*NAVigation Satellite Timing and Ranging Global Positioning System*) merupakan sebuah sistem radio navigasi yang mampu menentukan posisi menggunakan beberapa satelit yang saling terhubung. Proyek arsitektur dan sistem GPS dimulai pada tahun 1973 oleh Departemen Pertahanan Amerika Serikat. Satelit yang pertama diluncurkan pada tahun 1978, pada tanggal 26 Juni 1993 24 satelit diluncurkan, dan sistem GPS secara resmi dinyatakan beroperasi pada tahun 1994. Bradford Parkinson, Roger L. Easton, dan Ivan A. Dinobatkan sebagai orang yang telah menciptakan GPS.

Pada dasarnya GPS terdiri dari tiga segmen utama dalam pengoperasiannya, yaitu segmen angkasa (*space segment*) yang terdiri dari satelit-satelit GPS yang mengorbit sekitar 20.200 km di atas permukaan bumi, segmen sistem kontrol (*control system segment*) yang terdiri dari stasiun-stasiun pemonitor dan pengontrol satelit, dan segmen pengguna (*user segment*) yang terdiri dari pengguna GPS termasuk alat-alat penerima, pengolah sinyal dan data GPS. Konsep GPS dalam mengambil data didasarkan pada waktu yang didapat dari minimal tiga satelit untuk mendapatkan lokasi yang cukup akurat. Setiap satelit GPS memiliki pencatat waktu yang saling terhubung dengan satelit lainnya dan waktu yang ada di bumi dengan pembaharuan lokasi satelit secara terus-menerus. GPS memberikan sinyal satelit dengan kode khusus yang nantinya akan diproses oleh penerima GPS, hal ini memungkinkan penerima untuk menghitung posisi, waktu dan kecepatan.



Gambar 2.3 Segmen Angkasa GPS (ilmuterbang.com)

Jadi pada dasarnya, GPS (*Global Positioning System*) adalah suatu sistem navigasi yang menyediakan informasi lokasi, kecepatan tiga dimensi dan waktu secara kontinyu tanpa bergantung waktu dan cuaca di mana saja di bagian bumi di mana ada garis pandang yang tidak terhalang untuk tiga atau lebih satelit GPS. Luas spektrum ketelitian GPS dalam menentukan posisi merupakan salah satu keunggulan yang dimiliki. Pengguna GPS mempunyai keleluasan dalam menentukan posisi sesuai dengan tingkat ketelitian yang diperlukan secara optimal dan efisien sesuai waktu dan biaya. Sebab demikian GPS dapat memberikan pelayanan yang cukup baik sesuai dengan tuntutan ketelitian yang beragam misalnya ketelitian dalam level meter untuk pemetaan dan dalam level milimeter untuk penentuan posisi geodesi.

### 2.3 Waypoint

Pada dasarnya metode *waypoint* adalah metode navigasi yang menggunakan beberapa titik tujuan awal dalam menuju titik tujuan utama dengan menggunakan titik koordinat untuk menentukan lokasi tersebut. Koordinat-koordinat itu biasanya menyertakan longitude, latitude, dan kadang altitude untuk keperluan navigasi di udara. Namun dalam kasus penelitian ini *waypoint* yang penulis gunakan cukup dengan *latitude* dan *longitude* dikarenakan koordinat tersebut untuk system navigasi robot beroda yang berjalan di permukaan tanah. Pada metode *waypoint*

inilah kita akan membutuhkan dua komponen penting yaitu GPS dan Sensor Kompas. GPS untuk mencari lokasi robot sekarang dan lokasi tujuan robot yang diinisialisasikan dalam bentuk titik koordinat dua sumbu yaitu  $x$  dan  $y$  di mana  $x$  *latitude* dan  $y$  *longitude*. Dan Sensor Kompas untuk mencari arah *bearing* robot sekarang.

Dalam metode waypoint setelah menentukan koordinat-koordinat yang akan dituju maka kita membutuhkan sudut arah (*bearing*) tujuan antara titik posisi sekarang dan titik posisi tujuan. Kita bisa dapatkan *bearing* tujuan dengan menggunakan rumus sebagai berikut :

$$\beta = \text{atan2} (X, Y)$$

Di mana,  $X$  dan  $Y$  adalah dua nilai yang dapat dihitung dari:

$$X = \cos \theta_b * \sin \Delta L$$

$$Y = \cos \theta_a * \sin \theta_b - \sin \theta_a * \cos \theta_b * \cos \Delta L$$

Setelah mendapatkan arah tujuan maka selanjutnya adalah sejauh mana jarak yang harus ditempuh untuk mencapai titik tujuan yang diinginkan. Jarak bisa didapat dengan rumus sebagai berikut :

$$a = \sin^2 (\Delta\theta / 2) + \cos (\theta_a) * \cos (\theta_b) * \sin^2 (\Delta L / 2)$$

$$D = R * 2 * \text{atan2} (\sqrt{a}, \sqrt{1-a})$$

(Akshay Upadhyay, 2013)

Keterangan :

$\beta$	= Bearing	$\Delta L$	= Delta Longitude
$\theta_b$	= Latitude Tujuan	$D$	= Jarak
$\theta_a$	= Latitude Sekarang	$R$	= Jari-Jari Bumi (6372,79547759 km)

Pada contoh kasus misalnya, kita sudah inisialisasikan koordinat tujuan pada program yaitu (-8.165042, 113.714393) dan didapat lokasi robot sekarang oleh GPS adalah (-8.165172, 113.714344) dan arah *bearing* robot sekarang oleh sensor kompas adalah 102° dengan 0° adalah utara, 90° timur dan seterusnya. Maka sebelum kita melakukan perhitungan, kita perlu merubah format koordinat ke dalam bentuk radian.

$$\begin{aligned} X &= \cos(-0.14250913539165) * \sin(1.98468970954546 - 1.98469056475679) \\ &= 0.989862747 * -0.00000085521133375 \\ &= -0.0000008465418400338 \end{aligned}$$

$$\begin{aligned} Y &= \cos(-0.142506866463623) * \sin(-0.14250913539165) - \sin(-0.142506866463623) \\ &\quad * \cos(-0.14250913539165) * \cos(1.98468970954546 - 1.98469056475679) \\ &= 0.989863069 * -0.0000022459276956 * 0.989862747 * 1 \\ &= -0.0000022006241374512 \end{aligned}$$

$$\begin{aligned} \beta &= \text{atan2}(X, Y) \\ &= \text{atan2}(-0.0000008465418400338, -0.0000022006241374512) \\ &= -2.74 = 202^\circ \end{aligned}$$

Jadi selisih *bearing* 102° - 202° = 100° robot berputar 100° searah jarum jam.

$$\begin{aligned} a &= \sin^2(-0.000002268928028/2) + \cos(-0.142506866463623) * \cos(-0.142509135391650) * \sin^2(-0.000000855211334/2) \\ &= (0.0000000000012870086 + 0.989863069) * 0.989862747 \\ &\quad * 0.00000000000018284661 \\ &= 0.0000000000001792 \end{aligned}$$

$$\begin{aligned} D &= R * 2 * \text{atan2} \\ &(\sqrt{0.0000000000001792}, \sqrt{1 - 0.0000000000001792}) \\ &= 6372795.477598 * 2 * 0.00000122709100385997 \\ &= 15.64 \text{ meter} \end{aligned}$$

Jadi robot akan berjalan ke arah 202° sejauh 15.64 meter.

Dengan diketahuinya *bearing* tujuan maka bisa kita dapatkan selisih *bearing* tujuan dan *bearing* sekarang, sehingga robot akan mengetahui sejauh mana dia berputar untuk menuju *bearing* tujuan. Dan dengan diketahuinya jarak dari titik lokasi sekarang dengan titik lokasi tujuan maka robot bisa mengetahui sejauh mana dia berjalan ke depan untuk mencapai lokasi tujuan yang ditentukan dan melanjutkan ke tujuan selanjutnya dengan cara yang sama, begitulah metode *waypoint* bekerja.

## 2.4 Linkit ONE

The Linkit ONE adalah mikrokontroler yang dikembangkan dan dirilis oleh Mediatek dan SeeedStudio. Board ini dibuat untuk semua para pembuat di dunia ini untuk membuat peralatan yang dapat dikenakan oleh manusia dan proyek-proyek IOT. Hal spesial dari chip ini adalah sudah dilengkapi dengan Bluetooth, Wifi, GPS dan GSM yang dikemas dalam paket kecil. Menggunakan bahasa pemrograman yang sama dengan Arduino sehingga memudahkan proyek pembuatan dan prototipe. Pinout ini agak mirip dengan Arduino sehingga semua hiasan Arduino kompatibel dengannya. Board ini juga memiliki slot kartu SD, jack audio dan soket terpisah untuk I2C dan UART.



Gambar 2.4 Bentuk Fisik Linkit ONE (instructable.com)



Berikut ini adalah spesifikasi dari board ini:

Mikrokontroler :

- Chipset : MT 2502
- Inti : ARM7 EJ-S

Power :

- 3.7-4.2v Baterai Lithium ATAU
- 5v USB

Pin I / O digital :

- Nomor : 16 (D0-D13, SDA, SCL)
- Tegangan : 3.3v maks.

Masukan pin analog :

- No : 3 (A0, A1, A2)
- Tegangan : 0-5v

Pin output PWM :

- No: 2 (D3, D9)
- Tegangan: 0-3.3v

I2C :

- Nos : 1 (SDA, SCL)

SPI :

- Nos : 1 (D11, D12, D13)

UART :

- Nos: 1 (D0, D1)

Penyimpanan Pengguna:

- 10MB + kartu SD yang dapat diupgrade hingga 32GB ([instructable.com](http://instructable.com))

Pertimbangan penulis dalam memilih mikrokontroler Linkit One ini dikarenakan beberapa hal yaitu pertama, Linkit One masih belum terlalu populer dalam penggunaannya di Indonesia. Yang ke dua, dengan melihat spesifikasi yang dimiliki Linkit One membuat penelitian ini masih punya aspek yang bisa dikembangkan terutama dalam hal IOT untuk lebih mempermudah dalam pengalokasian dan memperluas manfaatnya.

### 2.5 Magnetic Compas HMC5883L

Sensor kompas HMC5883L merupakan salah satu dari jenis sensor elektronik yang dapat mengetahui kutub selatan dan utara bumi dan menampilkan nilai secara digital. Modul kompas juga dapat digunakan untuk mengukur kekuatan dari sumber magnet terdekat. Ukuran sensor kompas ini didesain khusus untuk sistem navigasi robot, ukurannya yang kecil sehingga mudah digunakan untuk alat yang kecil pula seperti robot beroda pemandu orang dengan penyandang tuna netra ini.

Fungsi Sensor Magnetic Compas HMC5883L dalam membantu navigasi robot biasanya untuk memberikan referensi robot berada di posisi mana dan mengarah kemana, kemudian posisi dan arah yang diberikan oleh Sensor Magnetic Compas HMC5883L tersebut sebagai referensi arah gerakan robot selanjutnya. Sensor magnetic compas ini menggunakan jalur komunikasi data I2C untuk komunikasi dengan mikrokontroler yaitu SDA dan SCL, sehingga bisa digunakan pada jenis mikrokontroler yang support dengan komunikasi data I2C. Sensor Magnetic Compas HMC5883L membutuhkan tegangan 3 V- 5 V dengan arus 15mA.



