



**IMPLEMENTASI *SYSTEM VOICE RECOGNITION* DAN *ROTARY ENCODER* PADA *MOBILE ROBOT* SEBAGAI SISTEM NAVIGASI DAN PERHITUNGAN POSISI ROBOT**

**SKRIPSI**

oleh

**Septian Dwi Ermansyah  
NIM 111910201099**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2016**



**IMPLEMENTASI *SYSTEM VOICE RECOGNITION* DAN *ROTARY ENCODER* PADA *MOBILE ROBOT* SEBAGAI SISTEM NAVIGASI DAN PERHITUNGAN POSISI ROBOT**

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Program Studi Teknik Elektro strata satu (S1) pada Fakultas Teknik

oleh

**Septian Dwi Ermansyah  
NIM 111910201099**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2016**

## **PERSEMBAHAN**

Dengan naungan rahmat Allah S.W.T. yang telah memberikan segala nikmat serta anugerah-Nya sehingga dapat kupersembahkan skripsi ini kepada :

1. Kedua orang tua saya Ibu Dwi Titik W. dan Bapak Irmanudin yang telah mendidik dengan penuh cinta dan kasih sayang hingga memperoleh gelar sarjana teknik (ST).
2. Kakak saya Novi Ermantika Sari yang selalu memberikan dorongan agar segera menyelesaikan skripsi ini;
3. Bapak Pembimbing saya Bambang Supeno, S.T., M.T. dan Widjonarko, AMd., S.T., M.T terima kasih atas bimbingan dan telah berbagi ilmu serta nasihatnya;
4. Ibu/Bapak Guru TK Muslimat NU Ketegan Sidoarjo, SDN Ketegan 1 Sidoarjo, SMPN 2 Taman Sidoarjo, SMA Kemala Bhayangkari 1 Surabaya, serta Ibu/Bapak Dosen Fakultas Teknik Universitas Jember;
5. Almamater tercinta Fakultas Teknik Universitas Jember.
6. Serta seluruh pihak yang telah membantu penulis untuk menyelesaikan skripsi ini yang tidak bias disebutkan satu persatu.

## MOTTO

*Kehidupan ini memang tidak akan sepenuhnya sempurna, tapi sikap baik anda bias menjadikannya terbaik dari yang bias anda hidupi. Dan itu sudah sempurna. (Mario Teguh)*

*Jangan pernah salahkan orang lain karena mengecewakan kamu. Salahkanlah diri kamu sendiri karena berharap terlalu banyak kepada mereka. (H.M. Scharde)*

*Tidak penting apaun agama atau sukumu. Kalau kamu bisa melakukan sesuatu yang baik untuk semua orang, orang tidak pernah Tanya apa agamamu. (K.H. Abdurrahman Wahid)*

## PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Septian Dwi Ermansyah

NIM : 111910201099

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “Implementasi Sistem *Voice Recognition* dan *Rotary Encoder* pada *Mobile Robot* sebagai Sistem Navigasi dan Perhitungan Posisi Robot” adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan. Saya bertanggung jawab 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, 27 Juni 2016  
Yang menyatakan,

Septian Dwi Ermansyah  
NIM 111910201099

## **SKRIPSI**

### **IMPLEMENTASI *SYSTEM VOICE RECOGNITION* DAN *ROTARY ENCODER* PADA *MOBILE ROBOT* SEBAGAI SISTEM NAVIGASI DAN PERHITUNGAN POSISI ROBOT**

Oleh

Septian Dwi Ermansyah  
NIM 111910201099

Pembimbing

Dosen Pembimbing Utama : Bambang Supeno S.T., M.T.

Dosen Pembimbing Anggota : Widjonarko, AMd., S.T., M.T.

## PENGESAHAN

Skripsi berjudul “Implementasi Sistem *Voice Recognition* dan *Rotary Encoder* pada *Mobile Robot* sebagai Sistem Navigasi dan Perhitungan Posisi Robot” telah diuji dan disahkan pada :

Hari :  
Tanggal :  
Tempat : Fakultas Teknik Universitas Jember

Tim Penguji:

Pembimbing Anggota,

Pembimbing Utama,

Widjonarko, AMd., S.T., M.T.

NIP. 19710908 199903 1 001

Bambang Supeno, S.T., M.T.

NIP. 19690630 199512 1 001

Penguji I,

Penguji II,

Sumardi, S.T., M.T.

NIP. 19670113 199802 1 001

Satryo Budi Utomo, S.T., M.T.

NIP. 1985026 200801 1 002

Mengesahkan,

Dekan Fakultas Teknik

Dr. Ir. Entin Hidayah M.U.M

NIP. 19661215 199503 2 001

# **IMPLEMENTASI *SYSTEM VOICE RECOGNITION* DAN *ROTARY ENCODER* PADA *MOBILE ROBOT* SEBAGAI SISTEM NAVIGASI DAN PERHITUNGAN POSISI ROBOT**

**Septian Dwi Ermansyah**

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

## **ABSTRAK**

Sistem kontrol robot pada umumnya berbentuk *joystick* atau *remote control* bertombol sehingga membutuhkan *hardware* tambahan untuk sistem kontrol robot. Berkembangnya teknologi *smartphone* android dapat mempermudah untuk membuat aplikasi sistem kontrol robot. Tujuan penelitian ini adalah mengontrol robot dengan *smartphone* dengan sistem *voice recognition* dan penerapan *rotary encoder* sebagai sensor pengukur jarak tempuh robot dan sensor kompas sebagai navigasi arah gerak robot. Teknologi *voice recognition* merupakan salah satu teknologi biometrika yang tidak memerlukan peralatan khusus dan relatif murah. Suara merupakan alat komunikasi yang unik dan dapat dibedakan dengan mudah. Pengujian robot dilakukan dengan 12 perintah dengan masing-masing perintah 6 kali percobaan. Nilai waktu respon sebesar 5 detik dengan *error X* sebesar 2,15cm dan *error Y* sebesar 1,37cm dengan *error* persen posisi robot sebesar 3,54%. Robot ini berhasil dan dapat digunakan dengan ketepatan 96,46%.

Kata kunci: *smartphone* android, *voice recognition*, *rotary encoder*, sensor kompas, biometri



# **IMPLEMENTATION VOICE RECOGNITION SYSTEM AND ROTARY ENCODER IN MOBILE ROBOT NAVIGATION SYSTEM AND CALCULATION AS A ROBOT POSITION**

**Septian Dwi Ermansyah**

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

## **ABSTRACT**

*Robot control system is generally shaped joystick or push button remote control thus requiring additional hardware for robotic control systems. The development of technology can simplify the android smartphone to make a robot control system applications. The purpose of this research is to control the robot with a smartphone with voice recognition systems and the application of rotary encoder as distance measuring sensor and sensor robot compass as a navigational direction of motion of the robot. Voice recognition technology is a biometric technology that requires no special equipment and is relatively inexpensive. Sound is a unique communication instrument and can be distinguished easily. Tests carried out by the robot 12 commands with each command six tries. The value of the response time by 5 seconds with error of 2,15cm X and Y for 1,37cm error with error percent position of the robot by 3.54%. This robot successfully and can be used with 96.46% accuracy.*

**Keywords:** *android smartphone, voice recognition, rotary encoder, compass sensor, biometry*

## RINGKASAN

**Implementasi *System Voice Recognition* dan *Rotary Encoder* pada *Mobile Robot* sebagai Sistem Navigasi dan Perhitungan Posisi Robot;** Septian Dwi Ermansyah; 111910201099; 2016; 53 halaman; Program Studi Strata Satu Teknik, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember.

Sistem kontrol robot pada umumnya berbentuk *joystick* atau *remote control* bertombol sehingga membutuhkan *hardware* tambahan untuk sistem kontrol robot. Berkembangnya teknologi *smartphone* android dapat mempermudah untuk membuat aplikasi sistem kontrol robot. Teknologi *smartphone* tersendiri sudah berkembang pesat di dunia dan berbagai macam versi sudah dirilis.

*Smartphone* juga mendukung fasilitas *voice recognition* yang dapat memungkinkan robot dikontrol dengan aplikasi *voice recognition* pada *smartphone*. Untuk koneksi antara *smartphone* dan robot menggunakan *bluetooth*. Perintah suara akan mengendalikan robot ke tempat yang akan dituju dengan bantuan sensor *rotary encoder* dan sensor kompas pada robot untuk membantu sistem navigasi robot menuju tempat yang dituju.

Teknologi *voice recognition* merupakan salah satu teknologi biometrika yang tidak memerlukan peralatan khusus dan relatif murah. Suara merupakan alat komunikasi yang unik dan dapat dibedakan dengan mudah. Tujuan penelitian ini adalah dapat mengontrol robot menggunakan *smartphone* android dengan suara dan dapat menerapkan sensor *rotary encoder* dan sensor kompas sebagai sistem navigasi robot agar menuju posisi yang ditentukan dengan tepat.

Dalam keseluruhan sistem yang dirancang dengan baik sesuai dengan harapan yang dibuktikan dengan waktu respon robot terhadap perintah rata-rata selama 5 detik dan *error* persen rata-rata ketepatan posisi robot hanya sebesar 3,54%. Waktu respon robot terhadap perintah sangat bergantung pada kecepatan jaringan internet pada *smartphone* sedangkan untuk ketepatan posisi robot sangat bergantung pada ketepatan pembacaan sensor kompas dan sensor *rotary encoder*.

## PRAKATA

Puji syukur atas kehadiran Tuhan Yang Maha Esa, atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Implementasi Sistem *Voice Recognition* dan *Rotary Encoder* pada *Mobile Robot* sebagai Sistem Navigasi dan Perhitungan Posisi Robot”. Skripsi ini disusun guna memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan beberapa pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Ibu Dr. Ir. Entin Hidayah M.U.M selaku Dekan Fakultas Teknik, Universitas Jember;
2. Bapak Dr. Bambang Sri Kaloko, S.T., M.T. selaku Ketua Jurusan Teknik Elektro, Universitas Jember;
3. Bapak Dedy Kurnia Setiawan, ST., MT. selaku Ketua Program Studi Strata-1 Teknik Elektro, Universitas Jember dan dosen pembimbing akademik saya;
4. Bapak Bambang Supeno, S.T., M.T. dan Bapak Widjonarko, AMd., S.T., M.T. selaku Dosen Pembimbing, yang senantiasa sabar dan meluangkan waktu, pikiran, dan perhatiannya dalam membimbing saya, serta tak henti-hentinya memberikan motivasi, dorongan dan semangat kepada saya dalam menyelesaikan skripsi ini;
5. Bapak Sumardi, S.T., M.T., selaku penguji pertama dan Bapak Satryo Budi Utomo, S.T., M.T. selaku penguji kedua yang telah memberikan saran dan waktu;
6. Keluarga besar yang selama ini memotivasi dan memberikan dukungan secara moral, spiritual, dan material terutama kepada kedua orang tua saya dan kakak saya;
7. Keluarga besar MM Jember terutama Mbak Zila, Aluk, Hesta, Mbak Alil, Mas Mais dkk;

8. Keluarga besar Teknik Elektro 2011 yang selalu berjuang bersama menghadapi setiap masalah dan kemenangan;
9. Sahabat kebersamaan Wildan, Deta, Putra, Miko, Reno, Jarwo, Norma, Rara dan Andini;
10. Keluarga kontrakan dan kosan Sumur Bor 99X;
11. Semua pihak yang telah membantu memberikan bantuan dan semangat dalam menyelesaikan kripsi ini terutama Abdul Fajar.

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

Jember, 27 Juni 2016

Penulis

## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	i
<b>HALAMAN PERSEMBAHAN</b> .....	ii
<b>HALAMAN MOTTO</b> .....	iii
<b>HALAMAN PERNYATAAN</b> .....	iv
<b>HALAMAN PEMBIMBINGAN</b> .....	v
<b>HALAMAN PENGESAHAN</b> .....	vi
<b>ABSTRACK</b> .....	vii
<b>RINGKASAN</b> .....	ix
<b>PRAKATA</b> .....	x
<b>DAFTAR ISI</b> .....	xii
<b>DAFTAR GAMBAR</b> .....	xv
<b>DAFTAR TABEL</b> .....	xvii
<b>BAB 1. PENDAHULUAN</b> .....	1
<b>1.1 Latar Belakang</b> .....	1
<b>1.2 Rumusan Masalah</b> .....	2
<b>1.3 Tujuan</b> .....	2
<b>1.4 Manfaat</b> .....	2
<b>1.5 Batasan Masalah</b> .....	3
<b>1.6 Sistematika Penelitian</b> .....	3
<b>BAB 2. TINJAUAN PUSTAKA</b> .....	4
<b>2.1 Arduino UNO R3</b> .....	4
<b>2.2 Android</b> .....	5
<b>2.3 Bluetooth Module HC-05</b> .....	6
<b>2.4 LCD 16x2</b> .....	8
<b>2.5 Driver Motor</b> .....	10
<b>2.6 Sensor Rotary Encoder</b> .....	12
<b>2.7 Modul Kompas</b> .....	17

<b>2.8 App. Inventor</b> .....	19
<b>2.9 Teori Odometry</b> .....	20
<b>BAB 3. METODOLOGI PENELITIAN</b> .....	23
<b>3.1 Waktu dan Tempat Penelitian</b> .....	23
3.1.1 Tempat Penelitian .....	23
3.1.2 Waktu Penelitian .....	23
<b>3.2 Prosedur Penelitian</b> .....	23
<b>3.3 Alat dan Bahan</b> .....	24
3.3.1 Alat .....	24
3.3.2 Bahan .....	24
3.3.3 Software .....	25
<b>3.4 Blok Diagram</b> .....	25
<b>3.5 Desain Penelitian</b> .....	26
<b>3.6 Flowchart</b> .....	29
<b>3.7 Perancangan Sistem</b> .....	30
3.7.1 Spesifikasi Motor .....	30
3.7.2 Rangkaian Driver Motor .....	31
3.7.3 Rangkaian Modul Bluetooth .....	32
3.7.4 Rangkaian Encoder .....	33
3.7.5 Rangkaian LCD .....	33
3.7.6 Sistem Android .....	34
3.7.7 Pola Lintasan .....	35
<b>3.8 Pengujian Sistem</b> .....	36
3.8.1 Pengujian Sensor Rotary Encoder .....	36
3.8.2 Pengujian Sensor Kompas .....	39
3.8.3 Pengujian Voice Recognition Android ke Robot .....	40
3.8.4 Pengujian Komunikasi Bluetooth terhadap Jarak .....	42
<b>BAB 4. HASIL DAN PEMBAHASAN</b> .....	43
<b>4.1 Pengujian Keseluruhan Sistem</b> .....	43
4.1.1 Pengujian dari Pos Home menuju Pos Area 1 .....	43

4.1.2	Pengujian dari Pos Area 1 menuju Pos Home .....	44
4.1.3	Pengujian dari Pos Home menuju Pos Area 2 .....	45
4.1.4	Pengujian dari Pos Area 2 menuju Pos Home .....	45
4.1.5	Pengujian dari Pos Home menuju Pos Area 3 .....	46
4.1.6	Pengujian dari Pos Area 3 menuju Pos Home .....	47
4.1.7	Pengujian dari Pos Area 1 menuju Pos Area 2 .....	47
4.1.8	Pengujian dari Pos Area 2 menuju Pos Area 1 .....	48
4.1.9	Pengujian dari Pos Area 1 menuju Pos Area 3 .....	49
4.1.10	Pengujian dari Pos Area 3 menuju Pos Area 1 .....	49
4.1.11	Pengujian dari Pos Area 2 menuju Pos Area 3 .....	50
4.1.12	Pengujian dari Pos Area 3 menuju Pos Area 2 .....	51
4.1.13	Rata-Rata Hasil Pengujian dan Analisa .....	51
<b>BAB 5. PENUTUP</b>	.....	<b>53</b>
<b>5.1 Kesimpulan</b>	.....	<b>53</b>
<b>5.2 Saran</b>	.....	<b>53</b>
<b>DAFTAR PUSTAKA</b>	.....	<b>54</b>
<b>LAMPIRAN</b>	.....	<b>55</b>