Gambar 2.5 Sensor *Magnetic Compas* HMC5883L (instructable.com)

Feature yang dimiliki sensor HMC5883L di antaranya *Power Requirements: 2.7 to 6.5 VDC, Communication Interface: I2C (up to 400 kHz)*, dimensi: 0.725 x 0.650 in (1.8 x 1.7 cm). Karena Sensor Magnetic Compas

HMC5883L menggunakan I2C kita dapat menghubungkannya dengan Linkit One dengan cara :

- Linkit One GND -> HMC5883L GND
- Linkit One 3.3V -> HMC5883L VCC
- Linkit One A4 (SDA) -> HMC5883L SDA
- Linkit One A5 (SCL) -> HMC5883L SCL

Dengan menggunakan sumber tegangan DC +5 volt dan memiliki komunikasi data berbasis I2C maka sensor kompas HMC5883L ini dapat diaplikasikan dengan baik menggunakan mikrokontroler maupun komputer. (e-belajarelektronika.com)

## 2.6 LCD 16 x 2

LCD (*Liquid Cristal Display*) merupakan salah satu media tampil yang menggunakan media cair untuk menampilkannya. LCD (*Liquid Cristal Display*) digunakan untuk menampilkan suatu data, baik berupa karakter, huruf ataupun grafik. LCD (*Liquid Cristal Display*) adalah salah satu jenis display elektronik yang dibuat dengan teknologi CMOS logic yang bekerja dengan cara memantulkan cahaya yang ada di sekelilingnya terhadap front-lit atau mentransmisikan cahaya dari back-lit. (elektronika-dasar.web.id.2012)



Gambar 2.6. LCD (leselektronika.com)

Pin, kaki atau jalur input dan kontrol dalam suatu LCD (*Liquid Cristal Display*) di antaranya adalah :

- **Pin data** adalah jalur untuk memberikan data karakter yang ingin ditampilkan menggunakan LCD (*Liquid Cristal Display*) dapat dihubungkan dengan bus data dari rangkaian lain seperti mikrokontroler dengan lebar data 8 bit.

- **Pin RS (*Register Select*)** berfungsi sebagai indikator atau yang menentukan jenis data yang masuk, apakah data atau perintah. Logika low menunjukkan yang masuk adalah perintah, sedangkan logika high menunjukkan data.
- **Pin R/W (*Read Write*)** berfungsi sebagai instruksi pada modul jika low tulis data, sedangkan high baca data.
- **Pin E (*Enable*)** digunakan untuk memegang data baik masuk atau keluar.
- **Pin VLCD** berfungsi mengatur kecerahan tampilan (kontras) di mana pin ini dihubungkan dengan trimpot 5 Kohm, jika tidak digunakan dihubungkan ke ground, sedangkan tegangan catu daya ke LCD sebesar 5 Volt. (elektronika-dasar.web.id, 2012)



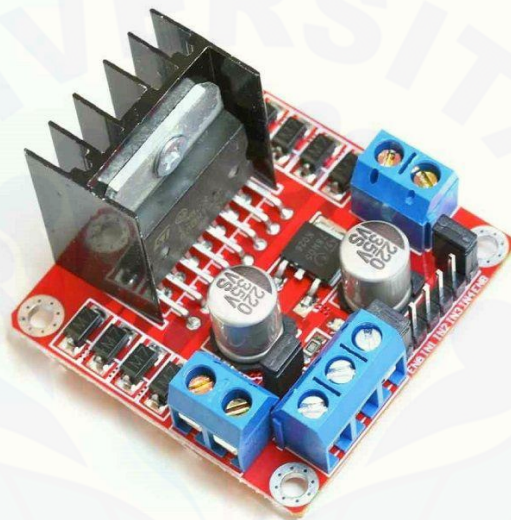
Gambar 2.7. I2C LCD. (belajararduino.com)

Karena LCD cukup banyak membutuhkan Pin maka penulis menggunakan modul I2C untuk membuat lcd bisa dioperasikan hanya dengan menggunakan komunikasi I2C yaitu dengan 4 Pin yaitu Ground, VCC, SDA dan SCL.

## 2.7 Driver Motor

Ada beberapa macam driver motor DC yang biasa kita pakai seperti menggunakan relay yang diaktifkan dengan transistor sebagai saklar, namun yang demikian dianggap tidak efisien dalam pengerjaan hardwarenya. Dengan berkembangnya dunia IC, sekarang sudah ada H Bridge yang dikemas dalam satu IC di mana memudahkan kita dalam pelaksanaan hardware dan kendalinya apalagi jika menggunakan mikrokontroler, penulis rasa akan lebih mudah lagi penggunaannya. IC yang familiar seperti IC L298. IC ini memiliki kelebihan dan

kekurangan masing-masing. Modul yang menggunakan IC driver L298 yang memiliki kemampuan menggerakkan motor DC sampai arus 4A dan tegangan maksimum 46 VoltDC untuk satu kanalnya. Rangkaian driver motor DC dengan IC L298 diperlihatkan pada gambar 1. Pin Enable A dan B untuk mengendalikan jalan atau kecepatan motor, pin Input 1 sampai 4 untuk mengendalikan arah putaran. Pin Enable diberi VCC 5 Volt untuk kecepatan penuh dan PWM (Pulse Width Modulation) untuk kecepatan rotasi yang bervariasi tergantung dari level highnya. (digilib.mercubuana.ac.id)

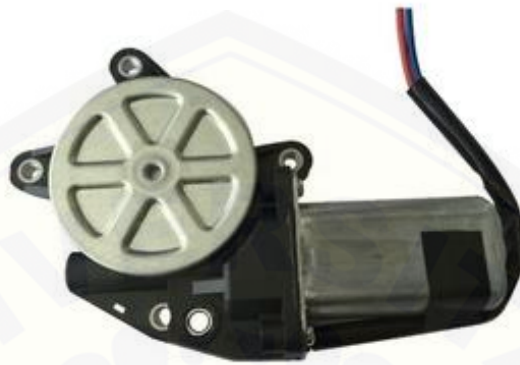


Gambar 2.8. Driver motor DC dengan L298. (instructables.com)

## 2.8 Motor DC

Motor DC merupakan salah satu jenis peralatan elektromekanik yaitu sebuah aktuator yang berfungsi untuk mengubah tenaga listrik menjadi tenaga mekanik. Motor DC menjadi sebuah aktuator yang sangat penting bagi segala jenis robot khususnya robot beroda. Dalam pemakaian motor DC pada robot beroda biasa digunakan untuk memutar roda. Sehingga sangat perlu kita ketahui bahwa kecepatan putar motor dipengaruhi oleh besar tegangan yang diberikan sedangkan arah putarnya ditentukan oleh polaritas tegangannya. Sedangkan untuk mengendalikan kecepatan dan arah putar motor kita tidak bisa menggunakan mikrokontroler secara langsung dikarenakan dibutuhkan arus yang besar sedangkan arus dari mikrokontroler sangat kecil. Dari permasalahan itu kita bisa menggunakan *Driver Motor* sebagai alternatif atau penghubung antara mikrokontroler dan motor

DC untuk mengatur kecepatan dan arah putarnya. Dalam penelitian ini penulis menggunakan motor DC power windows yang memiliki torsi cukup besar yaitu 3 N.m. (elib.unikom.ac.id)



Gambar 2.9 Motor DC *Power Window*. (alibaba.com)

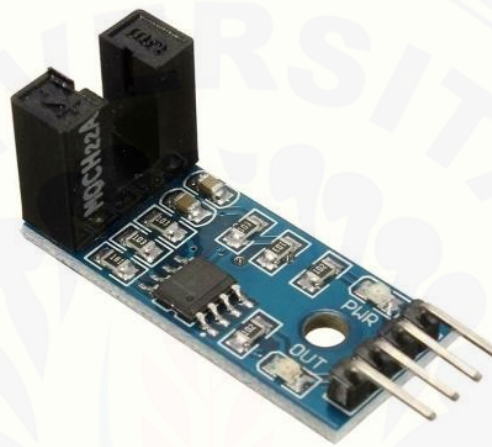
Prinsip kerja motor DC di antaranya adalah sebagai berikut :

1. Magnet pada stator memberikan garis-garis gaya medan magnet (fluks) antar kutub.
2. Di area medan magnet tadi terdapat penghantar yang akan dialiri oleh arus listrik.
3. Sehingga pada penghantar timbul gaya yang menghasilkan torsi dan membuat penghantar tadi berputar. (digilib.mercubuana.ac.id)

Dengan mempertimbangkan tujuan dari tugas akhir ini maka penulis menggunakan motor DC *Power Window* yang biasa digunakan untuk menaikkan dan menurunkan kaca candela pada mobil. Pilihan ini penulis pilih karena motor *Power Window* memiliki torsi yang cukup besar yaitu 3 Nm (30 Kg.cm) sehingga mampu memutar roda meski dengan beban robot yang cukup berat. Motor DC *Power Window* membutuhkan sumber tegangan 12 Volt dan arus dalam kondisi tanpa beban sebesar 2,8 Amper.

## 2.9 Sensor *Rotary Encoder*

*Rotary encoder* adalah divais elektromekanik yang dapat memonitor gerakan dan posisi. *Rotary encoder* umumnya menggunakan sensor optik untuk menghasilkan serial pulsa yang dapat diartikan menjadi gerakan, posisi, dan arah. Sehingga posisi sudut suatu poros benda berputar dapat diolah menjadi informasi berupa kode digital oleh *rotary encoder* untuk diteruskan oleh rangkaian kendali. *Rotary encoder* umumnya digunakan pada pengendalian robot, motor *drive*, dsb.



Gambar 2.10 *Rotary Encoder*. (instructable.com)

*Rotary encoder* tersusun dari suatu piringan tipis yang memiliki lubang-lubang pada bagian lingkaran piringan. LED ditempatkan pada salah satu sisi piringan sehingga cahaya akan menuju ke piringan. Di sisi yang lain suatu photo-transistor diletakkan sehingga photo-transistor ini dapat mendeteksi cahaya dari LED yang berseberangan. Piringan tipis tadi dikopel dengan poros motor, atau didesain berputar lainnya yang ingin kita ketahui posisinya, sehingga ketika motor berputar piringan juga akan ikut berputar. Apabila posisi piringan mengakibatkan cahaya dari LED dapat mencapai photo-transistor melalui lubang-lubang yang ada, maka photo-transistor akan mengalami saturasi dan akan menghasilkan suatu pulsa gelombang persegi. Semakin banyak deretan pulsa yang dihasilkan pada satu putaran menentukan akurasi *rotary encoder* tersebut, akibatnya semakin banyak jumlah lubang yang dapat dibuat pada piringan menentukan akurasi *rotary encoder* tersebut. (Rizqiawan, 2009)

### BAB 3. METODE PENELITIAN

Pada bab metode penelitian ini menjelaskan tentang beberapa hal pokok yaitu tempat dan waktu, ruang lingkup, diagram alir (*flowchart*), jenis dan sumber data serta metode pengumpulan data yang dilakukan dalam menyelesaikan penelitian tugas akhir ini. Dalam penelitian ini akan menghitung akurasi robot pemandu dalam menentukan lokasi target dan akurasi robot dalam menentukan jalur untuk memandu penyandang tuna netra.

#### 3.1 Waktu dan Tempat Kegiatan

Skripsi yang berjudul tentang “Rancang Bangun Sistem Navigasi Robot Pemandu Orang Dengan Penyandang Tuna Netra Dengan Metode *Waypoint*”. Pelaksanaan pembuatan alat akan dilaksanakan di Laboratorium Sistem Cerdas dan Robotika CDAST, Universitas Jember yang beralamat di Jln. Kalimantan, Krajan Timur, Sumbersari, Kec. Sumbersari, Kabupaten Jember. Sedangkan pengujian alat dilaksanakan di gedung KAUJE Universitas Jember kampus Tegalboto. Pembuatan alat ini akan dimulai pada bulan Februari 2019.

Tabel 3.1 Rencana Kegiatan Tugas Akhir

No.	Kegiatan	Bulan ke-								
		I			II			III		
1	Studi literatur dan Pembuatan laporan Bab 1 sampai Bab 3	■	■							
2	Pembuatan rangkaian penyusun sistem dan Konsultasi		■	■	■					
3	Proses Kalibrasi dan Konsultasi				■	■	■			
4	Pengujian alat dan Konsultasi					■	■	■	■	
5	Menganalisa data hasil pengujian dan Konsultasi								■	■
6	Pembuatan Laporan									■

Keterangan:



: Kegiatan dilaksanakan



### 3.2 Ruang Lingkup Kegiatan

Ruang lingkup kegiatan ini berisi tentang batasan-batasan masalah dalam pembuatan alat, di mana batasan-batasan tersebut adalah sebagai berikut:

- a. Menggunakan robot beroda berjenis *differensial steering*.
- b. Sensor ultrasonik di *body* depan robot untuk mendeteksi halangan dan lubang.
- c. Menggunakan mikrokontroler LinkiT ONE sebagai pemroses data, penentu posisi titik ordinat bumi (Modul GPS).
- d. *Magnetic compass* yang digunakan sebagai penentu arah robot.
- e. Sensor *rotary encoder* yang dipasang pada as roda kanan dan kiri robot.

### 3.3 Jenis dan Sumber Data

Jenis dan sumber data di sini akan menjelaskan tentang keseluruhan alat yang akan dibuat, sebagai berikut:

#### a. Alat dan Bahan

Alat dan bahan yang digunakan dalam penelitian sebagai berikut:

- 1) *Gear box* robot beroda digunakan sebagai kerangka dari robot.
- 2) Motor Dc yang berjumlah dua buah yang ditempatkan di kanan dan kiri robot digunakan untuk membuat robot dapat berpindah posisi.
- 3) *Driver motor* digunakan untuk mengontrol arah putaran dan kecepatan motor DC dan untuk melindungi mikrokontroler dari arus balik motor.
- 4) Sensor *rotary encoder*. Sensor *rotary encoder* yang berjumlah dua buah yang dipasang di as roda kanan dan kiri yang digunakan untuk merekam jumlah putaran roda sehingga diperoleh jarak tempuh robot.
- 5) *Magnetic compass* yang digunakan sebagai penentu arah robot.
- 6) LinkiT ONE sebagai pemroses data, penentu posisi titik ordinat bumi (Modul GPS) dan pemantau posisi robot yang mengirimkan data pada Android.

Alat dan bahan yang digunakan di atas juga mencakup seperti solder, PCB, timah, resistor, kabel pelangi, arpus ,bordan lain-lain.

### 3.4 Metode Pengumpulan Data

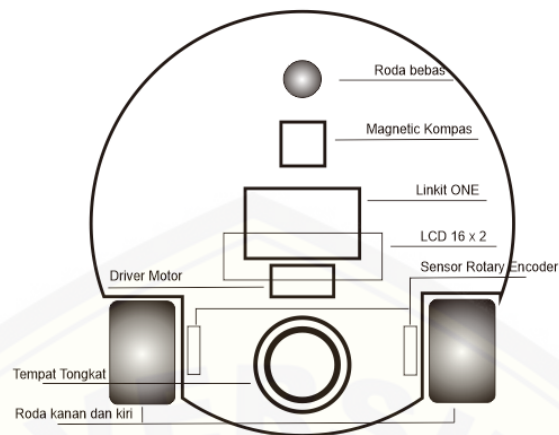
Dalam proses pembuatan rancang alat tugas akhir ini menggunakan sensor *rotary encoder*. Adapun langkah-langkah penelitian yaitu:

- a. Studi Literatur.
- b. Melakukan perancangan perangkat keras dan perangkat lunak.
- c. Melakukan pembuatan rangkaian penyusun sistem.
- d. Melakukan pengujian pengintegrasian perangkat keras dan perangkat lunak.  
Pertama pengujian ini dilakukan secara terpisah dan selanjutnya akan dilakukan pengujian secara keseluruhan.
- e. Menganalisa data yang telah diperoleh saat pengujian.

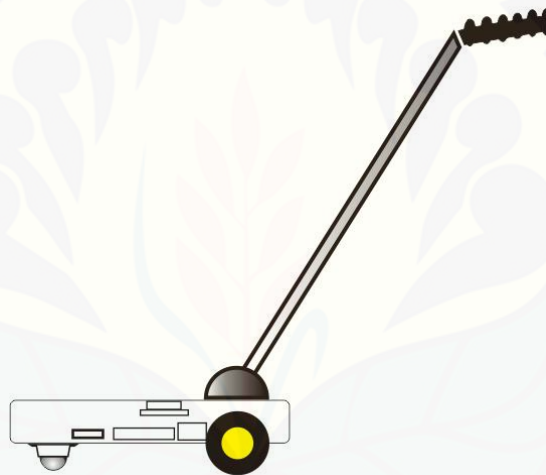
### 3.5 Perancangan Alat

#### 3.5.1 Perancangan Desain Alat

Robot beroda ini menggunakan Linkit ONE yang sudah dilengkapi dengan modul GPS sebagai penentu letak posisi ordinat bumi yang juga bisa dipantau dengan menggunakan android. Kemudian robot ini dilengkapi dengan modul kompas untuk mengetahui arah dari robot dan selanjutnya akan mengambil arah mana. Untuk meminimalkan eror dari posisi robot saat melakukan pergerakan, kami memasang dua buah sensor *rotary encoder* yang dipasang di as motor kanan dan motor kiri. Di mana sensor *rotary encoder* mampu menghitung kecepatan dan banyaknya putaran roda untuk mengetahui jarak tempuh dari robot. Kemudian kami gunakan LCD 16 x 2 sebagai penampil jarak robot dengan jarak tujuan.



Gambar 3.1 Perancangan Desain Alat

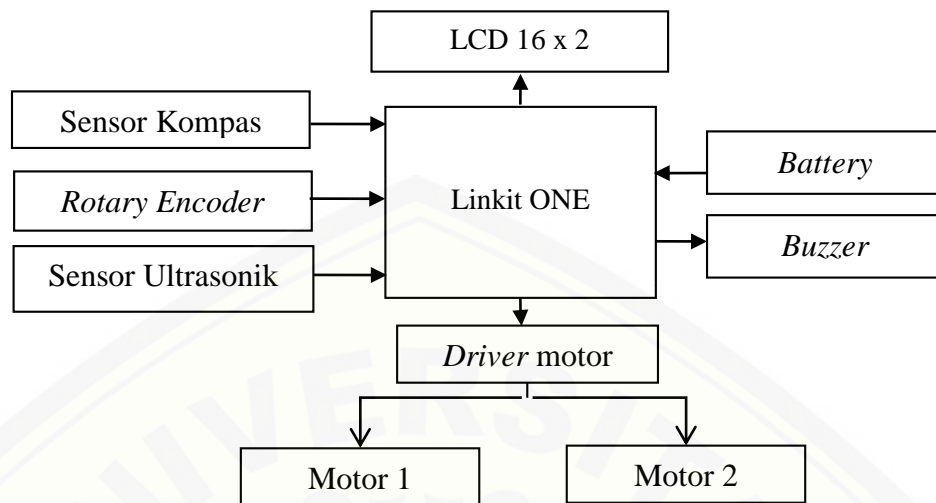


Gambar 3.2 Alat Tampak Samping

### 3.5.2 Perancangan Perangkat Keras

Pada bagian ini akan dijelaskan berupa perencanaan perangkat keras yang akan digunakan.

## 3.5.2.1 Blok Diagram

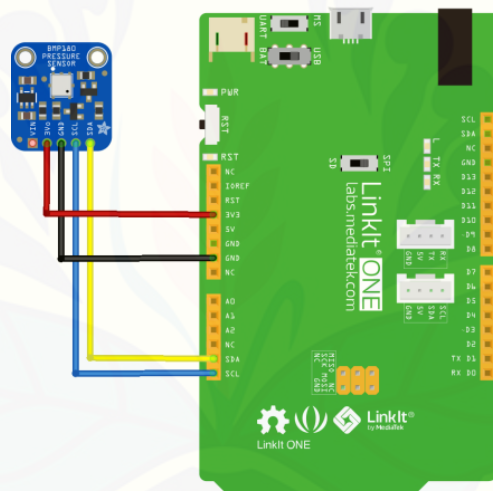


Gambar 3.3 Blok Diagram

Gambar 10. blok diagram robot beroda sebagai pemandu orang dengan Penyandang Tuna Netra yang melakukan navigasi secara otomatis di luar ruangan di mana menggambarkan *mindmap* dari alur cara kerja robot yang akan dibuat. Bagian masukan ada satu catu daya dengan menggunakan sebuah baterai sebagai sumber tegangan keseluruhan, sensor kompas yang memberikan informasi mengenai arah robot di awal dan memberikan referensi selanjutnya arah mana yang akan diambil oleh robot, sensor *rotary encoder* yang memberikan informasi mengenai jumlah putaran dari motor serta modul GPS yang sudah tertanam pada Linkit ONE sebagai pemberi jalur robot secara *virtual*. Pada bagian proses ada Linkit ONE sebagai mikrokontroler yang memberikan proses lanjutan dari data informasi masukan yang didapat. Kemudian pada bagian keluaran ada *Driver motor* yang memberikan kontrol kepada motor kanan dan motor kiri sehingga mampu memberikan posisi gerak yang tepat terintegrasi dengan sensor kompas dan *Rotary Encoder*, *Ultrasonic* untuk mendeteksi halang rintang dan lubang di depan robot, Linkit ONE untuk memberikan informasi posisi robot yang bisa dipantau dengan menggunakan PC serta *buzzer* sebagai penanda bahwa robot sudah berada di posisi tujuan.

### 3.5.2.2 Rangkaian Sensor Kompas dengan Linkit ONE

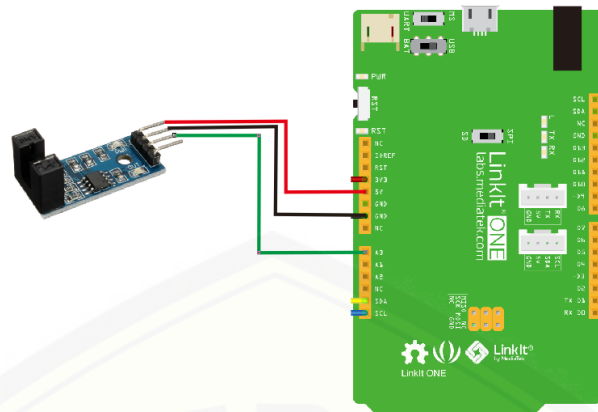
Linkit ONE merupakan board mikrokontroler yang menggunakan sistem dan bahasa pemrograman sama seperti Arduino namun board ini memiliki keunggulan dengan dilengkapi beberapa modul tambahan yang sudah tertanam pada board tersebut. Jadi, pada dasarnya pemasangan komponen dengan Linkit ONE sama seperti pada pemasangan komponen dengan Arduino. Sensor kompas memiliki lima pin di antaranya GND, VCC, SCL dan SDA. Untuk merangkai sensor kompas dengan Linkit ONE kita bisa menghubungkan GND sensor kompas dengan pin Ground yang ada pada Linkit ONE, VCC dihubungkan pada pin 5V pada Linkit ONE, pin SCL sensor kompas dihubungkan pada pin A4 Linkit ONE dan pin SDA sensor kompas dihubungkan pada pin A5 yang ada pada Linkit ONE.



Gambar 3.4 Rangkaian Sensor kompas dengan Linkit ONE. (iot.learning.net)

### 3.5.2.3 Pemasangan *Rotary Encoder* atau *Optocoupler* Pada Linkit ONE

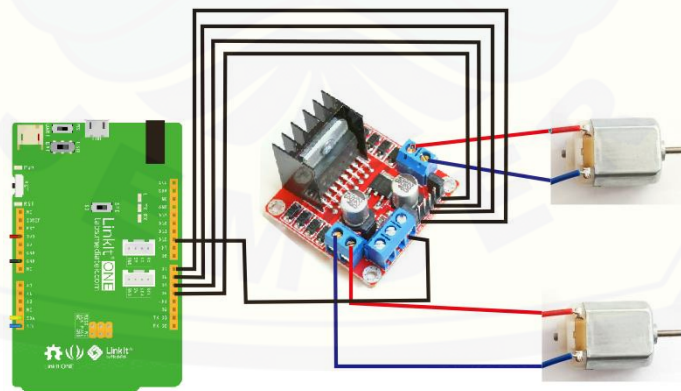
*Rotary Encoder* atau *Optocoupler* memiliki 4 pin yaitu pin VCC, ground, Analog Output dan Digital Output. Di mana dari empat pin tersebut hanya tiga pin yang digunakan yaitu pin Analog Output dan Pin Digital Output hanya dipakai salah satu saja tergantung keluaran apa yang kita inginkan. Sedangkan untuk merangkai *Rotary Encoder* dengan Linkit ONE, kita bisa menghubungkan pin VCC dengan pin 5V pada Linkit ONE, pin ground dengan pin GND pada Linkit ONE dan pin Analog Output dihubungkan dengan pin Analog pada Linkit ONE atau pin Digital Output dihubungkan dengan pin Digital pada Linkit ONE.