## DAFTAR GAMBAR

	Halaman
Gambar 2.1. Modul Kontroler Arduino Mega 2560 .....	4
Gambar 2.2. Perkembangan Versi-Versi Android .....	6
Gambar 2.3. Modul <i>Bluetooth</i> HC-05.....	8
Gambar 2.4. Modul LCD ( <i>Liquid Cristal Display</i> ) .....	8
Gambar 2.5. Modul <i>Driver Motor</i> .....	11
Gambar 2.6. Konfigurasi PIN Driver Motor L298 .....	12
Gambar 2.7. Optical Rotary Encoder .....	13
Gambar 2.8. Rangkaian Tipikal Penghasil Pulsa pada <i>Rotary Encoder</i> .....	13
Gambar 2.9. Absolute Rotary Encoder .....	14
Gambar 2.10. Susunan Piringan untuk Incremental Encoder .....	15
Gambar 2.11. Sinyal Keluaran <i>Encoder</i> untuk Pengukuran Kecepatan dengan <i>Frequencymeter</i> .....	16
Gambar 2.12. Pengukuran Kecepatan dengan Menggunakan Periodimeter	17
Gambar 2.13. Sensor GY-273.....	19
Gambar 2.14. Logo App Inventor .....	19
Gambar 2.15. Ilustrasi pada sumbu Cartesian .....	21
Gambar 2.16. Sudut $\alpha$ , $\beta$ , dan $\gamma$ .....	21
Gambar 3.1. Blok Diagram .....	25
Gambar 3.2. Komponen Gerak Robot .....	26
Gambar 3.3. Komponen Kontrol Robot.....	27
Gambar 3.4. Komponen LCD dan sensor kompas .....	27
Gambar 3.5. Desain lintasan .....	28
Gambar 3.6. <i>Flowchart</i> Sistem .....	29
Gambar 3.7. Dimensi Motor .....	31
Gambar 3.8. Rangkaian <i>Driver Motor</i> .....	32
Gambar 3.9. Rangkaian Modul <i>Bluetooth</i> .....	32
Gambar 3.10. Rangkaian Encoder .....	33
Gambar 3.11. Rangkaian LCD.....	34
Gambar 3.12. Proses <i>Input</i> Perintah Suara .....	34



Gambar 3.13. Tampilan Aplikasi Kontrol Robot.....	35
Gambar 3.14. Pola Lintasan.....	36
Gambar 3.15 Pengujian Sensor <i>Rotary Encoder</i> .....	37
Gambar 3.16 Grafik Hasil Pengujian Sensor <i>Rotary Encoder</i> .....	38
Gambar 3.17 Pengujian Sensor Kompas .....	39
Gambar 3.18 Grafik Hasil Pengujian Sensor Kompas.....	40
Gambar 3.19 Pengujian <i>Voice Recognition</i> pada Robot.....	41

## DAFTAR TABEL

	Halaman
Tabel 2.1 Spesifikasi Arduino UNO-R3 .....	4
Tabel 2.2 Data untuk pin LCD.....	9
Tabel 3.1 <i>Gear Ratio</i> Motor.....	31
Tabel 3.2 Pengujian Sensor <i>Rotary Encoder</i> .....	38
Tabel 3.3 Pengujian Sensor Kompas .....	39
Tabel 3.4 Pengujian <i>Voice Recognition</i> .....	41
Tabel 3.5 Pengujian Komunikasi <i>Bluetooth</i> .....	42
Tabel 4.1 Pengujian dari Pos <i>Home</i> menuju Pos Area 1 .....	43
Tabel 4.2 Pengujian dari Pos Area 1 menuju Pos <i>Home</i> .....	44
Tabel 4.3 Pengujian dari Pos <i>Home</i> menuju Pos Area 2 .....	45
Tabel 4.4 Pengujian dari Pos Area 2 menuju Pos <i>Home</i> .....	45
Tabel 4.5 Pengujian dari Pos <i>Home</i> menuju Pos Area 3 .....	46
Tabel 4.6 Pengujian dari Pos Area 2 menuju Pos <i>Home</i> .....	47
Tabel 4.7 Pengujian dari Pos Area 1 menuju Pos Area 2 .....	47
Tabel 4.8 Pengujian dari Pos Area 2 menuju Pos Area 1 .....	48
Tabel 4.9 Pengujian dari Pos Area 1 menuju Pos Area 3 .....	49
Tabel 4.10 Pengujian dari Pos Area 3 menuju Pos Area 1 .....	49
Tabel 4.11 Pengujian dari Pos Area 2 menuju Pos Area 3 .....	50
Tabel 4.12 Pengujian dari Pos Area 3 menuju Pos Area 2 .....	51
Tabel 4.13 Rata-Rata Hasil Pengujian Keseluruhan.....	51

## BAB 1. PENDAHULUAN

### 1.1. Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi pada era yang modern ini khususnya dalam bidang robotika membawa dampak positif dalam kehidupan manusia. Berbagai robot telah diciptakan untuk memenuhi kebutuhan manusia, meskipun robot dibuat dengan didasari hobi, sebagai sarana pendidikan, atau dalam segi komersial. Banyak sekali negara-negara maju dan berkembang mulai berlomba untuk menciptakan sebuah robot mutakhir dengan keistimewaan khusus dengan bergantung pada kebutuhan manusia.

Sistem kontrol robot beroda dengan perintah suara dan sensor posisi menggunakan *rotary encoder* memungkinkan orang mengendalikan posisi robot dengan perintah suara ke tempat yang diinginkan. Teknologi *voice recognition* merupakan salah satu teknologi biometrika yang tidak memerlukan peralatan khusus dan relatif murah. Suara merupakan alat komunikasi yang unik dan dapat dibedakan dengan mudah. Selain itu, digunakan teknologi *rotary encoder* digunakan untuk mendeteksi perpindahan atau perpindahan robot.

Pada penelitian yang dilakukan oleh Jatra Kurnia Ardi (2014) telah dirancang suatu sistem *voice recognition* dengan judul “Implementasi Sistem *Voice Recognition* pada Robot Pemindah Objek sebagai Sistem Navigasi”. Namun pada penelitian ini masih menggunakan pengenalan suara dengan *EasyVR* yang tidak menggunakan perangkat nirkabel pada alat penelitian tersebut dan memerintah robot dengan sedikit berteriak ke robot agar suara dapat didengar jelas oleh sensor.

Oleh karena itu, dalam perancangan ini menggunakan *Smartphone android* sebagai sistem navigasi yang akan mengirimkan perintah ke Arduino menggunakan *Bluetooth smartphone* dan *Bluetooth module* sebagai *receiver* yang akan diteruskan ke Arduino untuk mengontrol berjalanya *mobile robot* dan menambahkan sensor *rotary encoder* sebagai pengontrol banyaknya putaran roda untuk mencapai suatu tempat yang telah diperintahkan.

## 1.2. Rumusan Masalah

Dalam penelitian ini ada beberapa hal yang menjadi rumusan masalah diantaranya :

1. Bagaimana menerapkan *smartphone android* sebagai *voice recognition* untuk mengontrol *mobile robot*.
2. Bagaimana menerapkan *rotary encoder* sebagai pengontrol banyaknya putaran roda.
3. Bagaimana menerapkan sensor kompas sebagai pengontrol arah hadap *mobile robot*.
4. Bagaimana mengontrol posisi robot agar menuju posisi yang ditentukan dengan tepat.

## 1.3. Tujuan

Penelitian ini dilakukan karena memiliki beberapa tujuan, diantaranya :

1. Dapat menerapkan *smartphone android* sebagai *voice recognition* untuk mengontrol *mobile robot*.
2. Dapat menerapkan *rotary encoder* sebagai pengontrol banyaknya putaran roda.
3. Dapat menerapkan sensor kompas sebagai pengontrol arah hadap *mobile robot*.
4. Dapat mengontrol posisi robot agar menuju posisi yang ditentukan dengan tepat.

## 1.4. Manfaat

Penelitian ini memiliki beberapa manfaat diantaranya :

1. *Mobile robot* dapat dikendalikan dengan suara manusia untuk mencapai suatu tempat yang diperintahkan.
2. Sebagai referensi pengendalian *mobile robot* khususnya pengendalian robot dengan suara dan *rotary encoder* sebagai sensor posisi.
3. Memperkaya literatur dalam bidang robotika.

### **1.5. Batasan Masalah**

Berdasarkan uraian rumusan masalah tersebut, maka pembahasan pada skripsi dibatasi pada hal-hal sebagai berikut :

1. *Mobile robot* menggunakan *differential wheeled robot*.
2. Menggunakan lintasan berukuran 2 x 2 m dengan 4 pembagian titik posisi robot.
3. Lintasan datar dan tidak ada penghalang apapun.

### **1.6. Sistematika Penulisan**

Sistematika penulisan laporan akhir dimaksudkan untuk memberikan gambaran secara garis besar dan jelas tentang isi laporan akhir, sehingga jelas hubungan antara bab yang satu dengan bab-bab berikutnya

#### **BAB I : PENDAHULUAN**

Bab ini berisi tentang latar belakang, perumusan masalah, tujuan dan manfaat, pembatasan masalah, metodologi penulisan, serta sistematika penulisan.

#### **BAB II : TINJAUAN PUSTAKA**

Bab ini berisi tentang teori dasar yang berhubungan dengan alat yang dirancang.

#### **BAB III : METODELOGI PENELITIAN**

Bab ini berisi tentang perancangan dan cara kerja rangkaian.

#### **BAB IV : HASIL DAN PEMBAHASAN**

Bab ini membahas tentang hasil penelitian yang dilakukan serta analisa dan evaluasi terhadap data-data hasil yang diperoleh saat pengukuran.

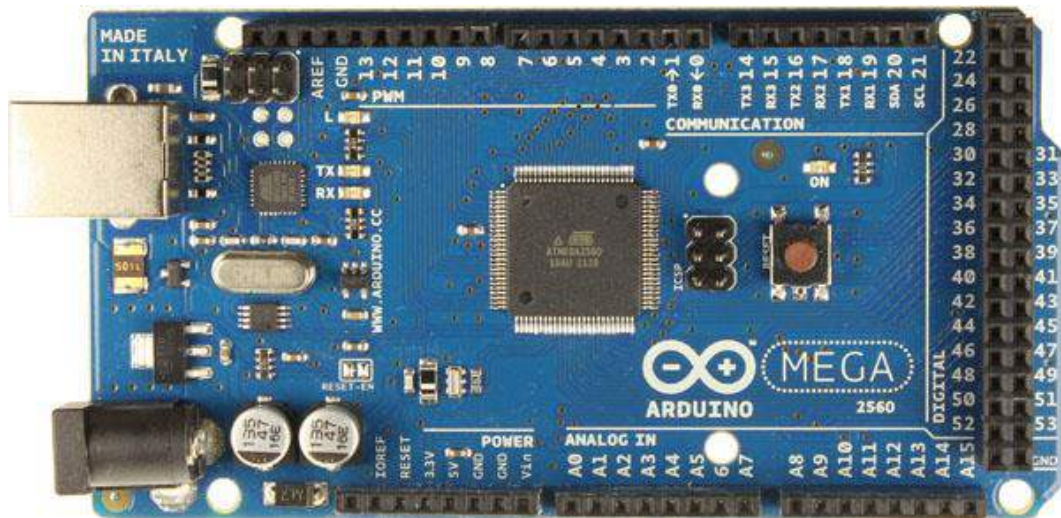
#### **BAB V : KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan yang diambil dari pembuatan alat dan saran-saran yang berguna untuk pengembangan alat tersebut.

## BAB 2. TINJAUAN PUSTAKA

### 2.1 Arduino Mega 2560

Arduino Mega 2560 adalah papan mikrokontroler berbasis ATmega 2560. 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), 16MHz kristal osilator, koneksi USB, jack power, header ICSP, dan tombol reset. Ini semua yang diperlukan untuk mendukung mikrokontroler. Cukup dengan menghubungkannya dengan adaptor AC-DC atau baterai untuk mulai mengaktifkannya.



Gambar 2.1. Modul Kontroler Arduino Mega 2560

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

Berikut ini adalah tabel spesifikasi dari Arduino Uno R3 dapat kita lihat pada Tabel 2.1 dibawah ini.

Tabel 2.1 Spesifikasi Arduino UNO-R3

Microcontroller	Atmega2560
Tegangan Operasi	5 V
Tegangan <i>Input</i>	7-12 V
Tegangan <i>Input</i> (Batas)	6-20 V
I/O Digital	54 (15 buah menyediakan PWM <i>output</i> )
Input Analog	16 buah
Arus DC per I/O pin	40 Ma

Arus DC untuk pin 3.3	50 Ma
Flash Memory	256KB ( 8KB digunakan untuk <i>bootloader</i> )
SRAM	8 Kb
EEPROM	4 Kb
Kecepatan Klok	16 MHz

---

Diterjemahkan dari: (Arduino MEGA2560 2015)

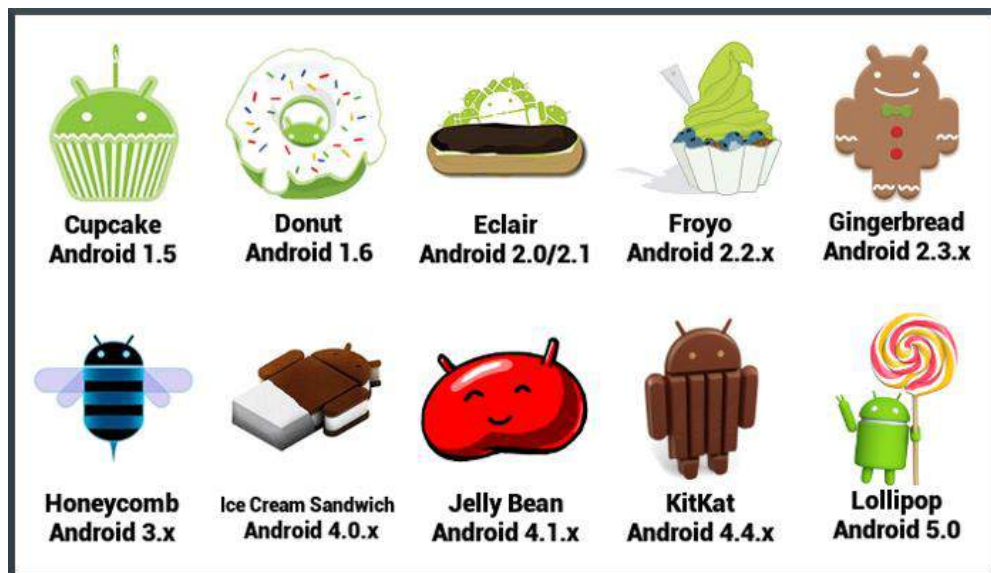
---

## 2.2 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya *Open Handset Alliance*, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008.

Antarmuka pengguna Android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata, seperti menggesek, mengetuk, mencubit, dan membalikkan cubitan untuk memanipulasi obyek di layar. Android adalah sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (*apps*) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman *Java*. Pada bulan Oktober 2012, ada sekitar 700.000 aplikasi yang tersedia untuk Android, dan sekitar 25 juta aplikasi telah diunduh dari *Google Play*, toko aplikasi utama Android. Sebuah survey pada bulan April-Mei 2013 menemukan bahwa Android adalah platform paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi seluler.

Aplikasi Android dikembangkan dalam bahasa pemrograman Java dengan menggunakan kit pengembangan perangkat lunak Android (SDK). SDK ini terdiri dari seperangkat perkakas pengembangan, termasuk debugger, perpustakaan perangkat lunak, emulator handset yang berbasis QEMU, dokumentasi, kode sampel, dan tutorial. Didukung secara resmi oleh lingkungan pengembangan terpadu (IDE) *Eclipse*, yang menggunakan *plugin Android Development Tools* (ADT). Perkakas pengembangan lain yang tersedia di antaranya adalah *Native Development Kit* untuk aplikasi atau ekstensi dalam C atau C++, *Google App Inventor*, lingkungan visual untuk pemrogram pemula, dan berbagai kerangka kerja aplikasi web seluler lintas platform.