Gambar 3.5 Rangkaian *Rotary Encoder* dengan Linkit ONE.

#### 3.5.2.4 Rangkaian *Driver Motor* dengan Linkit ONE

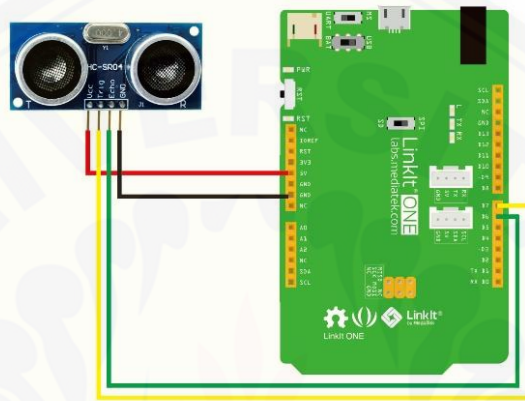
*Driver Motor* memiliki dua output ke motor1 dan dua output ke motor2, memiliki 4 pin *input control* untuk motor1 dan motor2 yang dihubungkan pada pin Linkit ONE, pin ground yang dihubungkan pada pin GND Linkit ONE, pin 5V untuk mikrokontroler yang dihubungkan pada pin PWM (*Pulse With Modulation*) dan pin 12V untuk sumber tegangan baterai. Di mana kita bisa mengatur arah putar motor melalui 4 pin *input control* yang mana dua untuk motor1 dan dua untuk motor2. Kemudian kita bisa mengatur kecepatan motor melalui pin 5V *driver motor* yang terhubung pada pin PWM Linkit ONE.



Gambar 3.6 Rangkaian *Driver Motor* dengan Linkit ONE.

#### 3.5.2.5 Pemasangan sensor *Ultrasonic* Pada Linkit ONE

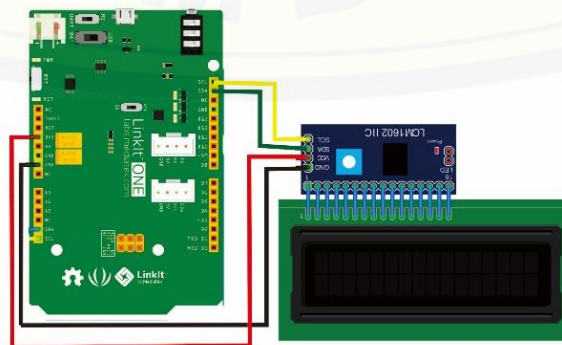
Sensor *Ultrasonic* memiliki empat pin di antaranya *Vcc*, *Trigger*, *Echo* dan *Ground*. Untuk merangkai sensor *Ultrasonic* dengan Linkit ONE kita bisa mulai dengan menghubungkan GND sensor *Ultrasonic* dengan pin Ground yang ada pada Linkit ONE, VCC dihubungkan pada pin 5V pada Linkit ONE, pin *Echo* sensor *Ultrasonic* dihubungkan pada pin D6 Linkit ONE dan pin *trigger* sensor *Ultrasonic* dihubungkan pada pin D7 Linkit ONE.



Gambar 3.7 Rangkaian sensor *Ultrasonic* dengan Linkit ONE

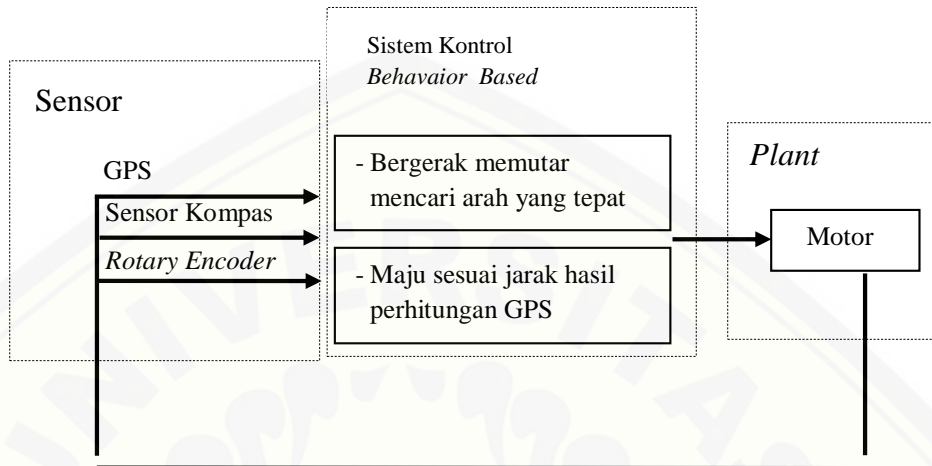
### 3.5.2.6 Pemasangan LCD 16 x 2 Pada Linkit ONE

LCD 16 x 2 memiliki 16 pin di mana pin 1 dan 16 adalah Ground, pin 2 dan 15 adalah VCC, pin 3 digunakan untuk mengatur kontras sehingga bisa kita hubungkan resistor variabel untuk mengatur kontrasnya, pin 4 (RS) dihubungkan pada pin mikrokontroler, pin 5 (RW) dihubungkan pada GND, pin 6 (E) dihubungkan pada pin mikrokontroler dan pin 11-14 sebagai jalur data dihubungkan pada digital pin mikrokontroler. (codepolitan.com)



Gambar 3.8 Rangkaian LCD 16 x 2 I2C dengan Linkit ONE

### 3.5.3 Perancangan Sistem Kontrol



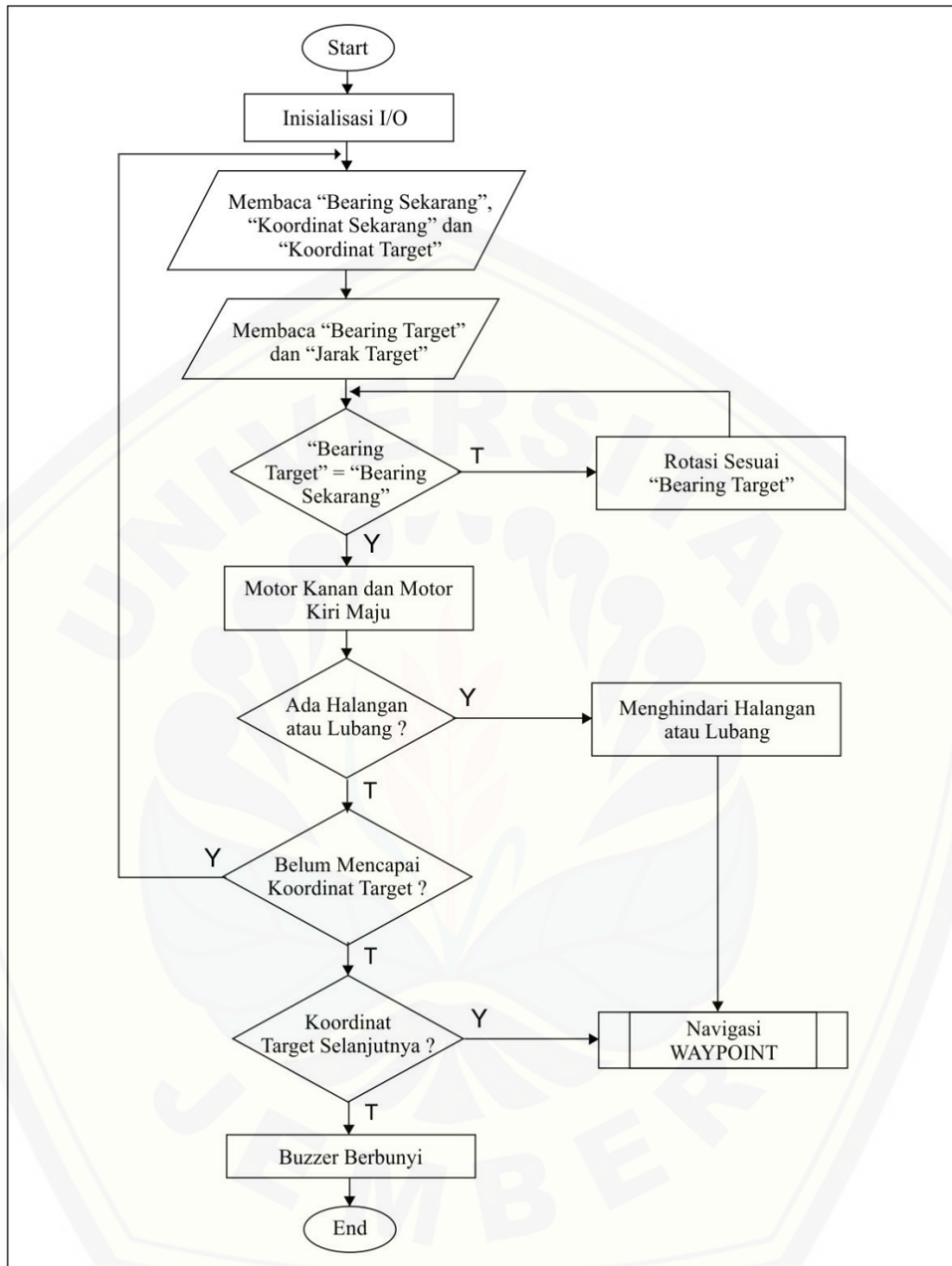
Gambar 3.9 Bagan Sistem Kontrol Robot

Pada perancangan kontrol robot ini, robot akan diprogram untuk berperilaku di mana GPS sebagai input yang akan memberikan jalur secara *virtual* kepada robot. Kemudian robot akan mulai bergerak memutar mencari arah yang sesuai dengan jalur yang sudah diberikan, di mana menggunakan sensor kompas dalam menentukan arah gerak yang benar. Setelah arah sudah sesuai dengan jalur dan posisi tujuan, maka robot mulai bergerak dikontrol dengan adanya sensor *Rotary Encoder*. Sehingga error dari jalur yang ditempuh oleh robot bisa diminimalkan.

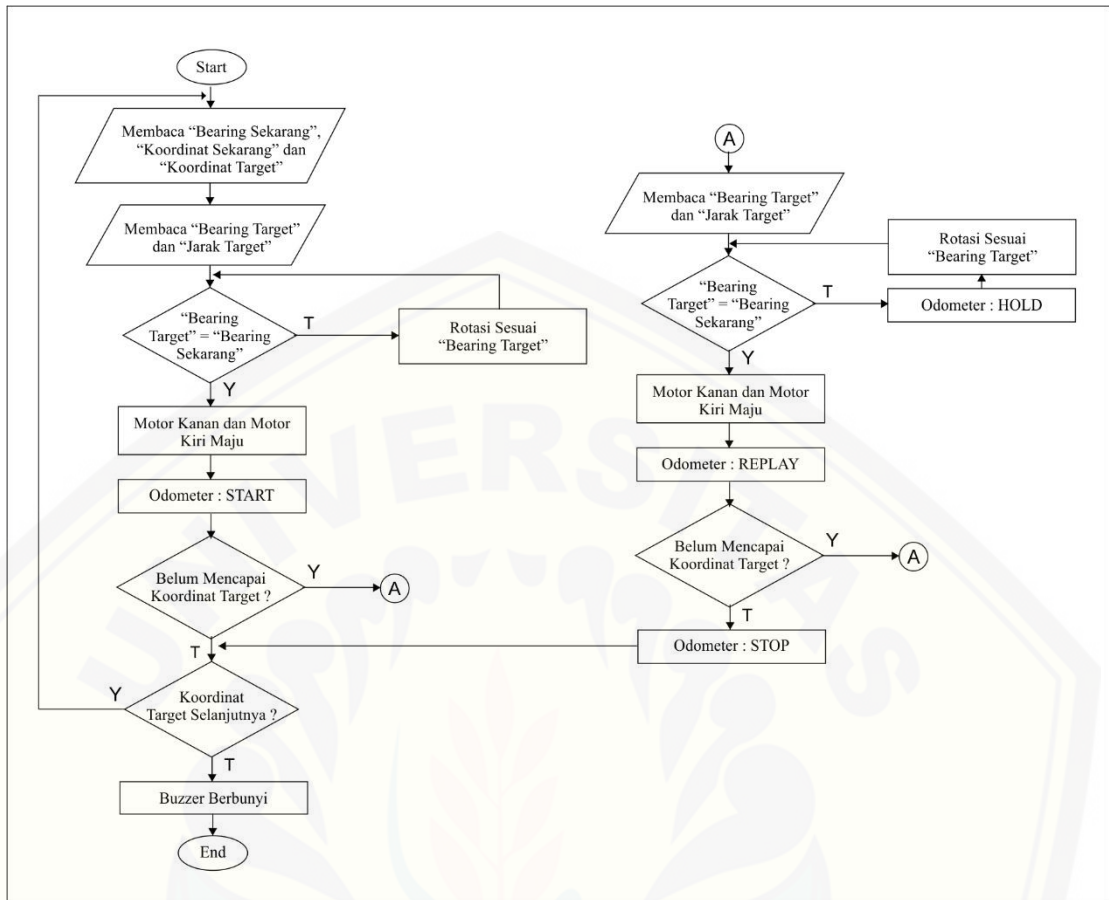
### 3.5.4 Perancangan Perangkat Lunak

#### 3.5.4.1 *Flowchart* Jalannya Robot Secara Keseluruhan



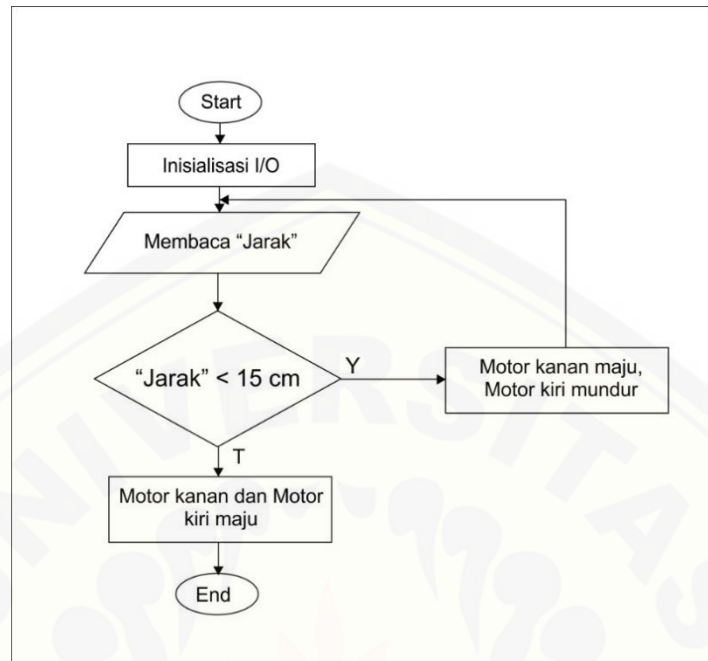


Gambar 3.10 Diagram Alir Perangkat Lunak Sistem Navigasi *Waypoint*.

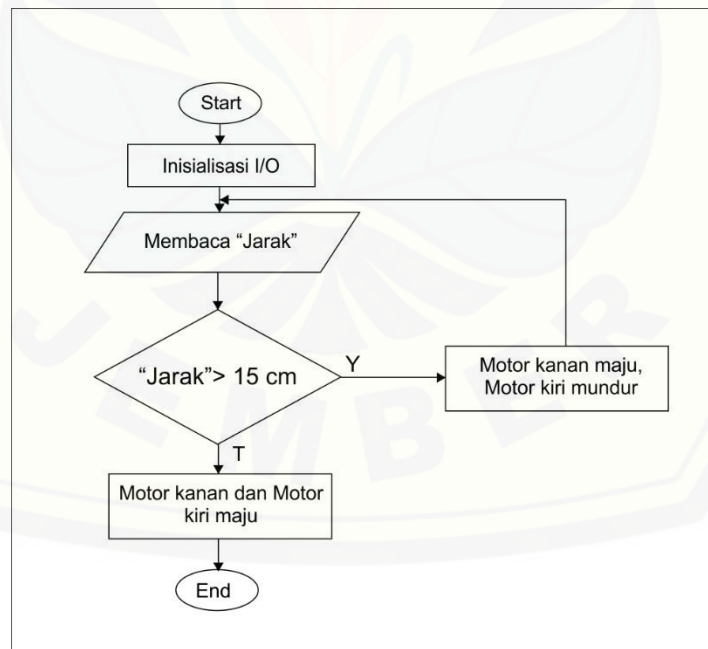


Gambar 3.11 Diagram Alir Perangkat Lunak Sub Fungsi Waypoint.

## 3.5.4.2 Flowchart Sensor Ultrasonik



Gambar 3.12 Diagram Alir Sensor Halangan.



Gambar 3.13 Diagram Alir Sensor Berlubang.

## BAB 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dari hasil yang telah diperoleh dalam penelitian yang sudah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Sensor HMC5883L memiliki akurasi yang tidak cukup baik. Hal demikian bisa dilihat pada Tabel 4.6 di mana sensor HMC5883L dengan duapuluh kali percobaan memiliki nilai pengukuran diantaranya dengan nilai eror lima percobaan di atas  $20^\circ$  dan rata-rata eror total  $9,05^\circ$ .
2. *Global Positioning system* Linkit ONE bekerja dengan cukup baik dan akurat. Hal ini bisa dilihat dari waktu rata-rata GPS Linkit ONE untuk mendapatkan sinyal satelit jika berada pada area terbuka adalah 23 detik dan membutuhkan waktu rata-rata 60 detik jika berada di area tertutup serta untuk mendapatkan latitude dan longitude yang stabil, robot membutuhkan waktu rata-rata 7 detik jika berada pada area terbuka dan membutuhkan waktu rata-rata 19 detik jika berada di area tertutup dengan nilai eror rata-rata di bawah radius 3 meter.
3. Robot dalam melakukan perhitungan mencari *bearing* dan jarak target memiliki eror yang cukup kecil di mana dalam tabel 4.8 percobaan *waypoint 1* dan *waypoint 2* robot mendapatkan nilai *bearing* dan jarak target  $200^\circ$  dan 15,43 meter sedangkan dalam perhitungan secara teori di bab 2 halaman 10 nilai *bearing* dan jarak target yang didapatkan adalah  $202^\circ$  dan 15,64 meter.
4. Robot beroda pemandu orang penyandang tuna netra dengan metode *waypoint* ini memiliki akurasi yang cukup baik di mana saat melakukan navigasi dengan empat *waypoint* yang berbelok-belok, robot masih memiliki nilai eror rata-rata di bawah radius 2 meter jika tidak ada halangan atau lubang dan nilai eror rata-rata di bawah radius 4 meter jika ada halangan atau lubang. Hal tersebut bisa dilihat dari Tabel 4.9 dan Tabel 4.10 pada nilai eror rata-rata setiap *waypoint* di 5 percobaan dengan dua pengujian.
5. System navigasi robot beroda dengan menggunakan metode *waypoint* memiliki akurasi yang lebih baik saat tidak ada halangan atau lubang dibandingkan saat ada halangan atau lubang. Hal ini bisa dilihat dari Tabel 4.9

nilai eror rata-rata pada saat tidak ada halangan atau lubang tidak lebih dari radius 2 meter sedangkan pada Tabel 4.10 nilai eror rata-rata pada saat ada halangan atau lubang bisa melebihi radius 2 meter.



## 5.2 Saran

Dengan harapan penelitian ini dapat dikembangkan dan disempurnakan di kemudian hari, penulis memberikan saran sebagai berikut:

1. Robot beroda pemandu orang penyandang tuna netra ini masih memerlukan banyak pengembangan dikarenakan masih banyak hal yang masih menyulitkan pengguna. Disarankan menambahkan output suara yang bisa memudahkan menggunakan dalam memahami kondisi yang ada di jalan.
2. Robot beroda pemandu orang penyandang tuna netra ini masih memiliki kekurangan dalam hal akurasi penentuan lokasi jika akan diterapkan secara serius untuk memandu orang buta. Sehingga disarankan menggunakan sensor *Global Positioning System* yang mampu mendapatkan sinyal satelit yang sangat cepat dan akurat.

**DAFTAR PUSTAKA**

- Abidin, Dr. Hasanuddin Z. 2000. Penentuan Posisi dengan GPS dan Aplikasinya. Jakarta : Pradnya Paramita.
- Awangga, Rolly Maulana. Februari 2019. Pengantar Sistem Informasi Geografis. Bandung : Kreatif Industri Nusantara.
- A. Hers, Marion dan Michael A. Jhonson. Desember 2010. A robotic guide for blind people. part 1. A multi nasional survey of the attitudes, requirements and preferences of potential end-users. Applied Bionics and Biomechanics, Vol. 7, No. 4, Desember 2010.
- Budiharto, Widodo. Juni 2006. Belajar Sendiri : Membuat Robot Cerdas. Jakarta : PT. Elex Media Komputindo.
- Hoque, Md. Ziaul. Maret 2016. Basic Concept of GPS and Its Applications IOSR Journal Of Humanities And Social Science , Volume 21. Issue 3. Ver. II. Diambil dari : [http://www.iosrjournals.org/iosr-jhss/papers/Vol.%2021%20Issue 3/Version-2/F2103023137.pdf](http://www.iosrjournals.org/iosr-jhss/papers/Vol.%2021%20Issue%203/Version-2/F2103023137.pdf). (09 Nopember 2019)
- Kulyukin, Vladimir, Chaitanya Gharpure dan Nathan De Graw. [Tanpa Tahun]. Human-Robot Interaction in a Robotic Guide for the Visually Impaired. Utah State University.
- Landasan Teori Motor DC Driver Motor. Diambil dari : [http://digilib.mercubuana.ac.id/manager/n!/file\\_skripsi/Isi2922943505149.pdf](http://digilib.mercubuana.ac.id/manager/n!/file_skripsi/Isi2922943505149.pdf). (05 Juni 2018)
- Liana, Diah. Desember 2016. Sistem Navigasi pada Mobile Robot dengan Global Positioning System (GPS). Annual Reasearch Seminar Vol. 2, No. 1, 6 Desember 2016.
- Nugroho, Seno. Agustus 2016. Sistem Navigasi Gerak Robo boat Berdasarkan GPS menggunakan Metode Waypoint. E-proceeding of engineering Vol. 3, No. 2, Agustus 2016.
- Purnomo, Dwi. 2016. Trigonometri, Ilmu Ukur Sudut. Malang : Gunung Samudera.
- Rizqiwani, A. 2009. Sekilas Rotary Encoder. Diambil dari : <https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/>. (05 Juni 2018)