Gambar 2.2. Perkembangan Versi-Versi Android

(Sumber : suteki-share.blogspot.com)

### 2.3 Bluetooth Module HC-05

HC-05 Adalah sebuah modul *Bluetooth SPP* ( *Serial Port Protocol*) yang mudah digunakan untuk komunikasi serial *wireless* ( nirkabel) yang mengkonversi *port* serial ke Bluetooth. HC-05 menggunakan modulasi bluetooth V2.0 + EDR ( *Enhanced Data Rate*) 3 Mbps dengan memanfaatkan gelombang radio berfrekuensi 2, 4 GHz. Modul ini dapat digunakan sebagai *slave* maupun *master*.



HC-05 memiliki 2 mode konfigurasi, yaitu *AT mode* dan *Communication mode*. *AT mode* berfungsi untuk melakukan pengaturan konfigurasi dari HC-05. Sedangkan *Communication mode* berfungsi untuk melakukan komunikasi bluetooth dengan piranti lain.

Dalam penggunaannya, HC-05 dapat beroperasi tanpa menggunakan *driver* khusus. Untuk berkomunikasi antar *Bluetooth*, minimal harus memenuhi dua kondisi berikut :

- Komunikasi harus antara *master* dan *slave*.
- *Password* harus benar ( saat melakukan pairing) .
- Jarak sinyal dari HC-05 adalah 30 meter, dengan kondisi tanpa halangan.

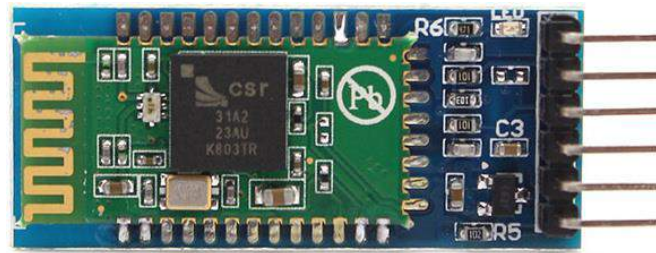
Adapun spesifikasi dari HC-05 adalah :

a. *Hardware* :

- Sensitivitas -80dBm ( *Typical*)
- Daya transmit RF sampai dengan + 4dBm.
- Operasi daya rendah 1, 8V – 3, 6V I/ O.
- Kontrol PIO.
- Antarmuka UART dengan *baudrate* yang dapat diprogram.
- Dengan antena terintegrasi.

b. *Software* :

- Default *baudrate* 9600, Data bit : 8, Stop bit = 1, *Parity* : *No Parity*, Mendukung *baudrate* : 9600, 19200, 38400, 57600, 115200, 230400 dan 460800.
- *Auto* koneksi pada saat *device* dinyalakan ( *default*) .
- *Auto reconnect* pada menit ke 30 ketika hubungan putus karena *range* koneksi.

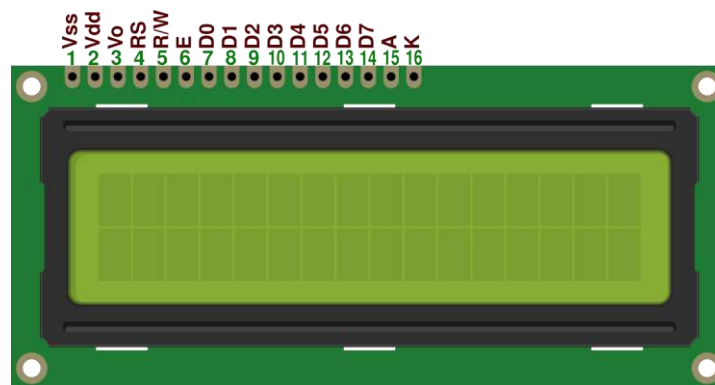


Gambar 2.3. Modul *Bluetooth* HC-05

(Sumber : [electrosome.com](http://electrosome.com))

## 2.4 LCD 16x2

Display elektronik adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD (*Liquid Cristal Display*) adalah salah satu jenis display elektronik yang dibuat dengan teknologi *CMOS logic* yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. LCD (*Liquid Cristal Display*) berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik.



Gambar 2.4. Modul LCD (*Liquid Cristal Display*)

(Sumber : [blog.electrotec.pe](http://blog.electrotec.pe))

Dalam modul LCD (*Liquid Cristal Display*) terdapat microcontroller yang berfungsi sebagai pengendali tampilan karakter LCD (*Liquid Cristal Display*). Microcontroller pada suatu LCD (*Liquid Cristal Display*) dilengkapi dengan memori dan register. Memori yang digunakan microcontroller internal LCD adalah : DDRAM (*Display Data Random Access Memory*) merupakan memori tempat karakter yang akan ditampilkan berada. CGRAM (*Character Generator Random*

*Access Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat diubah-ubah sesuai dengan keinginan. CGROM (*Character Generator Read Only Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut merupakan karakter dasar yang sudah ditentukan secara permanen oleh pabrikan pembuat LCD (*Liquid Cristal Display*) tersebut sehingga pengguna tinggal mengambilnya sesuai alamat memorinya dan tidak dapat merubah karakter dasar yang ada dalam CGROM. *Register control* yang terdapat dalam suatu LCD diantaranya adalah. Register perintah yaitu register yang berisi perintah-perintah dari mikrokontroler ke panel LCD (*Liquid Cristal Display*) pada saat proses penulisan data atau tempat status dari panel LCD (*Liquid Cristal Display*) dapat dibaca pada saat pembacaan data. Register data yaitu register untuk menuliskan atau membaca data dari atau ke DDRAM. Penulisan data pada register akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya.

Karena LCD sudah dilengkapi perangkat kontrol sendiri yang menyatu dengan LCD, maka kita mengikuti aturan standar yang telah disimpan dalam pengontrolan tersebut. konfigurasi pin yang terdapat dalam LCD adalah:

Tabel 2.2 Data untuk pin LCD

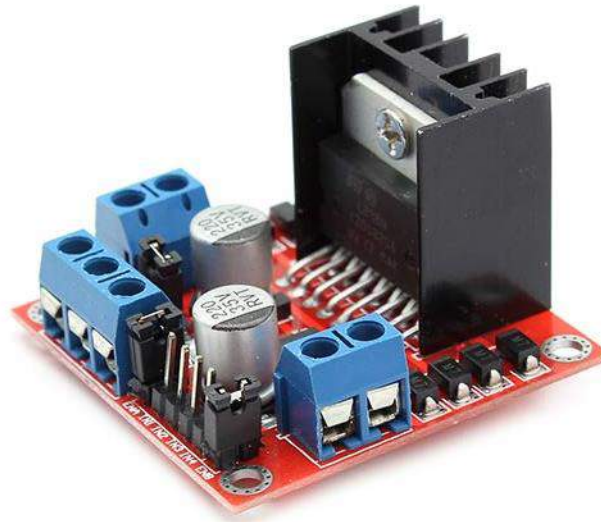
<b>PIN</b>	<b>SIMBOL</b>	<b>NILAI</b>	<b>FUNGSI</b>
1	Vss	–	<i>Power supply</i> 0 volt (ground)
2	Vdd/Vcc	–	<i>Power supply</i> Vcc
3	Vee	–	Seting kontras
4	RS	0/1	0: intruksi <i>input</i> / 1: data <i>input</i>
5	R/W	0/1	0: tulis ke LCD / 1: membaca dari LCD

PIN	SIMBOL	NILAI	FUNGSI
6	E	0→1	Mengaktifkan sinyal
7	DB0	0/1	Data pin 0
8	DB1	0/1	Data pin 1
9	DB2	0/1	Data pin 2
10	DB3	0/1	Data pin 3
11	DB4	0/1	Data pin 4
12	DB5	0/1	Data pin 5
13	DB6	0/1	Data pin 6
14	DB7	0/1	Data pin 7
15	VB+	–	<i>Power 5 Volt (Vcc) Lampu latar (jika ada)</i>
16	VB-	–	<i>Power 0 Volt (ground) Lampu latar (jika ada)</i>

## 2.5 Driver Motor

Motor DC tidak dapat dikendalikan secara langsung oleh mikrokontroler, karena kebutuhan arus listrik yang besar pada motor DC sedangkan arus keluaran pada mikro sangat kecil. *Driver* motor merupakan pilihan alternatif yang harus

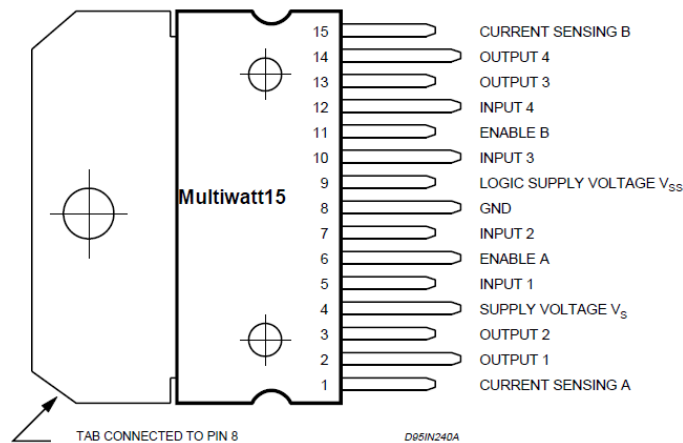
digunakan untuk mengendalikan motor DC pada robot beroda. Ada beberapa *driver* motor yang sering digunakan pada aplikasi robotika, yaitu menggunakan rangkaian *H-Bridge* transistor, *H-Bridge* MOSFET, dan IC *driver motor*.



Gambar 2.5. Modul *Driver Motor*

(Sumber : lectronilab.co)

*Driver motor* digunakan untuk mengontrol arah putaran dan kecepatan motor DC yang merupakan penggerak utama dari rangkaian proyek akhir ini. IC driver motor L298 yang didalamnya terdapat rangkaian *H-Bridge* akan mengontrol putaran motor sesuai data masukan digital yang berasal dari PLC Zelio SR2 B201 BD, dan pada IC L298 ini juga terdapat pin untuk pengaturan aplikasi PWM (*Pulse Width Modulator*) yang akan mengatur kecepatan motor dc yang dikendalikannya. L298 memiliki rangkaian dual *H-Bridge*, sehingga mampu mengendalikan dua buah motor DC sekaligus. Konstruksi pin *driver motor* DC IC L298 adalah sebagai berikut.



Gambar 2.6. Konfigurasi PIN Driver Motor L298

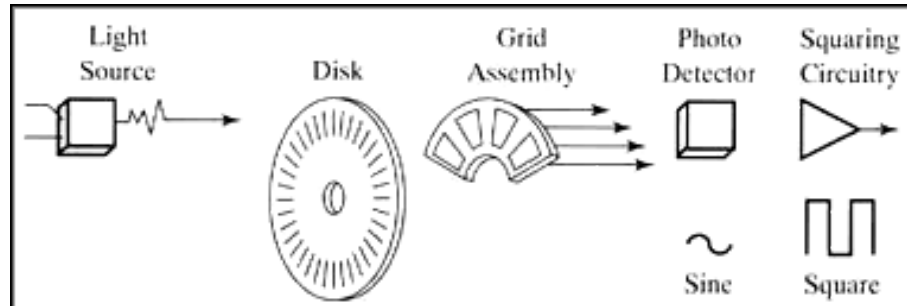
(Sumber : fos.cmb.ac.lk)

## 2.6 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*, dan sebagainya.

*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 divais 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. Gambar 2.7 menunjukkan bagan skematik sederhana dari *rotary encoder*. 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.



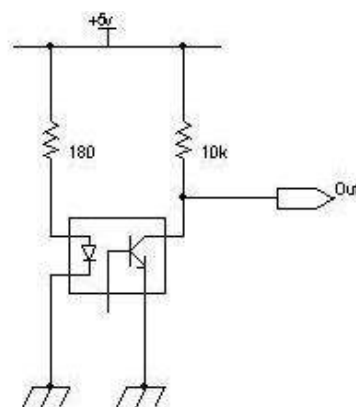
Gambar 2.7. Optical Rotary Encoder

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

Rangkaian penghasil pulsa (Gambar 2.8) yang digunakan umumnya memiliki output yang berubah dari +5V menjadi 0.5V ketika cahaya diblok oleh piringan dan ketika diteruskan ke *photo-transistor*. Karena divais ini umumnya bekerja dekat dengan motor DC maka banyak *noise* yang timbul sehingga biasanya *output* akan dimasukkan ke *low-pass filter* dahulu. Apabila *low-pass filter* digunakan, *frekuensi cut-off* yang dipakai umumnya ditentukan oleh jumlah slot yang ada pada piringan dan seberapa cepat piringan tersebut berputar, dinyatakan dengan:

$$f_c = \frac{s_w n}{60} \dots\dots\dots(2.1)$$

Dimana  $f_c$  adalah frekuensi *cut-off filter*,  $s_w$  adalah kecepatan piringan dan  $n$  adalah jumlah slot pada piringan.



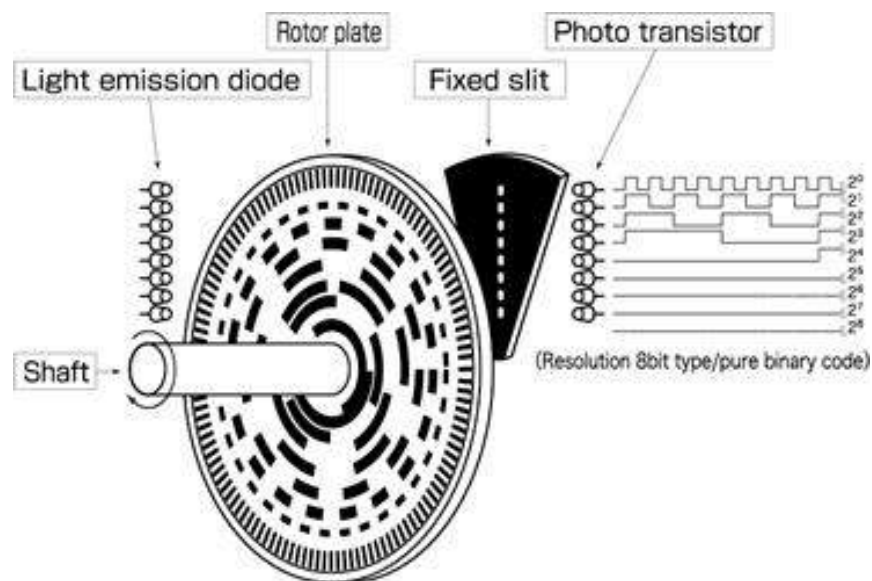
Gambar 2.8. Rangkaian Tipikal Penghasil Pulsa pada *Rotary Encoder*

(Sumber : [konversi.wordpress.com](http://konversi.wordpress.com))

*Rotary encoder* sendiri terbagi menjadi dua jenis jaitu :

a. *Absolute Rotary Encoder*

Prinsip kerja dari *absolute rotary encoder* ini menggunakan piringan yang memiliki pola unik serta sinyal optik yang tersusun dalam jalur *concentric*. Hal ini akan menghasilkan kode digital yang berbeda satu sama lain untuk menyatakan sejumlah posisi tertentu dari poros yang dihubungkan pada *encoder* ini. Piringan yang digunakan untuk *absolut encoder* ini sendiri tersusun dari segmen-segmen cincin konsentris yang dimulai dari bagian tengah piringan ke arah tepi luar piringan yang jumlah segmennya selalu dua kali jumlah segmen cincin sebelumnya. Sebagai contoh, cincin pertama yang terdapat pada bagian paling dalam memiliki satu segmen transparan dan satu segmen gelap, sedangkan cincin kedua memiliki dua segmen transparan dan dua segmen gelap, begitu seterusnya hingga cincin terluar.



Gambar 2.9. *Absolute Rotary Encoder*

(Sumber : [www.tamagawa-seiki.com](http://www.tamagawa-seiki.com))

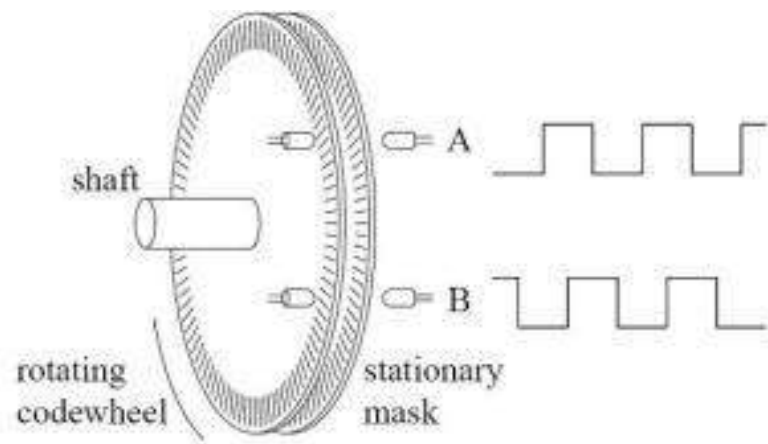
Kelipatan dua yang terdapat pada tiap-tiap segmen piringan tersebut yang akan membuat suatu sistem biner. Sistem biner ini juga dibentuk akibat terjadinya pasangan antara LED dan *photo transistor* sebanyak jumlah cincin



yang ada pada *absolut encoder* tersebut. Dengan membaca *output* biner yang dihasilkan maka posisi dari poros yang kita ukur dapat kita ketahui untuk diteruskan ke rangkaian pengendali. Semakin banyak bit yang kita pakai maka posisi yang dapat kita peroleh akan semakin banyak.

b. *Incremental Rotary Encoder*

*Incremental encoder* terdiri dari dua *track* atau *single track* dan dua sensor yang disebut *channel A* dan *B* (Gambar 7). Ketika poros berputar, deretan pulsa akan muncul di masing-masing *channel* pada frekuensi yang proporsional dengan kecepatan putar sedangkan hubungan fasa antara *channel A* dan *B* menghasilkan arah putaran. Dengan menghitung jumlah pulsa yang terjadi terhadap resolusi piringan maka putaran dapat diukur. Untuk mengetahui arah putaran, dengan mengetahui *channel* mana yang *leading* terhadap *channel* satunya dapat kita tentukan arah putaran yang terjadi karena kedua *channel* tersebut akan selalu berbeda fasa seperempat putaran (*quadrature signal*). Seringkali terdapat *output channel* ketiga, disebut *INDEX*, yang menghasilkan satu pulsa per putaran berguna untuk menghitung jumlah putaran yang terjadi.



Gambar 2.10. Susunan Piringan untuk Incremental Encoder

(Sumber : konversi.wordpress.com)

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

$$\omega_r = \frac{\alpha_f}{T} \dots\dots\dots(2.2)$$

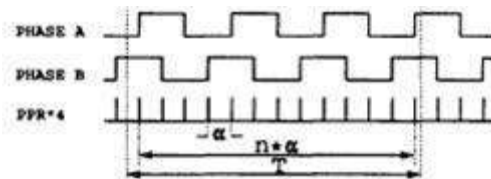
Cara yang sederhana untuk menentukan kecepatan dapat dengan *frequencymeter*, yakni menghitung jumlah pulsa dari *encoder*, n, pada selang waktu yang tetap, T, yang merupakan periode *loop* kecepatan (Gambar 10). Apabila  $\alpha$  adalah sudut antara pulsa *encoder*, maka sudut putaran pada suatu periode adalah:

$$\alpha_f = n\alpha \dots\dots\dots(2.3)$$

Sehingga kecepatan putar akan kita dapatkan sebagai:

$$\omega_r = \frac{\alpha_f}{T} \dots\dots\dots(2.4)$$

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



Gambar 2.11. Sinyal Keluaran *Encoder* untuk Pengukuran Kecepatan dengan *Frequencymeter*

(Sumber : konversi.wordpress.com)

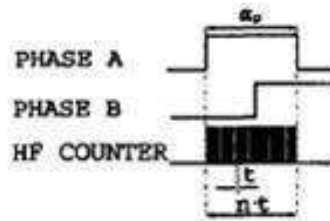
Cara yang lain adalah dengan menggunakan periodimeter. Dengan cara ini kita akan mengukur kecepatan tidak lagi dengan menghitung jumlah pulsa *encoder* tetapi dengan menghitung *clock* frekuensi tinggi (HF Clock) untuk sebuah pulsa dari *encoder* yaitu mengukur periode pulsa dari *encoder* (Gambar 11). Apabila  $\alpha_p$  adalah sudut dari pulsa *encoder*, t adalah periode dari HF clock, dan n adalah jumlah pulsa HF yang terhitung pada *counter*. Maka waktu untuk sebuah pulsa *encoder*,  $T_p$ , adalah:

$$T_p = nt \dots\dots\dots(2.5)$$

Sehingga kecepatan yang akan kita ukur dapat kita peroleh dengan:

$$\omega_r = \frac{\alpha_p}{T_p} \dots\dots\dots(2.6)$$

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



Gambar 2.12. Pengukuran Kecepatan dengan Menggunakan Periodimeter

(Sumber : konversi.wordpress.com)

## 2.7 Modul Kompas

Kompas adalah alat navigasi untuk menentukan arah berupa sebuah panah penunjuk magnetis yang bebas menyelaraskan dirinya dengan medan magnet bumi secara akurat. Kompas memberikan rujukan arah tertentu, sehingga sangat membantu dalam bidang navigasi. Arah mata angin yang ditunjuknya adalah utara, selatan, timur, dan barat. Apabila digunakan bersama-sama dengan jam dan sekstan, maka kompas akan lebih akurat dalam menunjukkan arah. Alat ini membantu perkembangan perdagangan maritim dengan membuat perjalanan jauh lebih aman dan efisien dibandingkan saat manusia masih berpedoman pada kedudukan bintang untuk menentukan arah.

Diciptakannya kompas *digital* bertujuan untuk melengkapi kebutuhan robotika yang semakin canggih. Dunia robotika ini sangat membutuhkan alat navigasi yang efektif dan efisien. Sementara itu alat sistem navigasi yang tersedia di pasaran harganya mahal. Sedangkan kompas sendiri merupakan sebuah alat sistem navigasi yang efektif dengan harga lebih murah. oleh karena itu kompas digital diharapkan bisa mensubstitusi alat sistem navigasi pada robot.

Modul Kompas GY-273 adalah sebuah modul yang digunakan untuk menunjukkan arah mata angin digital, atau juga disebut kompas digital. Modul ini

menggunakan komponen utama berupa IC HMC5883 yang merupakan IC kompas digital 3 axis yang memiliki interface berupa 2 pin I2C.

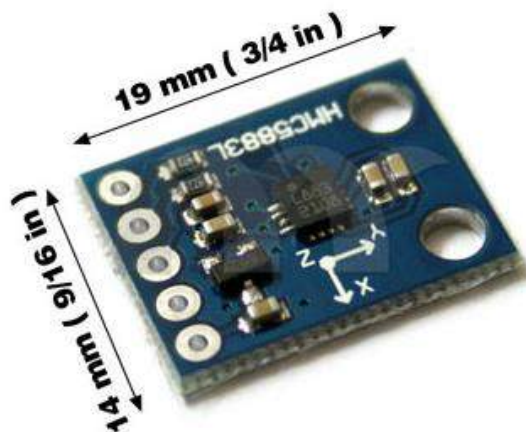
HMC5883 memiliki sensor *magneto-resistive* HMC118X series ber-resolusi tinggi, ditambah ASIC dengan konten *amplification, automatic degaussing strap driver, offset cancellation* dan 12 bit ADC yang memungkinkan keakuratan kompas mencapai 1 sampai 2 derajat. Modul ini biasa digunakan untuk keperluan sistem navigasi otomatis, *mobile phone, netbook* dan perangkat navigasi personal.

Modul ini memiliki 5 pin, diantaranya :

1. VCC (5V)
2. GND
3. SCL
4. SDA
5. DRDY

Berikut adalah beberapa fitur dari Modul Kompas GY-273 :

- Berbasis sensor *magnetoresistive 3 axis*.
- 12-Bit ADC terkopling dengan *Low Noise* AMR Sensor yang memiliki 2 *mili-gauss Field* dengan resolusi.kurang lebih 8 *Gauss Fields*.
- Tegangan kerja 5V DC.
- Menggunakan antarmuka I2C.
- Keluaran rata-rata maksimum 160 Hz.



Gambar 2.13. Sensor GY-273  
(Sumber : [www.nyplatform.com](http://www.nyplatform.com))

## 2.8 App. Invertor

App Inventor adalah tool yang digunakan untuk membuat aplikasi android berbasis web. Salah satu *feature* dari app inventor adalah untuk membuat program androidnya seperti menyusun puzzle. Kita pasangkan blok – blok tersebut agar saling terhubung. Setelah program sudah jadi kita dapat *build* menjadi file .apk untuk dicoba di smartphone android. Untuk menggunakan tools ini silahkan login dengan akun gmail anda di <http://ai2.appinventor.mit.edu/>.



Gambar 2.14. Logo App Inventor  
(Sumber : [georgepavlides.info](http://georgepavlides.info))

## 2.9 Teori Odometry

*Odometry* adalah penggunaan data dari pergerakan aktuator untuk memperkirakan perubahan posisi dari waktu ke waktu. *odometry* digunakan untuk memperkirakan posisi relatif terhadap posisi awal.

Untuk memperkirakan posisi relatif robot, digunakan perhitungan jumlah pulsa yang dihasilkan oleh sensor *rotary encoder* setiap satuan ukuran yang kemudian dikonversi menjadi satuan *millimeter*. Untuk mendapatkan jumlah pulsa setiap satu kali putaran roda digunakan rumus sebagai berikut:

$$K \text{ roda} = 2 \times \pi \times r$$

$$\text{pulsa\_per\_mm} = \text{resolusi\_enc} / K \text{ roda}$$

Pada sistem penggerak diferensial terdapat dua roda, yaitu roda kanan dan roda kiri dan dimisalkan jumlah pulsa\_per\_mm untuk roda kanan adalah *right\_encoder* dan roda kiri adalah *left\_encoder* dan jarak antara dua roda adalah *wheel\_base* maka didapatkan jarak tempuh (*distance*) dan sudut orientasi ( $\theta$ ). Rumusnya adalah sebagai berikut.

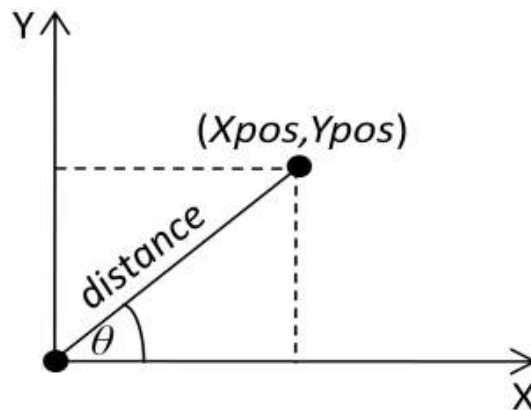
$$\text{distance} = (\text{left\_enc} + \text{right\_enc}) / 2$$

$$\theta = (\text{left\_enc} - \text{right\_enc}) / \text{wheel\_base}$$

Karena  $\theta$  adalah sudut dalam radian maka untuk mengetahui sudut dalam derajat (*heading*) digunakan rumus sebagai berikut :

$$\text{heading} = \theta \times \frac{180}{\pi}$$

Dari ketentuan diatas didapatkan bahwa nilai *heading* akan bernilai negatif (-) ketika robot berputar melawan arah jarum jam dan akan bernilai positif (+) ketika robot berputar searah dengan jarum jam. Dengan mengetahui jarak dan sudut (*distance* dan  $\theta$ ) maka kita dapat mengetahui koordinat X dan koordinat Y dengan persamaan trigonometri.



Gambar 2.15. Ilustrasi pada sumbu cartesian

Dari ilustrasi diatas maka koordinat dari robot dapat kita ketahui dengan rumus :

$$X_{pos} = distance \times \sin(\theta)$$

$$Y_{pos} = distance \times \cos(\theta)$$

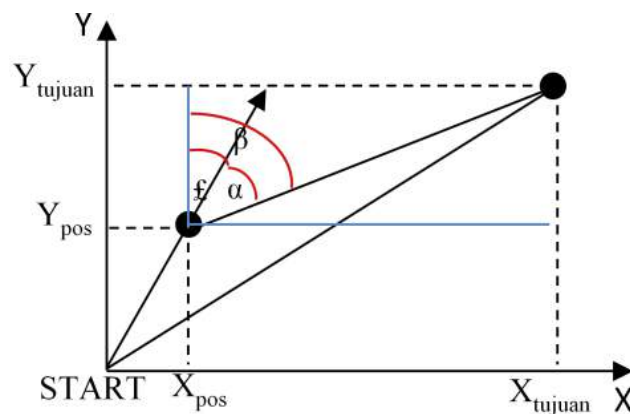
Untuk menentukan *error* arah hadap dari robot terhadap titik tujuan maka digunakan teorema *pythagoras* yang akan menghasilkan posisi (*path*) saat ini dan jarak terhadap titik tujuan, berikut perhitungannya:

$$x = X_{tujuan} - X_{pos}$$

$$y = Y_{tujuan} - Y_{pos}$$

$$target\_distance = \sqrt{x^2 + y^2}$$

*Heading* dari robot yang telah diketahui sehingga kita dapat menghitung *error* arah hadap (*heading error*) robot terhadap titik tujuan.

Gambar 2.16. Sudut  $\alpha$ ,  $\beta$ , dan  $\epsilon$

Pada gambar 2.18. menunjukkan ilustrasi untuk mencari *heading error* ( $a$ ) dimana  $\beta$  adalah *target bearing* yaitu sudut antara posisi robot saat ini terhadap titik tujuan. Sedangkan garis berwarna biru adalah garis bantu yang masing-masing sejajar dengan sumbu X dan sumbu Y.

Untuk mendapat nilai dari  $\beta$ , maka digunakan rumus sebagai berikut :

$$\beta = \arctan \frac{Y_{tujuan} - Y_{pos}}{X_{tujuan} - X_{pos}}$$

sehingga

$$a = \beta - \xi$$



## BAB 3. METODOLOGI PENELITIAN

### 3.1. Tempat dan Waktu Penelitian

Penelitian “Implementasi *System Voice Recognition* dan *Rotary Encoder* pada Robot Beroda sebagai Sistem Navigasi dan Perhitungan Posisi Robot” dilaksanakan di Laboratorium Sistem Kendali Universitas Jember di Jl. Slamet Riyadi no. 62 Patrang, Jember. Penelitian ini dilaksanakan mulai Bulan November 2015 hingga Bulan Juni 2016.

### 3.2. Prosedur Penelitian

Dalam pembuatan robot beroda ini dibutuhkan langkah-langkah prosedur penelitian sebagai berikut :

#### 1. Studi Literatur

Tahap awal dari penelitian ini mencari literatur dari hasil penelitian sebelumnya. Diharapkan dengan literatur yang didapat bias memberikan keyakinan bahwa penelitian ini dapat dilaksanakan dan memberikan arahan untuk mengurangi kesalahan dalam penelitian.

#### 2. Pembelian bahan pembuatan

Tahap kedua adalah pembelian material pendukung dalam pembuatan *mobile robot*. Pembelian ini meliputi alat dan komponen pendukung dalam pembuatan *mobile robot*.

#### 3. Pengerjaan alat baik mekanik dan rangkaian

Tahap ketiga adalah proses pengerjaan dimana alat dan bahan yang telah dibeli digunakan untuk membangun sistem mekanik dan komponen untuk pembuatan rangkaian *mobile robot*.

#### 4. Pengujian alat

Tahap keempat adalah pengujian alat diharapkan dengan adanya tahap ini fungsi kerja baik mekanik dan rangkaian yang telah dibuat sudah masuk dan sesuai dengan target yang diharapkan. Sehingga nantinya alat yang telah dibuat dapat bekerja secara maksimal.

#### 5. Perencanaan alat

Tahap kelima merupakan penggabungan dari mekanik dan rangkaian yang telah dibuat. Proses ini meliputi penempatan letak rangkaian sangat menentukan dalam proses pencarian data saat *mobile robot* berjalan.

#### 6. Pengujian alat dan analisis

Menguji sistem-sistem yang telah terintegrasi secara menyeluruh lebih dahulu, selanjutnya menganalisis data yang didapatkan pada saat pengujian. Diharapkan pada tahap ini dapat ditemukan sebuah gagasan baru dalam mengurangi kesalahan pada tiap tahapan di atasnya.