- Saiyam. (Tanpa Tahun). Microcontrollers. Diambil dari : <http://www.instructables.com/id/LinkIt-ONE-Getting-Started-Guide/>. (05 Juni 2017)
- Sulkhan Taufik, Ahmad. 2013. Sistem Navigasi Waypoint pada Autonomous Mobile Robot. Diambil dari : <http://elektro.studentjournal.ub.ac.id/index.php/teub/article/download/7/3>. (29 Mei 2018)
- Supriyanto, Raden dkk. 2010. Robotika. Jakarta : Universitas Gunadarma.
- Ulrich, Iwan dan Johann Borenstein. Maret 2001. *The GuideCane—Applying Mobile Robot Technologies to Assist the Visually Impaired*. *Systems And Humans*, Vol. 31, No. 2, March 2001. Diambil dari : <http://ieeexplore.ieee.org/document/911370> /? denied. (29 Mei 2018)
- Wahyudi. (Tanpa Tahun). Landasan Teori Motor DC Driver Motor. Diambil dari : [http://elib.unikom.ac.id/files/disk1/452/jbptunikompp-gdl-wahyudinim-22574-2-unikom\\_w-i.pdf](http://elib.unikom.ac.id/files/disk1/452/jbptunikompp-gdl-wahyudinim-22574-2-unikom_w-i.pdf). (05 Juni 2018)
2014. Sensor Kompas CMPS03. Diambil dari : <http://e-belajarelektronika.com/sensor-kompas-cmps03/>. (05 Juni 2018)
2012. Liquid Crystal Display (LCD) 16 x 2. Diambil dari : <http://www.leselektronika.com/2012/06/liquid-crystal-display-lcd-16-x-2.html>. (05 Juni 2018)
2016. Belajar Rotary Encoder. Diambil dari : <https://depokinstruments.com/tag/belajar-rotary-encoder/>. (05 Juni 2018)
2012. LCD (Liquid Cristal Display). Diambil dari : <http://elektronika-dasar.web.id/lcd-liquid-cristal-display/>. (05 Juni 2018)



**Lampiran :**

**A. Listing Program Linkit ONE**

**Program kalibrasi HCSR-04**

```
#define trigPin 11

#define echoPin 12

void setup() {
    // initialize serial communication:
    Serial.begin(9600);
}

void loop() {
    // establish variables for duration of the ping, and the
    distance result
    // in inches and centimeters:
    long duration, inches, cm;

    // The PING))) is triggered by a HIGH pulse of 2 or more
    microseconds.

    // Give a short LOW pulse beforehand to ensure a clean HIGH
    pulse:
    pinMode(trigPin, OUTPUT);
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
```

```
pinMode(echoPin, INPUT);  
  
duration = pulseIn(echoPin, HIGH);  
  
cm = microsecondsToCentimeters(duration);  
  
Serial.print(cm);  
Serial.print("cm");  
Serial.println();  
  
delay(100);  
}  
  
long microsecondsToCentimeters(long microseconds) {  
    return microseconds / 29 / 2;  
}
```

## **Program Percobaan System Haling Rintang Lubang**

```
#define trig1 4  
#define echo1 7  
#define trig2 8  
#define echo2 9  
#define trig3 10  
#define echo3 11  
#define trig4 12  
#define echo4 13
```

```
#define trig5 5
#define echo5 6
#define trig6 2
#define echo6 3

#define motor1 A0
#define motor2 A1
#define motor3 A2
#define motor4 13

long s1, s2, s3, s4, s5, s6;

void setup() {
  Serial.begin (9600);
  pinMode(trig1, OUTPUT);
  pinMode(echo1, INPUT);
  pinMode(trig2, OUTPUT);
  pinMode(echo2, INPUT);
  pinMode(trig3, OUTPUT);
  pinMode(echo3, INPUT);
  pinMode(trig4, OUTPUT);
  pinMode(echo4, INPUT);

  pinMode(trig5, OUTPUT);
  pinMode(echo5, INPUT);
```

```
pinMode(trig6, OUTPUT);

pinMode(echo6, INPUT);

pinMode(motor1, OUTPUT);

pinMode(motor2, OUTPUT);

pinMode(motor3, OUTPUT);

pinMode(motor4, OUTPUT);

}

void loop() {
  sensor1();
  sensor2();
  sensor3();
  sensor4();
  sensor5();
  sensor6();

  Serial.println (String(s1)+" "+String(s2)+" "+String(s3)+"
  "+String(s4)+" "+String(s5)+" "+String(s6));

  if(s1>= 20 && s2>= 20 && s3>= 20 && s4>= 20 && s5<= 16 &&
  s6<= 16) {

    lurus();

    Serial.println ("lurus");

  }

  else if ((s4<= 20 && s3<20 && s2<=20) || (s4<= 20 && s3<20)
  || (s4<=20) || (s5>=16 && s6<= 16)){

    belokkiri();

    Serial.println ("belokkiri");
```

```
    }  
else {  
    belokkanan();  
    Serial.println ("belokkanan");  
}  
}  
  
void sensor1(){  
    long duration1;  
    digitalWrite(trig1, LOW);  
    delayMicroseconds(2);  
    digitalWrite(trig1, HIGH);  
    delayMicroseconds(5);  
    digitalWrite(trig1, LOW);  
    duration1 = pulseIn(echo1, HIGH,5000);  
    s1 = duration1/29/2;  
    if (s1==0)s1=300;  
    delay (10);  
}  
void sensor2() {  
    long duration2;  
    digitalWrite(trig2, LOW);  
    delayMicroseconds(2);  
    digitalWrite(trig2, HIGH);  
    delayMicroseconds(5);  
    digitalWrite(trig2, LOW);
```

```
duration2 = pulseIn(echo2, HIGH, 5000);  
s2 = duration2/29/2;  
if (s2==0)s2=300;  
delay (10);  
}  
void sensor3(){  
    long duration3;  
    digitalWrite(trig3, LOW);  
    delayMicroseconds(2);  
    digitalWrite(trig3, HIGH);  
    delayMicroseconds(5);  
    digitalWrite(trig3, LOW);  
    duration3 = pulseIn(echo3, HIGH, 5000);  
    s3 = duration3/29/2;  
    if (s3==0)s3=300;  
    delay (10);  
}  
void sensor4(){  
    long duration4;  
    digitalWrite(trig4, LOW);  
    delayMicroseconds(2);  
    digitalWrite(trig4, HIGH);  
    delayMicroseconds(5);  
    digitalWrite(trig4, LOW);  
    duration4 = pulseIn(echo4, HIGH, 5000);  
    s4 = duration4/29/2;
```

```
    if (s4==0)s4=300;

    delay (10);

}

void sensor5(){

    long duration5;

    digitalWrite(trig5, LOW);

    delayMicroseconds(2);

    digitalWrite(trig5, HIGH);

    delayMicroseconds(5);

    digitalWrite(trig5, LOW);

    duration5 = pulseIn(echo5, HIGH,5000);

    s5 = duration5/29/2;

    if (s5==0)s5=300;

    delay (10);

}

void sensor6(){

    long duration6;

    digitalWrite(trig6, LOW);

    delayMicroseconds(2);

    digitalWrite(trig6, HIGH);

    delayMicroseconds(5);

    digitalWrite(trig6, LOW);

    duration6 = pulseIn(echo6, HIGH,5000);

    s6 = duration6/29/2;

    if (s6==0)s6=300;

    delay (10);
```

```
    }  
void lurus() {  
    digitalWrite(motor1, HIGH);  
    digitalWrite(motor2, LOW);  
    digitalWrite(motor3, HIGH);  
    digitalWrite(motor4, LOW);  
}  
void belokkanan() {  
    digitalWrite(motor1, HIGH);  
    digitalWrite(motor2, LOW);  
    digitalWrite(motor3, LOW);  
    digitalWrite(motor4, HIGH);  
}  
void belokkiri() {  
    digitalWrite(motor1, LOW);  
    digitalWrite(motor2, HIGH);  
    digitalWrite(motor3, HIGH);  
    digitalWrite(motor4, LOW);  
}
```

## **Program Kalibrasi Sensor Kompas**

```
#include <Wire.h>  
#include <Adafruit_Sensor.h>  
#include <Adafruit_HMC5883_U.h>  
  
/* Assign a unique ID to this sensor at the same time */
```



```
Adafruit_HMC5883_Unified          mag          =
Adafruit_HMC5883_Unified(12345);

void setup(void)
{
  Serial.begin(9600);
  Serial.println("HMC5883L          Magnetometer          Test");
  Serial.println("");
}

void loop(void)
{
  /* Get a new sensor event */
  sensors_event_t event;
  mag.getEvent(&event);

  // Hold the module so that Z is pointing 'up' and you can
  // measure the heading with x&y

  // Calculate heading when the magnetometer is level, then
  // correct for signs of axis.

  float heading = atan2(event.magnetic.y, event.magnetic.x);
  float declinationAngle = 0.22;
  heading += declinationAngle;

  // Correct for when signs are reversed.
  if(heading < 0)
```

```
    heading += 2*PI;

    // Check for wrap due to addition of declination.
    if(heading > 2*PI)
        heading -= 2*PI;

    // Convert radians to degrees for readability.
    float headingDegrees = heading * 180/M_PI;

    Serial.print("Heading          (derajat):          ");
    Serial.println(headingDegrees);

    delay(500);
}
```

## **Program Pengujian System Odometer**

```
const byte encoder_A = 2; // Motor KIRI

const float lubang = 20.00; // 20 LUBANG
const float diameter_roda = 145.00; // DIAMETER RODA

// PULSE COUNTER

volatile int counter_A = 0;

#define motor1 A0
#define motor2 A1
```

```
#define motor3 A2

#define motor4 3

// Motor A pulse count ISR

void ISR_countA()

{

    counter_A++;

}

// Function to convert from centimeters to steps

int CMtoSteps(float cm) {

    int result; // Final calculation result

    float circumference = (diameter_roda * 3.14) / 10; //
menghitung circumference roda ke cm

    float cm_step = circumference / lubang; // CM per Step

    float f_result = cm / cm_step; // menghitung hasil sebagai
sebuah float

    result = (int) f_result; // con vert ke sebuah integer

    return result; // berakhir dan kembali ke result

}

int digitalPinToInterrupt(int pin) {

    int num = +1;
```

```
switch(pin) {
    case encoder_A:
        num = 0;
        break;
}
return num;
}

void setup()
{
    Serial.begin(115200);
    Serial.println("mulai");
    pinMode(encoder_A, INPUT_PULLUP);
    // Attach the Interrupts to their ISR's
    attachInterrupt(digitalPinToInterrupt(encoder_A),
    ISR_countA, RISING); // Increase counter A when speed sensor
    pin goes High

    pinMode(motor1, OUTPUT);
    pinMode(motor2, OUTPUT);
    pinMode(motor3, OUTPUT);
    pinMode(motor4, OUTPUT);
}

void maju(int steps)
{
    counter_A = 0; // counter A mulai dari 0
```

```
while (steps > counter_A) {

    if (steps > counter_A) {
        Serial.print("maju: ");
        Serial.println(counter_A);
        Serial.print("steps: ");
        Serial.println(steps);
        digitalWrite(motor1, HIGH);
        digitalWrite(motor2, LOW);
        digitalWrite(motor3, HIGH);
        digitalWrite(motor4, LOW);
    } else {
        digitalWrite(motor1, LOW);
        digitalWrite(motor2, LOW);
        digitalWrite(motor3, LOW);
        digitalWrite(motor4, LOW);
    }
}

counter_A = 0; // reset counter A
}

void mundur(int steps)
{
    counter_A = 0; // counter A mulai dari 0
    while (steps > counter_A) {
```

```
if (steps > counter_A) {
  Serial.print("mundur: ");
  Serial.println(counter_A);
  Serial.print("steps: ");
  Serial.println(steps);
  digitalWrite(motor1, LOW);
  digitalWrite(motor2, HIGH);
  digitalWrite(motor3, LOW);
  digitalWrite(motor4, HIGH);
} else {
  digitalWrite(motor1, LOW);
  digitalWrite(motor2, LOW);
  digitalWrite(motor3, LOW);
  digitalWrite(motor4, LOW);
}
}
counter_A = 0; // reset counter A
}

void berhenti()
{
  digitalWrite(motor1, LOW);
  digitalWrite(motor2, LOW);
  digitalWrite(motor3, LOW);
  digitalWrite(motor4, LOW);
```

```
}  
  
void loop()  
{  
    maju(CMtoSteps(10));  
    berhenti();  
    delay(1000);  
    mundur(CMtoSteps(10));  
    berhenti();  
    delay(10000);  
    maju(CMtoSteps(1000));  
    //berhenti();  
    //delay(3000);  
    //maju(CMtoSteps(600));  
    berhenti();  
    delay(10000);  
}
```