### 3.3. Alat dan Bahan

Alat dan bahan yang digunakan dalam penelitian ini antara lain:

#### 3.3.1. Alat

1. Komputer/laptop
2. Tang kombinasi
3. Solder
4. Bor PCB
5. AVO meter
6. *Serial to usb Converter*
7. Obeng kombinasi
8. *Smartphone*

#### 3.3.2. Alat

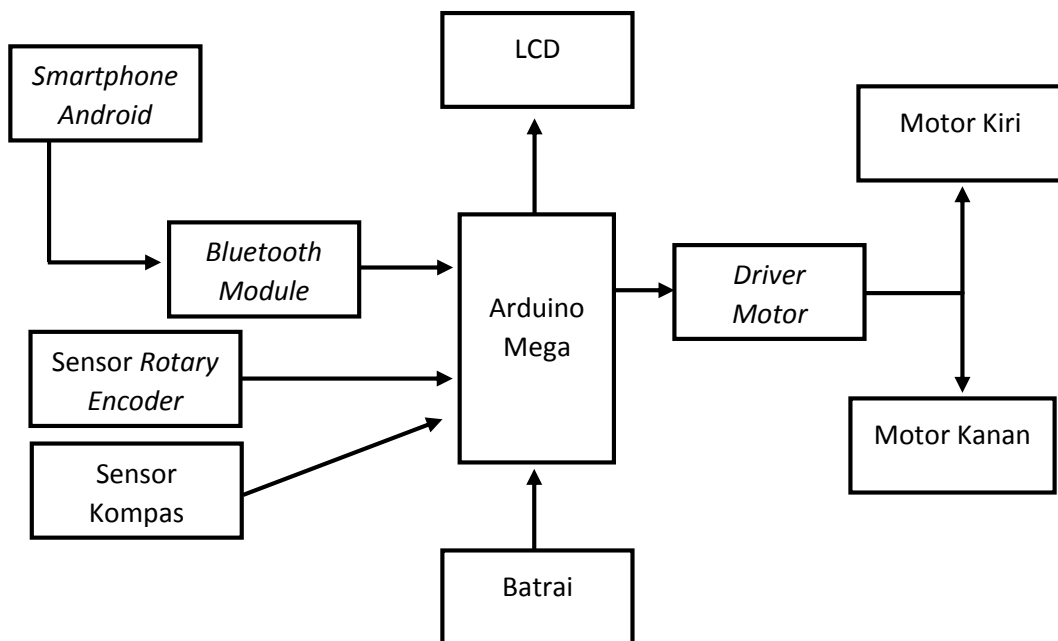
3. *Chasis Robot*
4. Roda Robot
5. Arduino
6. LCD
7. *Bluetooth module*
8. *Driver motor*
9. Batrai
10. Motor dengan *rotary encoder*

11. Sekrup dan mur baut serta komponen-komponen lainnya
12. Modul kompas
13. Buzzer
14. Kabel penghubung

### 3.3.3. Software

1. *Open-source* Arduino
2. Aplikasi kontrol robot pada *smartphone*
3. *App Inventor*

### 3.4. Blok Diagram



Gambar 3.1. Blok Diagram

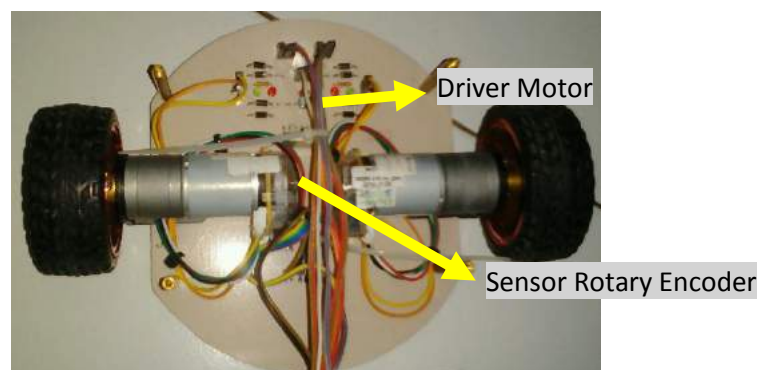
Pada gambar 3.1 menunjukkan blok sistem kerja mobile robot yang dikendalikan dengan suara berbasis Arduino. Dalam perancangan tersebut terdiri dari beberapa bagian blok sistem diantaranya Smartphone Android dengan software yang dibuat dengan App Inventor, blok sistem Bluetooth module, blok sistem sensor rotary encoder, blok sistem batrai, blok sistem LCD, blok sistem driver motor, dan blok sistem motor DC. Tegangan untuk mensupply mobile robot tersebut yaitu berasal dari batrai. Smartphone Android dengan software App

Inventor akan menerima perintah suara dan akan memproses perintah tersebut menjadi perintah yang dapat dimengerti oleh mikrokontroler Arduino. Setelah itu, perintah akan dikirim melalui Bluetooth yang terdapat pada smartphone dan akan diterima oleh Bluetooth module yang akan diteruskan ke dalam Arduino Mega.

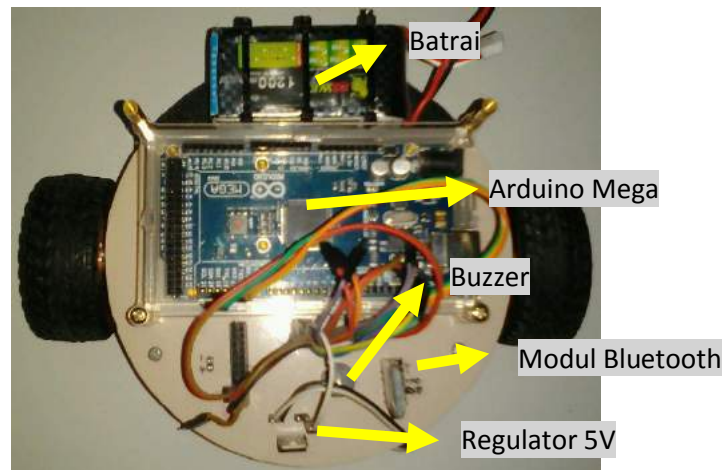
Kemudian Arduino akan mengolah perintah tersebut dan akan mengaktifkan driver motor untuk menjalankan motor DC beserta rodanya untuk menuju tempat yang telah diperintah. Disini dalam sensor rotary encoder dan sensor kompas akan memandu robot dengan cara menghitung jarak yang ditempuh robot dan mengatur sudut belok robot agar dapat menuju tempat yang telah diperintah.

### 3.5. Desain Penelitian

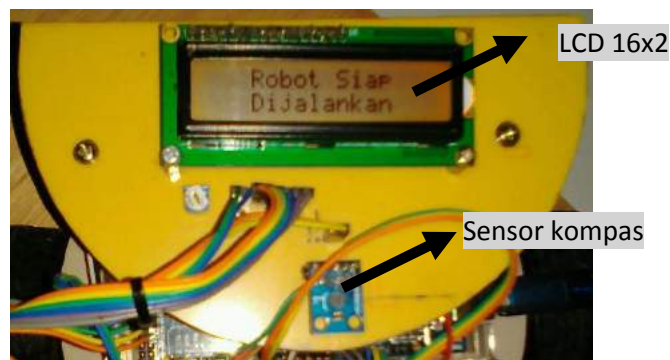
Desain *hardware*, dan lintasan “Implementasi *System Voice Recognition* dan *Rotary Encoder* pada Robot Beroda sebagai Sistem Navigasi dan Perhitungan Posisi Robot” adalah sebagai berikut:



Gambar 3.2. Komponen Gerak Robot



Gambar 3.3. Komponen Kontrol Robot

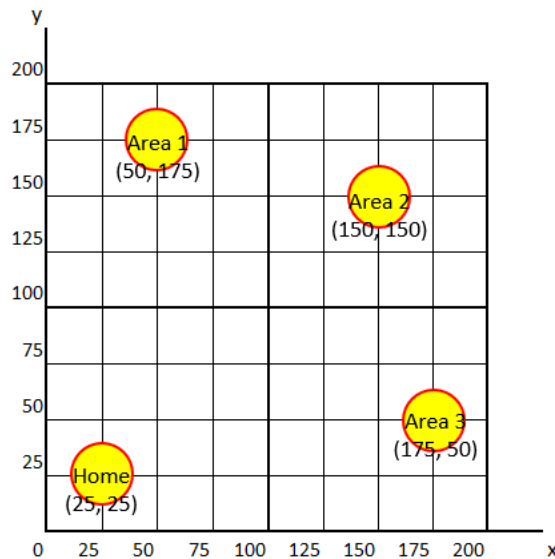


Gambar 3.4. Komponen LCD dan sensor kompas

Pada gambar 3.2 terlihat driver motor L298N untuk mengatur pergerakan motor DC pada robot dan mengatur kecepatan putar motor tersebut. Motor yang digunakan berjenis JGA25-371 yang sudah terpasang sensor rotary encoder. Sensor rotary encoder berguna untuk menghitung pulsa yang dibutuhkan untuk roda berputar sebanyak satu kali dan akan dikonversi menjadi penghitung jarak tempuh robot.

Pada gambar 3.3 terlihat Arduino Mega yang digunakan untuk mengolah program dan menjalankan program. Terdapat Bluetooth module yang bertujuan untuk mengkoneksikan robot dengan smartphone android. Ada pula buzzer yang berfungsi untuk memberikan peringatan berupa suara dan regulator 7805 untuk membuat input tegangan 5 volt yang berasal dari batri.

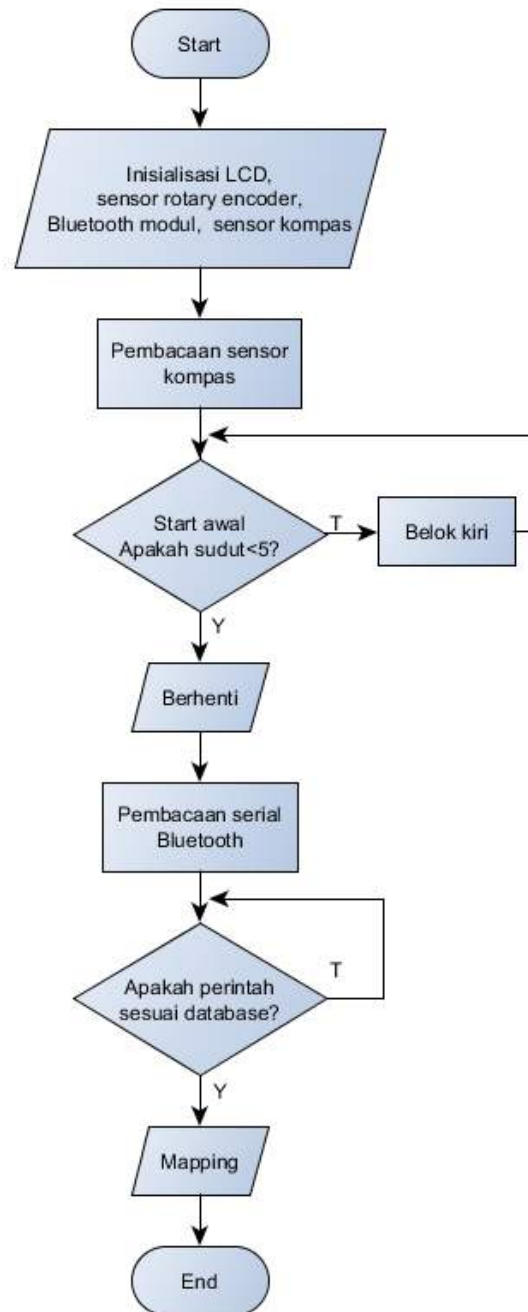
Pada gambar 3.4 terdapat LCD yang berguna untuk menampilkan karakter dan sensor kompas HMC5883L yang berfungsi sebagai navigasi robot. Khusus untuk sensor kompas harus berada pada ketinggian sekitar 20cm dari lantai karena medan magnet pada motor mempengaruhi kinerja sensor kompas.



Gambar 3.5. Desain lintasan

Untuk lintasan sendiri terdiri dari 4 pos dengan pembagian area 1 sampai 3 dan sebuah home untuk menjadi titik awal robot dijalankan. Titik home mempunyai koordinat titik (25,25), titik area 1 mempunyai koordinat titik (50,175), titik area 2 mempunyai koordinat titik (150,150), dan titik area 3 mempunyai koordinat titik (175,50).

### 3.6. Flowchart



Gambar 3.6. *Flowchart* Sistem

Pada gambar 3.6 ditunjukkan alur sistem proses implementasi *system voice recognition* dan *rotary encoder* pada *mobile robot* sebagai sistem navigasi dan perhitungan posisi robot berbasis Arduino mega. Pertama yaitu inisialisasi terhadap LCD, *Bluetooth module* dan sensor yang digunakan yaitu sensor *rotary encoder*.

Kemudian proses pembacaan sensor kompas yang digunakan untuk menentukan arah hadap robot pada titik  $0^\circ$ . Apabila sudut hadap robot diatas  $5^\circ$  maka robot akan berputar ke kiri hingga sudut hadap robot kurang dari  $5^\circ$ . Apabila sudut hadap robot sudah kurang dari  $5^\circ$  maka robot akan berhenti dan siap untuk diperintah.

Selanjutnya, pada proses pembacaan sensor suara dimana seseorang akan mengucapkan kata ke *smartphone android* dan akan mengirimkan perintah melalui *Bluetooth* dan akan diterima oleh *Bluetooth module* untuk diteruskan kedalam Arduino Mega. Dilanjutkan pada proses pembacaan, jika perintah yang diucapkan seseorang sesuai dengan *database* maka dapat meneruskan ke proses selanjutnya dan jika perintah yang diucapkan tidak sesuai *database* maka kembali lagi ke proses pembacaan sensor suara.

*Database* disini sudah diolah menjadi perintah *mapping* dimana perintah sudah dikombinasikan dengan perintah terakhir yang diberikan pada robot sehingga robot mengetahui posisi terakhir robot dan posisi yang akan dituju oleh robot hanya dengan serial *Bluetooth* yang tersedia.

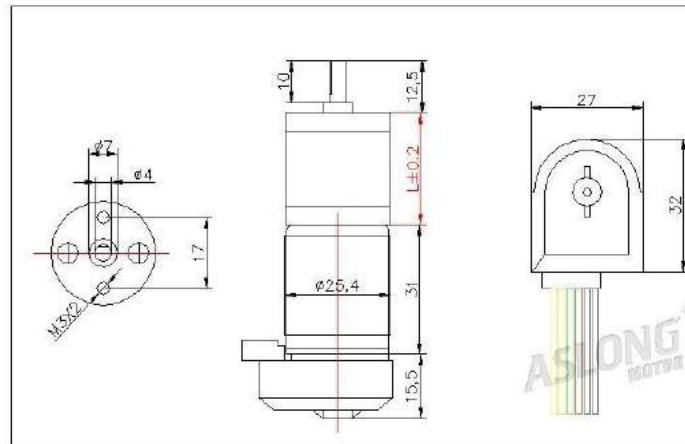
### **3.7. Perancangan Sistem**

Pada subbab ini dijelaskan tentang penjelasan rangkaian secara menyeluruh pada *mobile* robot dengan rincian yaitu spesifikasi motor, rangkaian *driver* motor, rangkaian modul *Bluetooth*, rangkaian *encoder*, rangkaian *LCD*, sistem android dan pola lintasan

#### **3.7.1. Spesifikasi Motor**

Pada penelitian ini menggunakan motor dengan *rotary encoder* dengan tipe motor JGA25-371-34. Motor ini menggunakan tegangan *input* sebesar 6-12 volt dengan *gear ratio* 1:34. Untuk gambar spesifikasi motor dan gear ratio motor dapat dilihat pada gambar 3.7 dan tabel 3.1





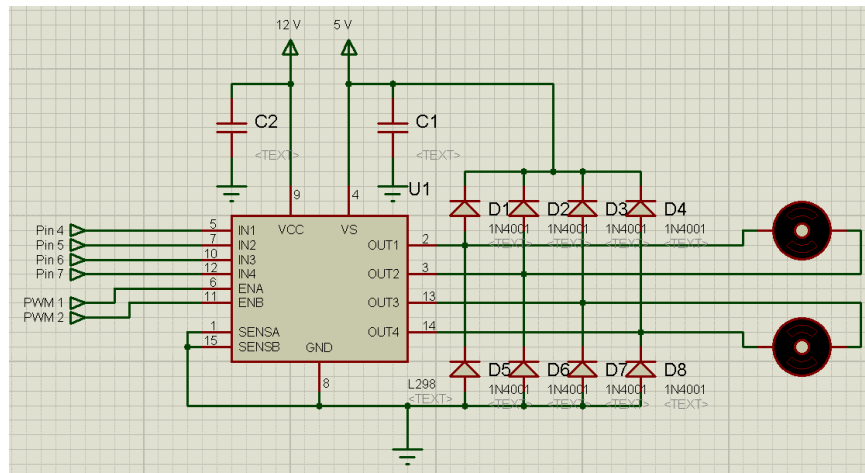
Gambar 3.7. Dimensi Motor

Tabel 3.1 *Gear Ratio* Motor

Gear ratio 1:XX	DC Volt.V	rpm/m	ma	rpm/m	ma	kg.cm	W	ma	kg.cm
1: 21	6V	150	30	70	280	0.4	1.25	900	1.3
	12V	201	46	168	250	0.53	1.25	1000	2.65
1: 34	6V	75	30	60	280	0.8	1.25	900	2.5
	12V	126	46	100	250	1.2	1.25	1000	4.2
1: 103	6V	22	30	18	280	1.9	1.25	1000	6
	12V	41	46	32	150	2.5	1.25	1200	12.5
Motor Gearratio 1:21,1:34 and 1:103 are available, please choose the one you need.									