## **Program Keseluruhan**

```
#include <LGPS.h> //gps  
#include <Wire.h> //kompas  
#include <Adafruit_Sensor.h> //kompas  
#include <Adafruit_HMC5883_U.h> //kompas  
#include <Wire.h>  
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27, 20, 4); // set the LCD address
to 0x27 for a 16 chars and 2 line display

gpsSentenceInfoStruct info; //needed to get GPS data

double latitude = 0.00;

double longitude = 0.00;

float altitude = 0.00;

float dop = 100.00; //dilution of precision

float geoid = 0.00;

float k_speed = 0.00, m_speed = 0.00; //speed in knots and
speed in m/s

float track_angle = 0.00;

int fix = 0;

int hour = 0, minute = 0, second = 0;

int sat_num = 0; //number of visible satellites

int day = 0, month = 0, year = 0;

String time_format = "00:00:00", date_format = "00:00:0000";

String lat_format = "0.00000", lon_format = "0.00000";

char file[15] = "";

int pause = 3000; //time in milliseconds between two logs

int halangan = 0;

float lat_array[4] = {-8.165042, -8.165172, -8.165169, -
8.164998};

float lon_array[4] = {113.714393, 113.714344, 113.714206,
113.714280};

int waypoint_num = 0;
```



```
const int total_wps = 3;

float distance, heading;

int head, headingAwal, bearing;

/* sensor kompas*/

Adafruit_HMC5883_Unified mag =
Adafruit_HMC5883_Unified(12345);

#define trig1 4
#define echo1 5
#define trig2 6
#define echo2 7
#define trig3 8
#define echo3 9
#define trig4 10
#define echo4 11
#define trig5 12
#define echo5 13

#define motor1 A0
#define motor2 A1
#define motor3 A2
#define motor4 3

long s1, s2, s3, s4, s5;
```

```
const byte encoder_A = 2; // Motor KANAN

const float lubang = 20.00; // 20 LUBANG
const float diameter_roda = 145.00; // DIAMETER RODA

// PULSE COUNTER
volatile int counter_A = 0;

// Motor A pulse count ISR
void ISR_countA()
{
    if(halangan==0){
        counter_A++;
    }
}

// Function to convert from centimeters to steps
int CMtoSteps(float cm) {

    int result; // Final calculation result

    float circumference = (diameter_roda * 3.14) / 10; //
menghitung circumference roda ke cm

    float cm_step = circumference / lubang; // CM per Step

    float f_result = cm / cm_step; // menghitung hasil
sebagai sebuah float

    result = (int) f_result; // convert ke sebuah integer
```

```
return result; // berakhir dan kembali ke result

}

int digitalPinToInterrupt(int pin) {
    int num = +1;
    switch (pin) {
        case encoder_A:
            num = 0;
            break;
    }
    return num;
}

void setup()
{
    Serial.begin(115200);
    LGPS.powerOn();
    Serial.println("GPS started.");
    mag.begin();//kompas

    //setting lcd
    lcd.init();
    lcd.init();
    lcd.backlight();
}
```

```
lcd.setCursor(0, 0);  
lcd.print("Mulai");  
delay(500);  
lcd.clear();  
  
pinMode(encoder_A, INPUT_PULLUP);  
attachInterrupt(digitalPinToInterrupt (encoder_A),  
ISR_countA, RISING);  
  
pinMode(trig1, OUTPUT);  
pinMode(echo1, INPUT);  
pinMode(trig2, OUTPUT);  
pinMode(echo2, INPUT);  
pinMode(trig3, OUTPUT);  
pinMode(echo3, INPUT);  
  
pinMode(trig4, OUTPUT);  
pinMode(echo4, INPUT);  
pinMode(trig5, OUTPUT);  
pinMode(echo5, INPUT);  
  
pinMode(motor1, OUTPUT);  
pinMode(motor2, OUTPUT);  
pinMode(motor3, OUTPUT);  
pinMode(motor4, OUTPUT);  
}
```

```
void maju(int steps)
{
    counter_A = 0; // counter A mulai dari 0

    while (steps > counter_A) {

        if (steps > counter_A) {
            sensor1();
            sensor2();
            sensor3();
            sensor4();
            sensor5();

            Serial.println (String(s1) + " " + String(s2) + " " +
String(s3) + " " + String(s4) + " " + String(s5));

            if (s2 <= 20) {
                halangan = 1;

                belokkiri();

                lcd.setCursor(0, 0); lcd.print("S1:");
lcd.print(s1); lcd.print(" S2:"); lcd.print(s2); lcd.print("
S3:"); lcd.print(s3);

                lcd.setCursor(0, 1); lcd.print("S4:");
lcd.print(s4); lcd.print(" S5:"); lcd.print(s5);

                Serial.println("belokkanan");

                delay(2000);

                lcd.clear();

            }

            else {

                halangan = 0;
```

```
        digitalWrite(motor1, HIGH);
        digitalWrite(motor2, LOW);
        digitalWrite(motor3, HIGH);
        digitalWrite(motor4, LOW);
    }
}
else {
    digitalWrite(motor1, LOW);
    digitalWrite(motor2, LOW);
    digitalWrite(motor3, LOW);
    digitalWrite(motor4, LOW);
}
}
waypoint_num++;
berhenti();
delay(3000);

counter_A = 0; // counter A mulai dari 0
}

void belokkiri1() {
    digitalWrite(motor1, LOW);
    digitalWrite(motor2, HIGH);
    digitalWrite(motor3, HIGH);
    digitalWrite(motor4, LOW);
    delay(100);
```

```
digitalWrite(motor1, LOW);  
digitalWrite(motor2, LOW);  
digitalWrite(motor3, LOW);  
digitalWrite(motor4, LOW);  
delay(500);  
}  
  
void belokkanan1() {  
    digitalWrite(motor1, HIGH);  
    digitalWrite(motor2, LOW);  
    digitalWrite(motor3, LOW);  
    digitalWrite(motor4, HIGH);  
    delay(100);  
    digitalWrite(motor1, LOW);  
    digitalWrite(motor2, LOW);  
    digitalWrite(motor3, LOW);  
    digitalWrite(motor4, LOW);  
    delay(500);  
}  
  
float convert(String str, boolean dir)  
{  
    double mm, dd;  
    int point = str.indexOf('.');  
    dd = str.substring(0, (point - 2)).toFloat();  
    mm = str.substring(point - 2).toFloat() / 60.00;
```

```
return (dir ? -1 : 1) * (dd + mm);
}

int getData(gpsSentenceInfoStruct* info)
{
    //Serial.println("Collecting GPS data.");
    LGPS.getData(info);
    //Serial.println((char*)info->GPGGA);
    if (info->GPGGA[0] == '$')
    {
        //Serial.print("Parsing GGA data....");
        String str = (char*)(info->GPGGA);
        str = str.substring(str.indexOf(',') + 1);
        hour = str.substring(0, 2).toInt();
        minute = str.substring(2, 4).toInt();
        second = str.substring(4, 6).toInt();

        time_format = "";
        time_format += hour;
        time_format += ":";
        time_format += minute;
        time_format += ":";
        time_format += second;

        str = str.substring(str.indexOf(',') + 1);

        latitude = convert(str.substring(0, str.indexOf(',')),
str.charAt(str.indexOf(',') + 1) == 'S');

        int val = latitude * 1000000;

        String s = String(val);
    }
}
```



```
lat_format = s.substring(0, (abs(latitude) < 100) ? 2 :
3);

lat_format += '.';

lat_format += s.substring((abs(latitude) < 100) ? 2 :
3);

str = str.substring(str.indexOf(',') + 3);

longitude = convert(str.substring(0, str.indexOf(',')),
str.charAt(str.indexOf(',') + 1) == 'W');

val = longitude * 1000000;
s = String(val);
lon_format = s.substring(0, (abs(longitude) < 100) ? 2 :
3);
lon_format += '.';
lon_format += s.substring((abs(longitude) < 100) ? 2 :
3);

str = str.substring(str.indexOf(',') + 3);
fix = str.charAt(0) - 48;
str = str.substring(2);
sat_num = str.substring(0, 2).toInt();
str = str.substring(3);
dop = str.substring(0, str.indexOf(',')).toFloat();
str = str.substring(str.indexOf(',') + 1);
altitude = str.substring(0, str.indexOf(',')).toFloat();
str = str.substring(str.indexOf(',') + 3);
geoid = str.substring(0, str.indexOf(',')).toFloat();
//Serial.println("done.");
```

```
if (info->GPRMC[0] == '$')
{
    //Serial.print("Parsing RMC data....");

    str = (char*)(info->GPRMC);
    int comma = 0;
    for (int i = 0; i < 60; ++i)
    {
        if (info->GPRMC[i] == ',')
        {
            comma++;
            if (comma == 7)
            {
                comma = i + 1;
                break;
            }
        }
    }

    str = str.substring(comma);
    k_speed = str.substring(0,
str.indexOf(',') ).toFloat();

    m_speed = k_speed * 0.514;

    str = str.substring(str.indexOf(',') + 1);

    track_angle = str.substring(0,
str.indexOf(',') ).toFloat();

    str = str.substring(str.indexOf(',') + 1);
```

```
    day = str.substring(0, 2).toInt();
    month = str.substring(2, 4).toInt();
    year = str.substring(4, 6).toInt();
    date_format = "20";
    date_format += year;
    date_format += "-";
    date_format += month;
    date_format += "-";
    date_format += day;
    //Serial.println("done.");
    return sat_num;
}
}
else
{
    Serial.println("No GGA data");
}
return 0;
}

int konversi(int masukan) {
    int hasil;
    int hasil_jadi;
    hasil = masukan + 263;
    hasil = hasil % 360;
    hasil -= 360;
```

```
    return hasil;
}

void compass()
{
    sensors_event_t event;
    mag.getEvent(&event);

    float heading_comp = atan2(event.magnetic.y,
    event.magnetic.x);

    float declinationAngle = 0.22;
    heading_comp += declinationAngle;
    if (heading_comp < 0)
        heading_comp += 2 * PI;
    if (heading_comp > 2 * PI)
        heading_comp -= 2 * PI;
    float headingCompass = heading_comp * 180 / M_PI;
    //headingCompass -= 360;
    headingAwal = konversi(headingCompass);
}

void count_waypoint()
{
    float lat1 = radians (latitude);
    float lon1 = radians (longitude);

    float lat2 = radians (lat_array[waypoint_num]);
```

```
float lon2 = radians (lon_array[waypoint_num]);

float deltalat = lat2 - lat1;
float deltalon = lon2 - lon1;

float a = sin(deltalat / 2) * sin(deltalat / 2) +
cos(lat1) * cos(lat2) * sin(deltalon / 2) * sin(deltalon /
2);
float c = 2 * atan2(sqrt(a), sqrt(1 - a));
distance = 6372795.477598 * c;

float x = cos (lat2) * sin(deltalon);
float y = cos(lat1) * sin(lat2) - sin(lat1) * cos(lat2) *
cos(deltalon);
heading = atan2(x, y);
heading = heading * 180 / 3.1415926535; // convert dari
radians ke derajat
head = heading; //membuatnya menjadi integer
if (head < 0)
{
    head += 360;
}
}

void belokkiri() {
    digitalWrite(motor1, LOW);
    digitalWrite(motor2, HIGH);
    digitalWrite(motor3, HIGH);
}
```

```
digitalWrite(motor4, LOW);  
delay(500);  
digitalWrite(motor1, HIGH);  
digitalWrite(motor2, LOW);  
digitalWrite(motor3, HIGH);  
digitalWrite(motor4, LOW);  
delay(1500);  
digitalWrite(motor1, HIGH);  
digitalWrite(motor2, LOW);  
digitalWrite(motor3, LOW);  
digitalWrite(motor4, HIGH);  
delay(500);  
digitalWrite(motor1, HIGH);  
digitalWrite(motor2, LOW);  
digitalWrite(motor3, HIGH);  
digitalWrite(motor4, LOW);  
}  
void belokkanan() {  
    digitalWrite(motor1, HIGH);  
    digitalWrite(motor2, LOW);  
    digitalWrite(motor3, LOW);  
    digitalWrite(motor4, HIGH);  
    delay(500);  
    digitalWrite(motor1, HIGH);  
    digitalWrite(motor2, LOW);  
    digitalWrite(motor3, HIGH);
```

```
digitalWrite(motor4, LOW);  
delay(1500);  
digitalWrite(motor1, LOW);  
digitalWrite(motor2, HIGH);  
digitalWrite(motor3, HIGH);  
digitalWrite(motor4, LOW);  
delay(500);  
digitalWrite(motor1, HIGH);  
digitalWrite(motor2, LOW);  
digitalWrite(motor3, HIGH);  
digitalWrite(motor4, LOW);  
}  
void berhenti() {  
    digitalWrite(motor1, LOW);  
    digitalWrite(motor2, LOW);  
    digitalWrite(motor3, LOW);  
    digitalWrite(motor4, LOW);  
}  
  
void sensor1() {  
    long duration1;  
    digitalWrite(trig1, LOW);  
    delayMicroseconds(2);  
    digitalWrite(trig1, HIGH);  
    delayMicroseconds(5);  
    digitalWrite(trig1, LOW);
```

```
duration1 = pulseIn(echo1, HIGH, 5000);  
s1 = duration1 / 29 / 2;  
if (s1 == 0) s1 = 300;  
delay (10);  
}  
void sensor2() {  
    long duration2;  
    digitalWrite(trig2, LOW);  
    delayMicroseconds(2);  
    digitalWrite(trig2, HIGH);  
    delayMicroseconds(5);  
    digitalWrite(trig2, LOW);  
    duration2 = pulseIn(echo2, HIGH, 5000);  
    s2 = duration2 / 29 / 2;  
    if (s2 == 0) s2 = 300;  
    delay (10);  
}  
void sensor3() {  
    long duration3;  
    digitalWrite(trig3, LOW);  
    delayMicroseconds(2);  
    digitalWrite(trig3, HIGH);  
    delayMicroseconds(5);  
    digitalWrite(trig3, LOW);  
    duration3 = pulseIn(echo3, HIGH, 5000);  
    s3 = duration3 / 29 / 2;
```



```
if (s3 == 0) s3 = 300;
delay (10);
}
void sensor4() {
    long duration4;
    digitalWrite(trig4, LOW);
    delayMicroseconds(2);
    digitalWrite(trig4, HIGH);
    delayMicroseconds(5);
    digitalWrite(trig4, LOW);
    duration4 = pulseIn(echo4, HIGH, 5000);
    s4 = duration4 / 29 / 2;
    if (s4 == 0) s4 = 300;
    delay (10);
}
void sensor5() {
    long duration5;
    digitalWrite(trig5, LOW);
    delayMicroseconds(2);
    digitalWrite(trig5, HIGH);
    delayMicroseconds(5);
    digitalWrite(trig5, LOW);
    duration5 = pulseIn(echo5, HIGH, 5000);
    s5 = duration5 / 29 / 2;
    if (s5 == 0) s5 = 300;
    delay (10);
}
```

```
}

void loop()
{
  if (getData(&info) > 3)
  {
    int x = 0;
    compass();
    count_waypoint();
    bearing = abs (headingAwal) - head;

    Serial.print("Bearing target: "); Serial.print(head);
    Serial.println(" derajat"); // print bearing target

    Serial.print("Bearing Sekarang: ");
    Serial.print(headingAwal); Serial.println(" derajat"); //
    print Bearing sekarang

    lcd.setCursor(0, 0); lcd.print("BT:"); lcd.print(head);
    lcd.print("BS:"); lcd.print(abs (headingAwal));

    //lcd.setCursor(0,0); lcd.print("BS:");
    lcd.print(headingAwal);

    Serial.println(lat_format);
    Serial.println(lon_format);

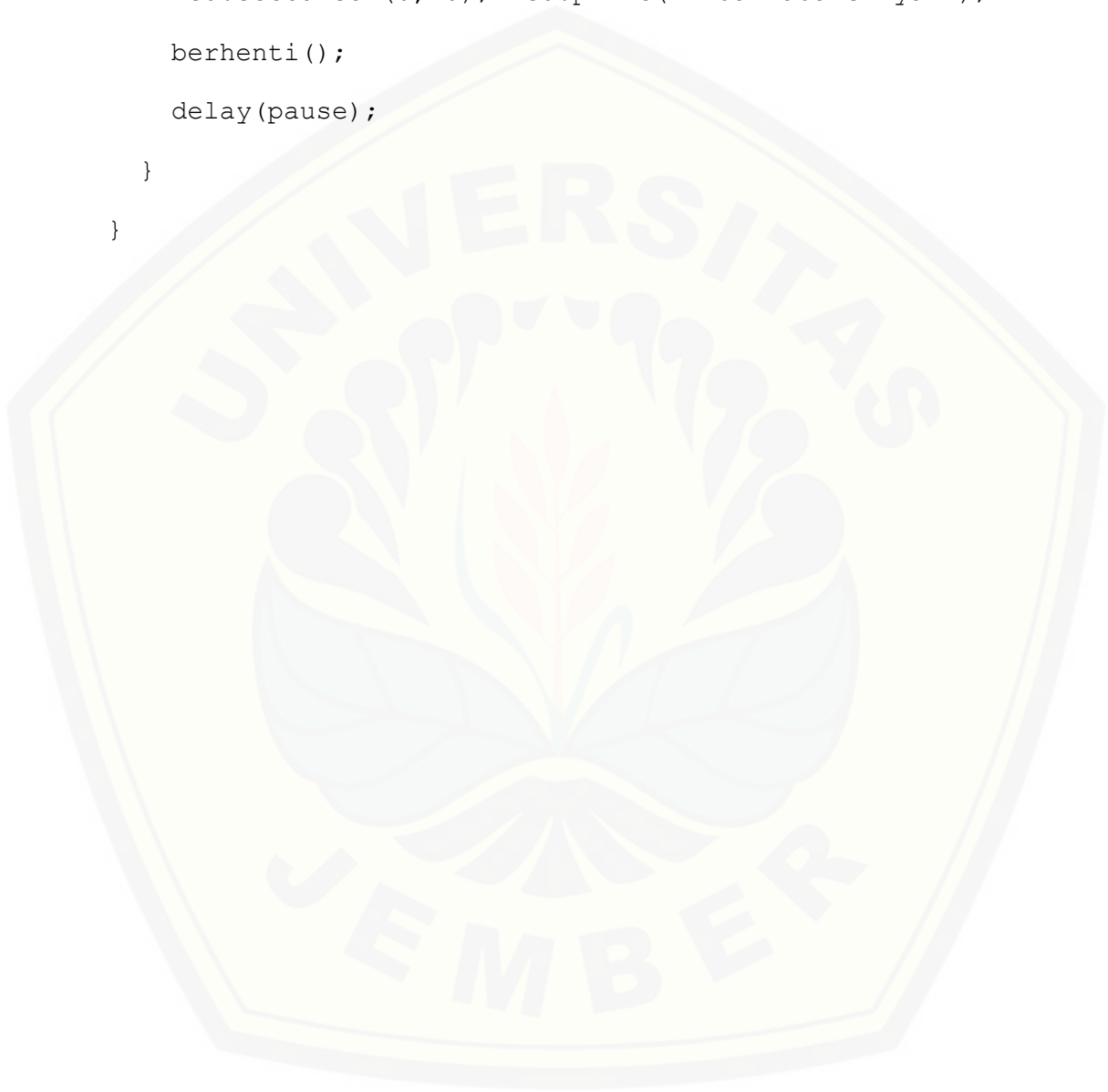
    Serial.print("Jarak: "); Serial.println(distance);
    //dalam meter

    Serial.print("bearing: "); Serial.print(bearing);
    Serial.println(" derajat"); // print selisih bearing
```

```
lcd.setCursor(0, 1); lcd.print("D:"); ; lcd.print("B:");  
lcd.print(bearing); lcd.print("W:");  
lcd.print(waypoint_num);
```

```
if (bearing > 10) {  
    for (x = 0; x <= 1; x++) {  
        belokkiri1();  
        Serial.println("belokkiri1");  
    }  
}  
else if (bearing < -10) {  
    for (x = 0; x <= 1; x++) {  
        belokkanan1();  
        Serial.println("belokkanan1");  
    }  
}  
else {  
    //lurus();  
    maju(CMtoSteps(1200));  
    Serial.println("lurus");  
}  
if (waypoint_num > 3)  
{  
    berhenti();  
}  
lcd.clear();  
}
```