### 3.7.2. Rangkaian *Driver* Motor

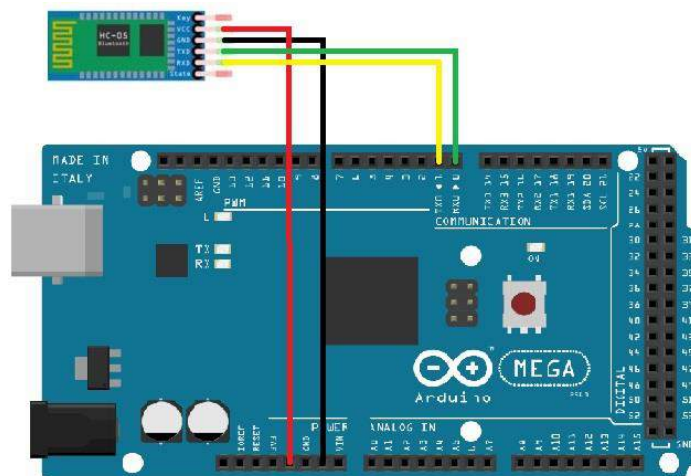
Penelitian ini menggunakan *driver* motor L298D dengan arus maksimal sebesar 2A. Pada tegangan kerja *driver* (Vs) menggunakan tegangan sebesar 5 volt dan tegangan kerja motor menggunakan tegangan sebesar 12 volt. Untuk gambar rangkaian *driver* motor dapat dilihat pada gambar 3.8.



Gambar 3.8. Rangkaian *Driver Motor*

### 3.7.3. Rangkaian Modul *Bluetooth*

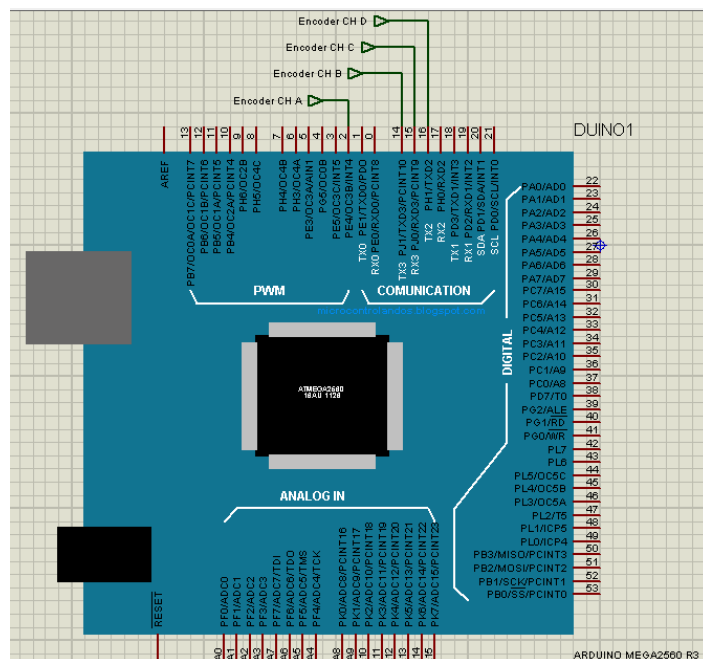
Penelitian ini menggunakan modul *Bluetooth* dengan tipe HC 05. Modul *Bluetooth* ini menggunakan tegangan kerja sebesar 5 volt dan pin TXD dan pin RXD untuk komunikasi serial. Untuk dapat menggunakan komunikasi serial tersebut harus menghubungkan secara terbalik yaitu TXD dengan RX0 dan RXD dengan TX0 . Untuk gambar rangkaian modul *bluetooth* dapat dilihat pada gambar 3.9.



Gambar 3.9. Rangkaian Modul *Bluetooth*

### 3.7.4. Rangkaian Encoder

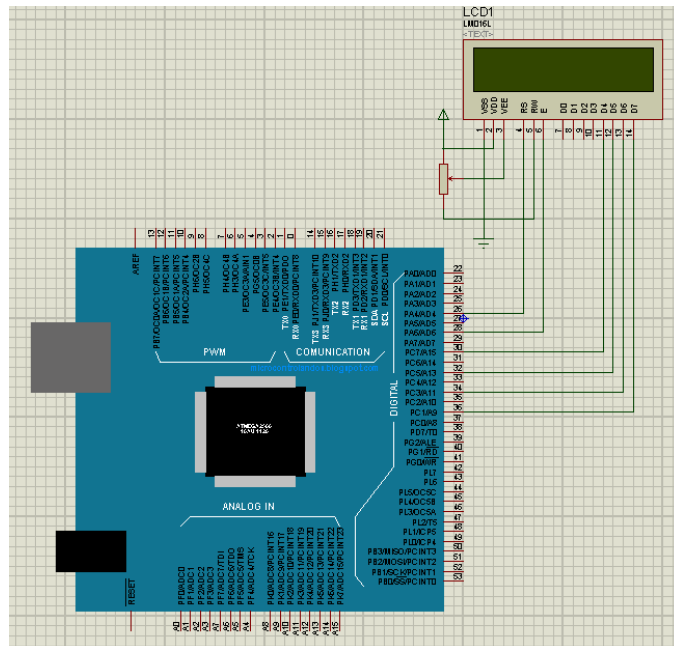
Pada penelitian ini menggunakan *encoder* dengan spesifikasi 12.000 ppr yaitu menghasilkan 12.000 pulsa tiap sekali putaran. Setiap motor, memiliki 2 *channel encoder* jadi setiap pasang motor memiliki 4 *channel encoder* namun pada penelitian ini hanya digunakan 1 *channel* yaitu *channel A* yang dihubungkan dengan pin 2. Untuk gambar rangkaian encoder dapat dilihat pada gambar 3.10.



Gambar 3.10. Rangkaian Encoder

### 3.7.5. Rangkaian LCD

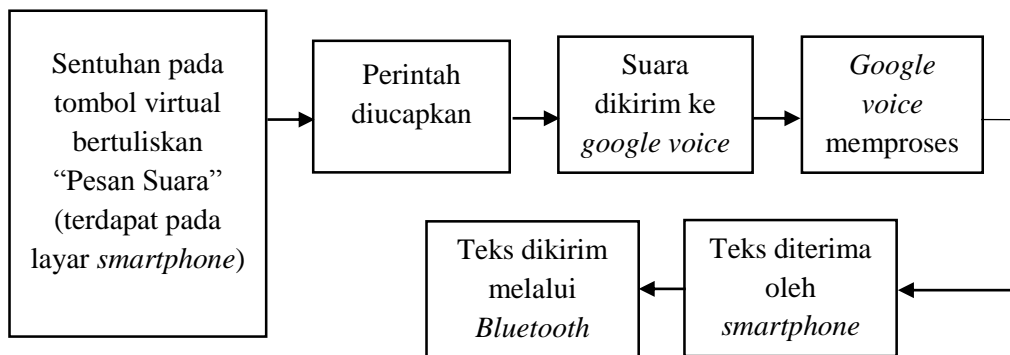
Pada penelitian ini menggunakan *display* LCD 16x2 yang dapat menampilkan 32 karakter pada layar LCD. Pin Rs dan E pada LCD dihubungkan dengan pin 26 dan 28. Pada pin D4-D7 dihubungkan dengan pin 30, 32, 34, 36. Untuk gambar rangkaian LCD dapat dilihat pada gambar 3.11.



Gambar 3.11. Rangkaian LCD

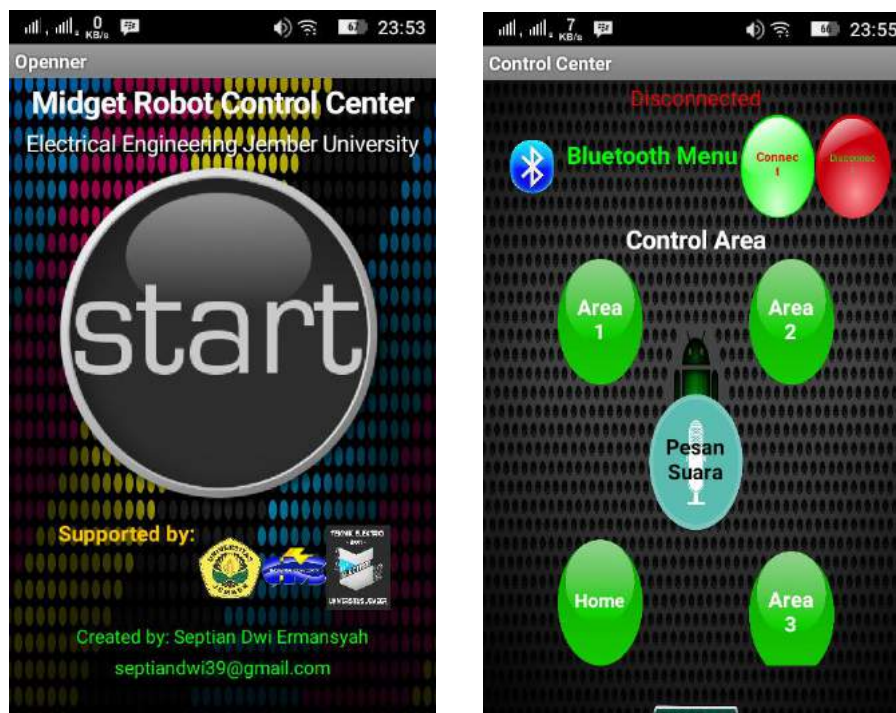
**3.7.6. Sistem Android**

Aplikasi yang telah dibuat dengan menggunakan *app inventor* memiliki 2 *input* yaitu *input* sentuhan yang bertuliskan Pesan Suara yang mempunyai fungsi untuk memberikan perintah dengan perintah suara dan Area 1, Area 2, Area 3, dan Home yang mempunyai fungsi hanya mengirimkan perintah yang telah diprogramkan.



Gambar 3.12. Proses *Input* Perintah Suara

Pada gambar 3.12 merupakan proses *input* menggunakan perintah suara. Langkah pertama yaitu menyentuh tombol yang bertuliskan “Pesan Suara” sehingga muncul notifikasi aplikasi siap menerima perintah suara. Setelah perintah diterima oleh *smartphone* maka suara tersebut dikirimkan ke *Google Voice* melalui akses internet untuk dirubah menjadi sebuah teks. Setelah dirubah menjadi teks, maka teks tersebut dikirimkan kembali melalui akses internet ke *smartphone*. Setelah *smartphone* menerima teks perintah maka *smartphone* akan mengirimkan perintah ke modul *Bluetooth* yang terintegrasi dengan arduino menggunakan jaringan *Bluetooth*. Berikut adalah tampilan aplikasi arduino tersebut:



a. Screen 1

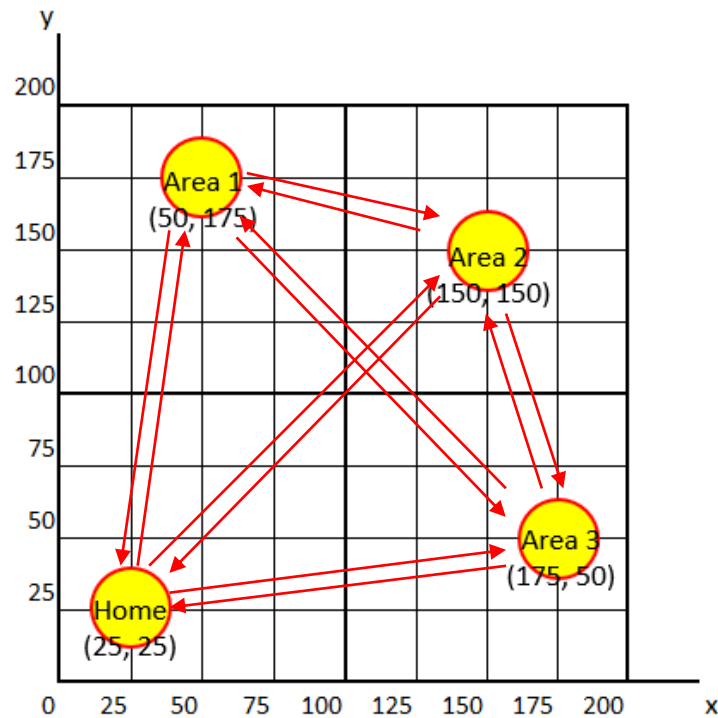
b. Screen 2

Gambar 3.13. Tampilan Aplikasi Kontrol Robot

### 3.7.7. Pola Lintasan

Pada penelitian ini menggunakan lintasan berukuran 2m x 2m dengan 4 titik pos yaitu pos *home* yang berada di titik (25,25), pos area 1 yang berada dititik (50, 175), pos area 2 yang berada dititik (150, 150), dan pos area 3 yang berada dititik (175,50). Pada lintasan ini terdapat 12 pola lintasan yaitu saat *home* menuju area 1,

saat area 1 menuju *home*, saat *home* menuju area 2, saat area 2 menuju *home*, saat *home* menuju area 3, saat area 3 menuju *home*, saat area 1 menuju area 2, saat area 2 menuju area 1, saat area 1 menuju area 3, saat area 3 menuju area 1, saat area 2 menuju area 3, dan saat area 3 menuju area 2.



Gambar 3.14. Pola Lintasan

### 3.8. Pengujian Sistem

Pada subbab ini dijelaskan tentang pengujian rangkaian secara menyeluruh pada *mobile* robot dengan rincian yaitu pengujian sensor rotary encoder, pengujian sensor kompas, dan pengujian voice recognition android ke robot.