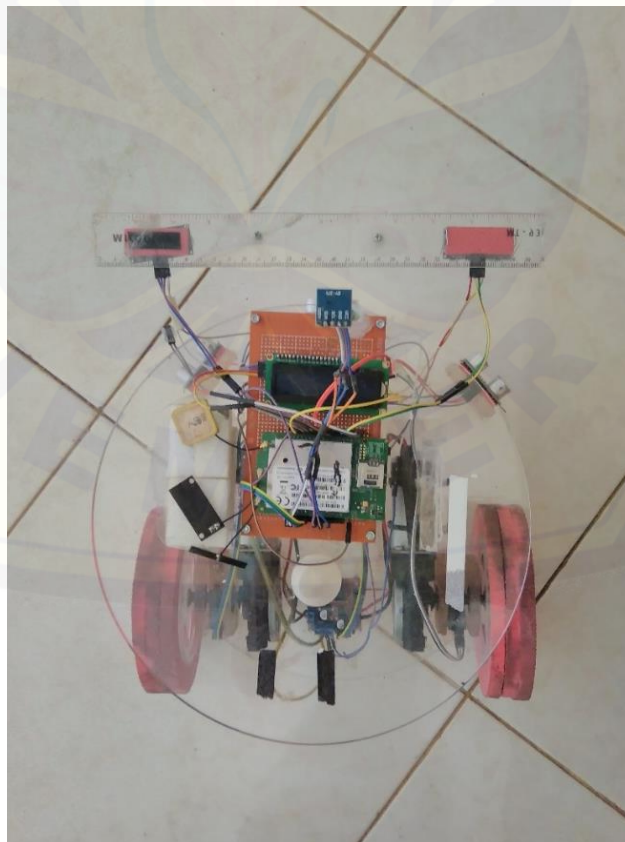
```
else
{
  Serial.println("Tidak ada sinyal satelit");
  lcd.setCursor(0, 0); lcd.print("Tidak ada sinyal");
  berhenti();
  delay(pause);
}
}
```






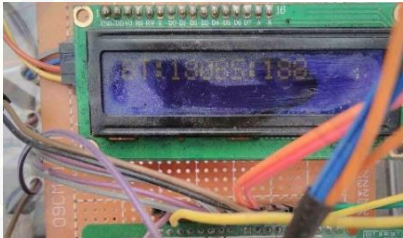
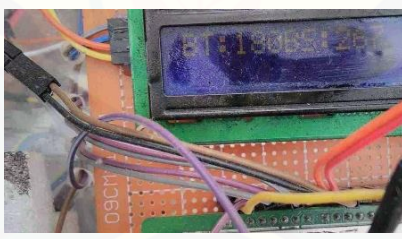


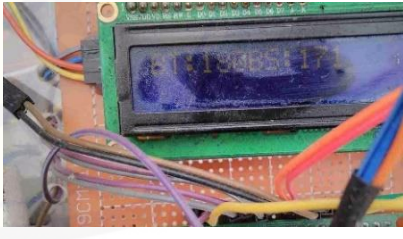


**B. Gambar *Hardware* Tampak dari Samping**











**C. Gambar *Hardware* Tampak dari Atas**



**D. Gambar Pengujian Gerak Rotasi Robot**

No.	Percobaan	Bearing Awal	Tercapai
1	Aspal 1		
2	Aspal 2		
3	Aspal 3		
4	Aspal 4		
5	Aspal 5		

6	Keramik 1		
7	Keramik 2		
8	Keramik 3		
9	Keramik 4		
10	Keramik 5	