#### 3.8.1. Pengujian Sensor Rotary Encoder

Merujuk pada bab 2 poin “Sensor *Rotary Encoder*” digunakan untuk menghasilkan beberapa pulsa dalam setiap putaran motor. Setiap putaran motor akan dikonversikan menjadi jarak dan pulsa akan dikonversikan menjadi jarak tempuh robot. Pada gambar 3.15 dapat diketahui bahwa sensor *rotary encoder* dapat bekerja dengan baik, hal tersebut dibuktikan dengan perpindahan robot pada program sejauh 50 cm dan perpindahan robot diukur oleh meteran sejauh 50 cm



Gambar 3.15 Pengujian Sensor *Rotary Encoder*

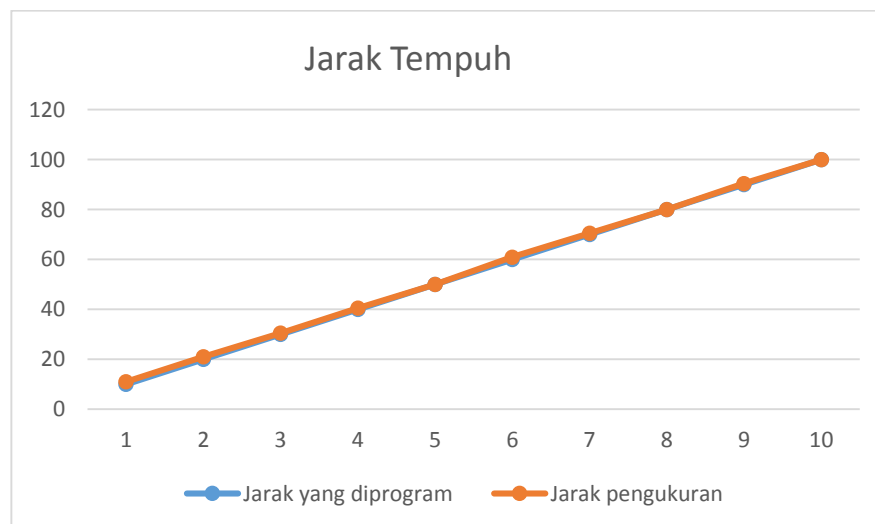
Tabel 3.2 berikut adalah hasil pengukuran kalibrasi sensor *rotary encoder* dengan hasil pengukuran menggunakan mistar. Untuk menentukan *error %* dari hasil pengujian sensor *rotary encoder* maka menggunakan persamaan 3.1:

$$E\% = \frac{\text{Hasil teori} - \text{Hasil praktik}}{\text{Hasil teori}} \times 100\% \dots \dots \dots (3.1)$$

Berdasarkan tabel 3.2 terdapat 10 kondisi jarak antara pengukuran kalibrasi sensor *rotary encoder* dengan pengukuran dengan mistar terlihat selisihnya tidak begitu jauh. Dengan *error %* minimal 0%, maksimal 10% dan rata-rata 2,08%. Pada gambar 3.16 terdapat perbedaan grafik jarak kalibrasi sensor *rotary encoder* dengan pengukuran jarak yang ditempuh menggunakan mistar. Dalam melakukan pengukuran menggunakan satuan *centimeter*.

Tabel 3.2 Pengujian Sensor *Rotary Encoder*

No	Jarak yang di program (cm)	Jarak pengukuran (cm)	Error %
1.	10 cm	11 cm	10%
2.	20 cm	21 cm	5%
3.	30 cm	30.5 cm	1.66%
4.	40 cm	40.5 cm	1.25%
5.	50 cm	50 cm	0%
6.	60 cm	61 cm	1.66%
7.	70 cm	70.5 cm	0.71%
8.	80 cm	80 cm	0%
9.	90 cm	90.5 cm	0.55%
10.	100 cm	100 cm	0%
Rata-Rata			2.08%

Gambar 3.16 Grafik Hasil Pengujian Sensor *Rotary Encoder*



### 3.8.2. Pengujian Sensor Kompas

Pada gambar 3.17 dapat diketahui bahwa sensor kompas HMC5883L dapat bekerja dengan baik, hal tersebut dibuktikan dengan keadaan robot yang semula berputar ke kiri dan harus berhenti pada sudut  $0^\circ$  namun dalam pengujian robot berhenti pada sudut  $359^\circ$  dengan nilai *error* persen 0,28%



Gambar 3.17 Pengujian Sensor Kompas

Dalam proyek akhir ini sensor menggunakan sensor kompas GY-273 dengan inti sensor HMC5883L yang merupakan IC kompas digital 3 axis yang memiliki *interface* I2C.

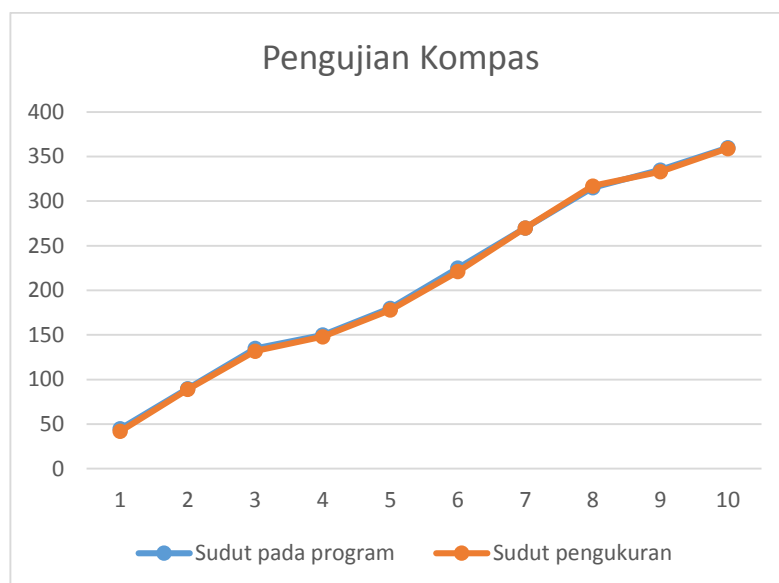
Dari data tabel 3.4 berikut terlihat bahwa antara program dengan hasil pengukuran memiliki selisih yang tidak jauh berbeda. Dengan *error* persen minimal 0%, maksimal 1,11%, dan rata-rata 0,56%. Dalam melakukan pengukuran menggunakan satuan derajat.

Tabel 3.3 Pengujian Sensor Kompas

No	Sudut pada program (derajat)	Sudut pengukuran (derajat)	<i>Error</i> %
1	$45^\circ$	$42^\circ$	0.83%
2	$90^\circ$	$89^\circ$	0.28%
3	$135^\circ$	$132^\circ$	0.83%
4	$150^\circ$	$148^\circ$	0.56%
5	$180^\circ$	$178^\circ$	0.56%

Tabel 3.3 Pengujian Sensor Kompas

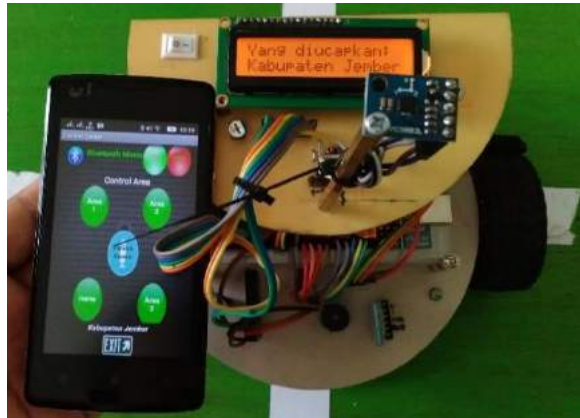
6	225 °	221 °	1.11%
7	270 °	270 °	0%
8	315 °	317 °	0.56%
9	335 °	333 °	0.56%
10	360 °	359 °	0.28%
Rata-rata			0.56%



Gambar 3.18 Grafik Hasil Pengujian Sensor Kompas

### 3.8.3. Pengujian *Voice Recognition* Android ke Robot

Pada pengujian kali ini bertujuan untuk mengetahui apakah kata-kata yang diucapkan oleh seseorang dapat dikonversi dengan baik menjadi sebuah *text* dan akan dikirimkan ke arduino untuk diolah menjadi perintah dengan bantuan koneksi *Bluetooth*. Pengujian ini dilakukan dengan cara mengucapkan 10 kata secara bergantian sebanyak 10 kali dengan kata pengujian seperti pada tabel 3.4.



Gambar 3.19 Pengujian *Voice Recognition* pada Robot

Pada pengujian kata “Area 1” dan “Go go go” mendapatkan *error* paling tinggi yaitu sebesar 20% dan 30%. Hal tersebut terjadi karena kesalahan mengucapkan kata dengan tidak begitu jelas dan kegagalan *google voice recognition* untuk mengubah ucapan menjadi sebuah *text*. Rata-rata *error* yang terjadi namun masih tergolong rendah karena memiliki rata-rata *error* persen sebesar 6%.

Tabel 3.4 Pengujian *Voice Recognition*

No	Kata Pengujian	10 Kali Percobaan		<i>Error %</i>
		Berhasil	Gagal	
1	Area 1	8	2	20%
2	Area 2	10	0	0%
3	Area 3	9	1	10%
4	<i>Home</i>	10	0	0%
5	Universitas Jember	10	0	0%
6	Teknik Elektro	10	0	0%
7	Kabupaten Jember	10	0	0%
8	Robot Cerdas	10	0	0%
9	<i>Together Forever</i>	10	0	0%
10	<i>Go go go</i>	7	3	30%
Rata-rata				6%

### 3.8.4. Pengujian Komunikasi *Bluetooth* terhadap Jarak

Pada pengujian kali ini bertujuan untuk mengetahui berapa jarak ideal komunikasi maksimal *Bluetooth* antara robot dengan *smartphone* android. Pengujian dilakukan dengan dengan cara memberikan perintah kepada robot dengan jarak 1 sampai 12 meter untuk menguji jarak ideal komunikasi maksimal *Bluetooth*. Maka didapatkan data seperti tabel 3.5.

Tabel 3.5 Pengujian Komunikasi *Bluetooth*

No	Jarak	Tanpa Penghalang	Dengan Penghalang
1	1 meter	Berhasil	Berhasil
2	2 meter	Berhasil	Berhasil
3	3 meter	Berhasil	Berhasil
4	4 meter	Berhasil	Berhasil
5	5 meter	Berhasil	Berhasil
6	6 meter	Berhasil	Berhasil
7	7 meter	Berhasil	Berhasil
8	8 meter	Berhasil	Berhasil
9	9 meter	Berhasil	Gagal
10	10 meter	Berhasil	Gagal
11	11 meter	Berhasil	Gagal
12	12 meter	Gagal	Gagal

## BAB 5. PENUTUP

### 5.1. Kesimpulan

Berdasarkan pengujian yang telah dilakukan dari penelitian dengan judul “Implementasi Sistem *Voice Recognition* dan *Rotary Encoder* pada *Mobile Robot* sebagai Sistem Navigasi dan Perhitungan Posisi Robot” maka dapat ditarik kesimpulan sebagai berikut.

1. Kecepatan waktu respon tercepat adalah 2 detik dan waktu respon terlama adalah 10 detik. (lihat tabel 4.5 hal 46 dan tabel 4.3 hal 45)
2. Waktu tempuh robot tercepat sebesar 12 detik dan waktu tempuh terlama sebesar 28 detik. (lihat tabel 4.8 hal 48 dan tabel 4.6 hal 47)
3. Error ketepatan koordinat X terjauh adalah 14 cm dengan error persen 9,4 % sedangkan error ketepatan koordinat Y terjauh adalah 9 cm dengan error persen 6 % sehingga error rata-rata dari seluruh data sebesar 3,54%. (lihat tabel 4.5 hal 46, tabel 4.2 hal 44, dan tabel 4.13 hal 51)
4. Ketepatan robot sebesar 96,46%.

### 5.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat diberikan untuk lebih menyempurnakan hasil penelitian ini atau untuk dikembangkan lebih lanjut yakni antara lain:

1. Sensor kompas sebaiknya menggunakan yang lebih presisi dan tidak mudah terpengaruh oleh medan magnet disekitar sensor.
2. Untuk motor dengan sensor *rotary encoder* sebaiknya menggunakan *encoder* dengan ppr sedang (antara 30-300 ppr) untuk menghindari *lost* pembacaan sensor.

## DAFTAR PUSTAKA

- Ardi, Jatra Kurnia. 2014. *Implementasi Sistem Voice Recognition Pada Robot Pemindah Objek Sebagai Sistem Navigasi*. Malang: Skripsi Jurusan Teknik Elektro FT-UB.
- Dwi Septian Suyadhi, Taufiq. 2010. *Buku Pintar Robotika Bagaimana Merancang & Membuat Robot Sendiri*. Yogyakarta: ANDI.
- Hakim, Lukman. 2013. *Implementasi Perhitungan Posisi Robot Dengan FPGA Menggunakan Rotary Encoder*. Surabaya: Skripsi Jurusan Teknologi Industri ITS
- Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik (Sistem Pengaturan)*. Jakarta: Erlangga.
- Patiung, Fajar Timang. 2013. *Rancang Bangun Robot Beroda Dengan Pengendali Suara*. Manado: Skripsi Jurusan Teknik Elektro FT-UNSRAT
- Putra, Darma. 2009. *Sistem Biometrika*. Yogyakarta: Andi
- Saputra, Andri. 2014. *Alat Kendali Lampu Rumah Menggunakan Bluetooth Berbasis Android*. Palembang: Skripsi Jurusan Teknologi dan Informatika STMIK PalComTech
- Saputri, Zaratul Nisa. 2014. *Aplikasi Pengenalan Suara Sebagai Pengendali Peralatan Listrik Berbasis Arduino UNO*. Malang: Skripsi Jurusan Teknik Elektro FT-UB
- Sudiby, Rahardhita Widyatra. 2012. *Pemetaan Posisi Dan Sistem Navigasi Mobile Robot Dalam Ruang Menggunakan Sensor Perpindahan Jenis Optical Laser*. Surabaya: Skripsi Jurusan Teknik Elektro FT-PENS
- Syam, Rafiuddin. 2013. *Dasar Dasar Teknik Sensor*. Makasar: Buku Ajar Teknik Mesin FT-Universitas Hasanuddin

[https://id.wikipedia.org/wiki/Android\\_\(sistem\\_operasi\)](https://id.wikipedia.org/wiki/Android_(sistem_operasi)) diakses pada tanggal 3 Oktober 2015 pukul 19.13

<http://www.geraicerdas.com/mikrokontroler/module/bluetooth-module-hc-05-detail> diakses pada tanggal 3 Oktober 2015 pukul 19.34

<http://assembly-pro.indonetwork.co.id/5438720/bluetooth-hc-05.html> diakses pada tanggal 3 Oktober 2015 pukul 19.57

<http://elektronika-dasar.web.id/teori-elektronika/lcd-liquid-cristal-display/> diakses pada tanggal 3 Oktober 2015 pukul 21.17

<https://mikrokontrolerindonesia.wordpress.com/2014/08/19/data-pin-pinout-lcd-16x2-dengan-konfigurasinya/> diakses pada tanggal 3 Oktober 2015 pukul 21.31

<http://docslide.net/documents/rotary-encoder-55b088610edd6.html> diakses pada tanggal 4 Oktober 2015 pukul 12.45

<https://konversi.wordpress.com/2009/06/12/sekilas-rotary-encoder/> diakses pada tanggal 4 Oktober 2015 pukul 13.25

## LAMPIRAN

### 1. Program Arduino Mega

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(26, 28, 30, 32, 34, 36);
#include <Wire.h>
#include <HMC5883L.h>
const byte simbol_derajat = B11011111; //Simbol Degree
int pulsa = 0;

String perintah = "";
String last_perintah = "home";

String inData = "";
boolean parsing = false;

HMC5883L compass;
int error = 0;
int sudut = 0;
int p1 = 10;
int p2 = 11;

void maju(){
    digitalWrite(4,HIGH);
    digitalWrite(5,LOW);
    digitalWrite(6,HIGH);
    digitalWrite(7,LOW);
    analogWrite(p1,100);
    analogWrite(p2,100);
}

void belok_kanan(){
    digitalWrite(4,LOW);
```



```
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
analogWrite(p1,85);
analogWrite(p2,85);
}

void belok_kiri(){
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,LOW);
digitalWrite(7,HIGH);
analogWrite(p1,85);
analogWrite(p2,85);
}

void berhenti(){
digitalWrite(4,HIGH);
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
digitalWrite(7,HIGH);
analogWrite(p1,0);
analogWrite(p2,0);
}

void buzzer(){
digitalWrite(17,HIGH);
delay(100);
digitalWrite(17,LOW);
delay(100);
digitalWrite(17,HIGH);
delay(100);
digitalWrite(17,LOW);
```

```
    delay(100);
    digitalWrite(17,HIGH);
    delay(100);
    digitalWrite(17,LOW);
    delay(100);
}

void buzzer1(){
    digitalWrite(17,HIGH);
    delay(100);
    digitalWrite(17,LOW);
    delay(100);
}

void menuju(){
    lcd.clear();
    lcd.setCursor(2,0);
    lcd.print("robot menuju");
    lcd.setCursor(4,1);
    lcd.print(perintah);
}

void sampai(){
    lcd.clear();
    lcd.setCursor(2,0);
    lcd.print("robot sampai");
    lcd.setCursor(5,1);
    lcd.print(perintah);
}

void maju_utara(){
    maju();
    delay(10);
```

```
akses_sudut ();
if (sudut < 1 && sudut > 359) //218
{
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
analogWrite(p1,150);
analogWrite(p2,150);
}
else if (sudut >=1 && sudut < 50)
{
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
analogWrite(p1,150);
analogWrite(p2,130);
}
else if (sudut <= 359 && sudut > 300)
{
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
analogWrite(p1,130);
analogWrite(p2,150);
}
}

void maju_selatan(){
  maju();
```

```
delay(10);
akses_sudut ();
if (sudut < 226 && sudut > 224) //218
{
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
analogWrite(p1,150);
analogWrite(p2,150);
}
else if (sudut >= 226)
{
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
analogWrite(p1,150);
analogWrite(p2,130);
}
else if (sudut <= 224)
{
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
analogWrite(p1,130);
analogWrite(p2,150);
}
}

void maju_timur(){
```

```
maju();
delay(10);
akses_sudut ();
if (sudut < 83 && sudut > 81) //84
{
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
analogWrite(p1,150);
analogWrite(p2,150);
}
else if (sudut >= 83)
{
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
analogWrite(p1,150);
analogWrite(p2,130);
}
else if (sudut <= 81)
{
digitalWrite(4,HIGH);
digitalWrite(5,LOW);
digitalWrite(6,HIGH);
digitalWrite(7,LOW);
analogWrite(p1,130);
analogWrite(p2,150);
}
}
```

```
void maju_barat(){
    maju();
    delay(10);
    akses_sudut ();
    if (sudut < 305 && sudut > 303) //301
    {
        digitalWrite(4,HIGH);
        digitalWrite(5,LOW);
        digitalWrite(6,HIGH);
        digitalWrite(7,LOW);
        analogWrite(p1,150);
        analogWrite(p2,150);
    }
    else if (sudut >= 305)
    {
        digitalWrite(4,HIGH);
        digitalWrite(5,LOW);
        digitalWrite(6,HIGH);
        digitalWrite(7,LOW);
        analogWrite(p1,150);
        analogWrite(p2,130);
    }
    else if (sudut <= 303)
    {
        digitalWrite(4,HIGH);
        digitalWrite(5,LOW);
        digitalWrite(6,HIGH);
        digitalWrite(7,LOW);
        analogWrite(p1,130);
        analogWrite(p2,150);
    }
}
```

```

    }
}

void init_HMC(){
    Wire.begin();
    compass = HMC5883L();
    error = compass.SetScale(1.3);
    if(error != 0)
        compass.GetErrorText(error);

    error = compass.SetMeasurementMode(Measurement_Continuous);
    if(error != 0) // If there is an error, print it out.
        compass.GetErrorText(error);
}

void akses_sudut(){
    MagnetometerRaw raw = compass.ReadRawAxis();
    MagnetometerScaled scaled = compass.ReadScaledAxis();
    int MilliGauss_OnThe_XAxis = scaled.XAxis;
    float heading = atan2(scaled.YAxis, scaled.XAxis);
    float declinationAngle = 0.0457;
    heading += declinationAngle;

    if(heading < 0)
        heading += 2*PI;

    if(heading > 2*PI)
        heading -= 2*PI;

    int headingDegrees = heading * 180/M_PI;
    sudut = headingDegrees;
}

```

```
void setup() {
  Serial.begin(9600);
  lcd.begin(16,2);
  init_HMC();
  attachInterrupt(0,rutinInterupsi0,FALLING);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(p1, OUTPUT);
  pinMode(p2, OUTPUT);
  pinMode(17, OUTPUT);
  start_awal();
}

void loop() {
  terima_serialBT();
  mapping();
}

void start_awal(){
  lcd.setCursor(3,0);
  lcd.print("Welcome to");
  lcd.setCursor(2,1);
  lcd.print("Midget Robot");
  delay(3000);
  lcd.clear();
  lcd.setCursor(3,0);
  lcd.print("Septian Dwi E");
```



```
lcd.setCursor(4,1);
lcd.print("111910201099");
delay(3000);
lcd.clear();
lcd.setCursor(2,0);
lcd.print("Teknik Elektro");
lcd.setCursor(4,1);
lcd.print("Univ. Jember");
delay(3000);

sudut = 360;
while(sudut > 10){
  akses_sudut();
  belok_kiri();
  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("Mencari Arah");
  lcd.setCursor(6,1);
  lcd.print(sudut);
  lcd.write(simbol_derajat);
  delay(100);
}
berhenti();
buzzer();
delay(1000);
lcd.clear();
lcd.setCursor(3,0);
lcd.print("TERDETEKSI");
delay(1500);
lcd.clear();
lcd.setCursor(3,0);
```

```
lcd.print("Robot Siap");
lcd.setCursor(3,1);
lcd.print("Dijalankan");
delay(100);

}

void belok_kanan_dengansudut(int x){
  while(1){
    akses_sudut();
    belok_kanan();
    if(sudut > x-6 && sudut < x+6){
      break;
    }
    lcd.setCursor(13,1);
    lcd.print(sudut);
    lcd.write(simbol_derajat);
    delay(100);
  }
  berhenti();
  buzzer1();
  delay(500);
}

void belok_kiri_dengansudut(int x){
  while(1){
    akses_sudut();
    belok_kiri();
    if(sudut > x-6 && sudut < x+6){
      break;
    }
    lcd.setCursor(13,1);
```

```
    lcd.print(sudut);
    lcd.write(simbol_derajat);
    delay(100);
}
berhenti();
buzzer1();
delay(500);
}
void utara_denganjarak(int x){
    unsigned int distance = x*590;
    pulsa = 0;
    while(pulsa < distance){
        maju_utara();
        akses_sudut();
        lcd.setCursor(13,1);
        lcd.print(sudut);
        lcd.write(simbol_derajat);
    }
    pulsa=0;
    berhenti();
    buzzer1();
    delay(10);
}

void selatan_denganjarak(int x){
    unsigned int distance = x*590;
    pulsa = 0;
    while(pulsa < distance){
        maju_selatan();
        akses_sudut();
        lcd.setCursor(13,1);
```

```
lcd.print(sudut);
lcd.write(simbol_derajat);
}
pulsa=0;
berhenti();
buzzer1();
delay(10);
}

void timur_denganjarak(int x){
  unsigned int distance = x*590;
  pulsa = 0;
  while(pulsa < distance){
    maju_timur();
    akses_sudut();
    lcd.setCursor(13,1);
    lcd.print(sudut);
    lcd.write(simbol_derajat);
  }
  pulsa=0;
  berhenti();
  buzzer1();
  delay(10);
}

void barat_denganjarak(int x){
  unsigned int distance = x*590;
  pulsa = 0;
  while(pulsa < distance){
    maju_barat();
    akses_sudut();
```

```
lcd.setCursor(13,1);
lcd.print(sudut);
lcd.write(simbol_derajat);
}
pulsa=0;
berhenti();
buzzer1();
delay(10);
}

void rutinInterupsi0(){
  pulsa++;
}

void terima_serialBT(){
  while(Serial.available()){
    delay(100);
    char inChar = Serial.read();
    inData += inChar;
    parsing = true;
  }

  if(parsing == true){
    perintah = inData;
    //lcd.clear();
    //lcd.setCursor(8,0);
    //lcd.print(perintah);

    parsing = false;
    inData = "";
  }
}
```

```
}  
  
void mapping(){  
  if(perintah == "home"){  
    if(last_perintah == "home"){  
      buzzer();  
    }else if(last_perintah == "area 1"){  
      menuju();  
      belok_kiri_dengansudut(304);  
      barat_denganjarak(25);  
      belok_kiri_dengansudut(225);  
      selatan_denganjarak(75);  
      selatan_denganjarak(75);  
      sampai();  
      buzzer();  
      belok_kanan_dengansudut(0);  
    }else if(last_perintah == "area 2"){  
      menuju();  
      belok_kiri_dengansudut(304);  
      barat_denganjarak(50);  
      barat_denganjarak(75);  
      belok_kiri_dengansudut(225);  
      selatan_denganjarak(50);  
      selatan_denganjarak(75);  
      sampai();  
      buzzer();  
      belok_kanan_dengansudut(0);  
    }else if(last_perintah == "area 3"){  
      menuju();  
      belok_kiri_dengansudut(304);  
      barat_denganjarak(75);
```

```
    barat_denganjarak(75);  
    belok_kiri_dengansudut(225);  
    selatan_denganjarak(25);  
    sampai();  
    buzzer();  
    belok_kanan_dengansudut(0);  
}  
  
last_perintah = perintah;  
perintah = "";  
  
}else if(perintah == "area 1"){  
    if(last_perintah == "home"){  
        menuju();  
        belok_kanan_dengansudut(0);  
        utara_denganjarak(75);  
        utara_denganjarak(75);  
        belok_kanan_dengansudut(82);  
        timur_denganjarak(25);  
        sampai();  
        buzzer();  
        belok_kiri_dengansudut(0);  
    }else if(last_perintah == "area 1"){  
        buzzer();  
    }else if(last_perintah == "area 2"){  
        menuju();  
        belok_kanan_dengansudut(0);  
        utara_denganjarak(25);  
        belok_kiri_dengansudut(304);  
        barat_denganjarak(100);  
        sampai();
```

```
buzzer();
belok_kanan_dengansudut(0);
}else if(last_perintah == "area 3"){
    menuju();
    belok_kanan_dengansudut(0);
    utara_denganjarak(50);
    utara_denganjarak(75);
    belok_kiri_dengansudut(304);
    barat_denganjarak(50);
    barat_denganjarak(75);
    sampai();
    buzzer();
    belok_kanan_dengansudut(0);
}
```

```
last_perintah = perintah;
perintah = "";
```

```
}else if(perintah == "area 2"){
    if(last_perintah == "home"){
        menuju();
        belok_kanan_dengansudut(0);
        utara_denganjarak(50);
        utara_denganjarak(75);
        belok_kanan_dengansudut(82);
        timur_denganjarak(50);
        timur_denganjarak(75);
        sampai();
        buzzer();
        belok_kiri_dengansudut(0);
    }else if(last_perintah == "area 1"){
```



```
menuju();
belok_kanan_dengansudut(82);
timur_denganjarak(100);
belok_kanan_dengansudut(225);
selatan_denganjarak(25);
sampai();
buzzer();
belok_kiri_dengansudut(0);
}else if(last_perintah == "area 2"){
    buzzer();
}else if(last_perintah == "area 3"){
    menuju();
    belok_kanan_dengansudut(0);
    utara_denganjarak(100);
    belok_kiri_dengansudut(304);
    barat_denganjarak(25);
    sampai();
    buzzer();
    belok_kanan_dengansudut(0);
}
```

```
last_perintah = perintah;
perintah = "";
}else if(perintah == "area 3"){
    if(last_perintah == "home"){
        menuju();
        belok_kanan_dengansudut(82);
        timur_denganjarak(75);
        timur_denganjarak(75);
        belok_kiri_dengansudut(0);
        utara_denganjarak(25);
```

```
sampai();  
buzzer();  
belok_kiri_dengansudut(0);  
}else if(last_perintah == "area 1"){  
  menuju();  
  belok_kanan_dengansudut(82);  
  timur_denganjarak(50);  
  timur_denganjarak(75);  
  belok_kanan_dengansudut(225);  
  selatan_denganjarak(50);  
  selatan_denganjarak(75);  
  sampai();  
  buzzer();  
  belok_kiri_dengansudut(0);  
}else if(last_perintah == "area 2"){  
  menuju();  
  belok_kanan_dengansudut(82);  
  timur_denganjarak(25);  
  belok_kanan_dengansudut(225);  
  selatan_denganjarak(100);  
  sampai();  
  buzzer();  
  belok_kiri_dengansudut(0);  
}else if(last_perintah == "area 3"){  
  buzzer();  
}  
last_perintah = perintah;  
perintah = "";  
}  
  
}
```

## 2. Tampilan Aplikasi Android



a. Screen 1



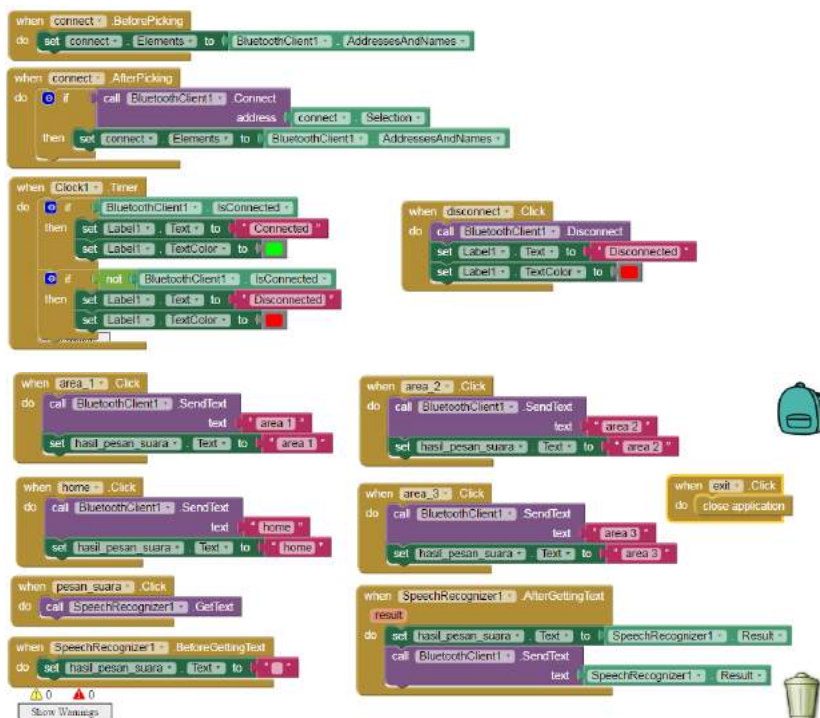
b. Screen 2

## 3. Program Aplikasi Android

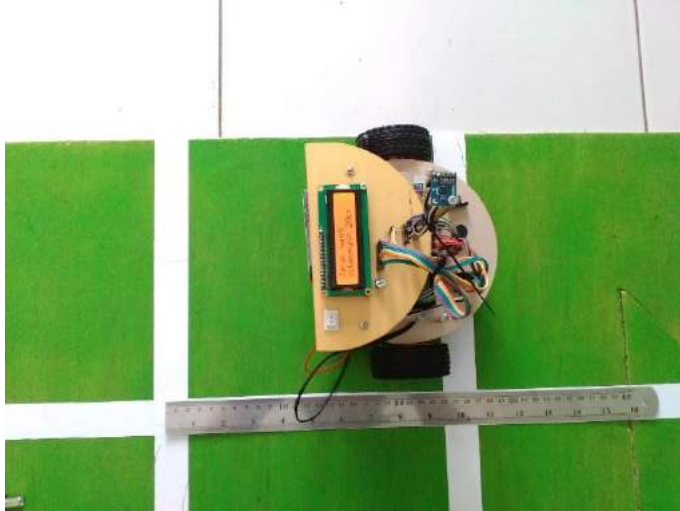
### a) Program Screen 1



### b) Program Screen 2



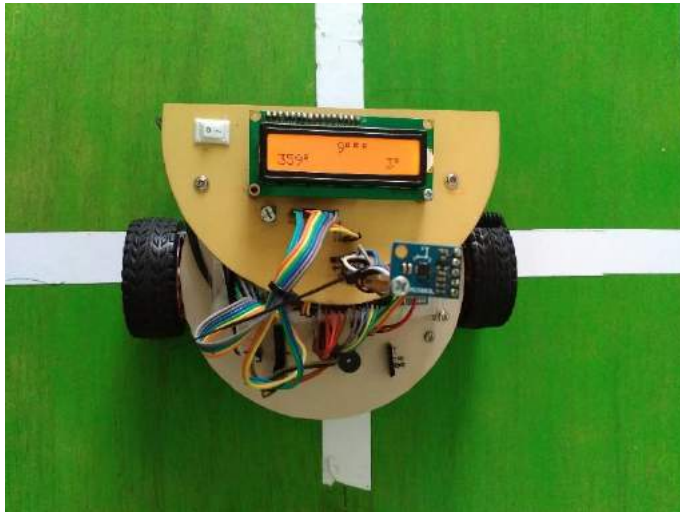
#### 4. Gambar Alat



Gambar kalibrasi jarak



Gambar pengujian *speech recognition*



Gambar pengujian sensor kompas



Gambar lintasan keseluruhan