



**IMPLEMENTASI *GYRODOMETRY FOUR OMNI-DIRECTIONAL ROBOT*  
PADA ROBOT *HYBRID* KONTES ROBOT ABU INDONESIA 2016**

**Skripsi**

Oleh

**Abdur Rokhim**

**NIM 131910201010**

**JURUSAN TEKNIK ELEKTRO STRATA 1  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2017**



**IMPLEMENTASI *GYRODOMETRY FOUR OMNI-DIRECTIONAL ROBOT*  
PADA ROBOT *HYBRID* KONTES ROBOT ABU INDONESIA 2016**

**Skripsi**

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

Oleh

**Abdur Rokhim**

**NIM 131910201010**

**JURUSAN TEKNIK ELEKTRO STRATA 1  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2017**

## PERSEMBAHAN

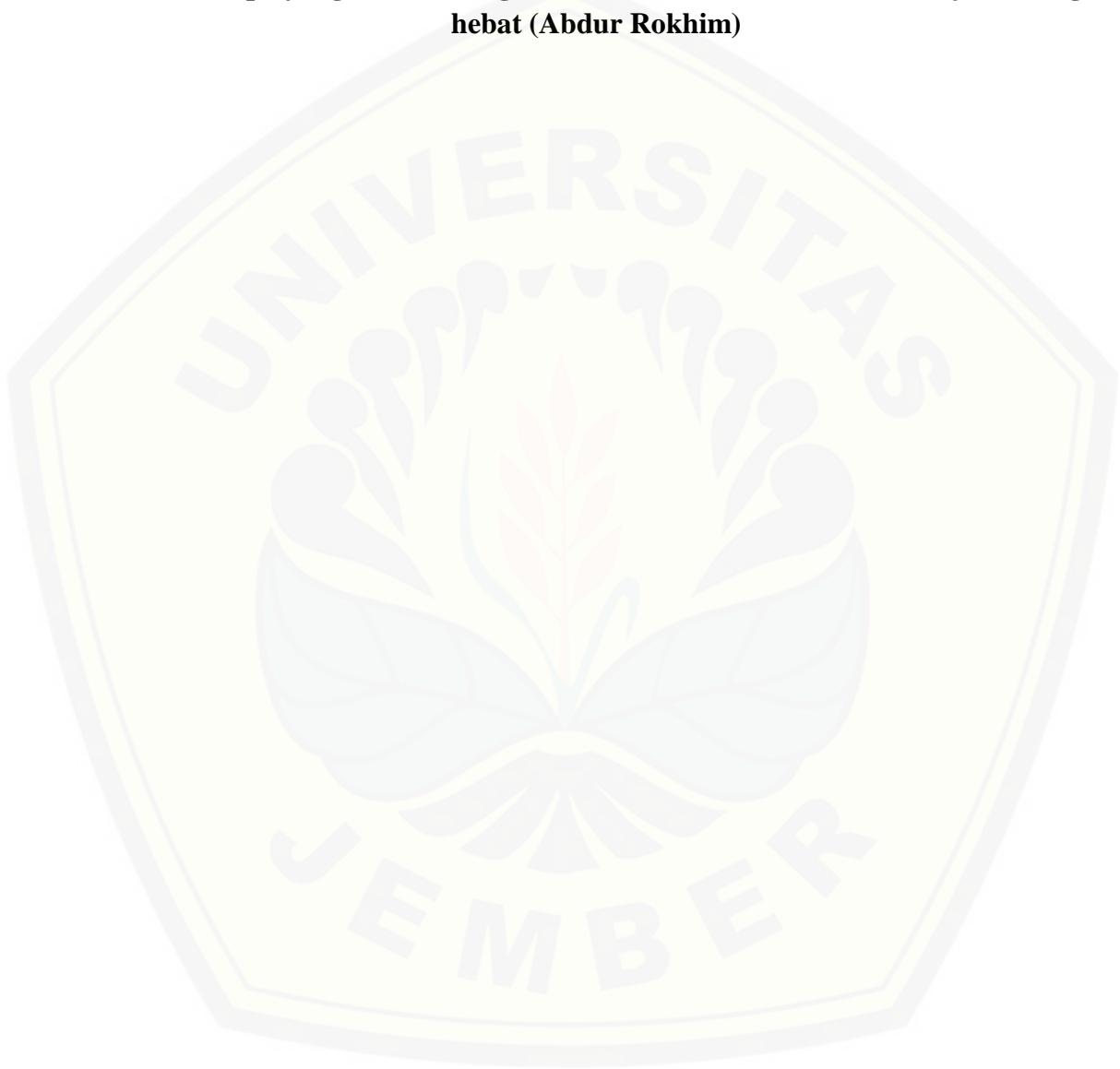
Alhamdulillah, puji syukur kehadiran Allah SWT atas kasih setia-Nya yang telah melimpahkan segala rahmat yang tak ternilai, sehingga saya bisa menyelesaikan penelitian ini. Akhirnya, saya persembahkan skripsi ini kepada:

1. Allah SWT yang Maha Pengasih lagi Maha Penyayang.
2. Nabi besar Muhammad SAW, yang menjadi suri tauladan bagi seluruh umat.
3. Dua orang yang kini telah bersama-Nya, ayahku Nurhadi dan kakak tercinta Abdul Rozak.
4. Ibuku tercinta, Aminatus Sholihah yang telah menjadi seorang yang tidak akan pernah tergantikan dalam hidupku.
5. Saudara-saudaraku tercinta Abdul Ro'uf, Ilmiyah dan Ni'matul Karimah yang menjadi penyemangat serta pembimbing bagiku.
6. Irzunnafis C. C. terima kasih telah menjadi teman, sahabat, kekasih dan teman berantem bagiku. Dan terima kasih untuk kebersamaannya sampai saat ini.
7. Teman-teman INTEL 2013 dan ELKA13 yang telah menjadikan kehidupan dimasa kuliah terasa sangat indah.
8. Ex-pengurus Robotika dan anggota UKM Robotika. Devry, Dika (Kambing), Riski (Bima), Fathor, Rizky, Wandu (Bocil), Mas Wawan, Mas Fajar dan kawan-kawan terimakasih atas kebersamaan, kepedulian serta ilmu yang telah diberikan.
9. Seluruh asisten Laboratorium Teknik Elektro Universitas Jember. Terima kasih telah menjadi lawan diskusi yang menyenangkan.
10. Guru-guruku sejak TK hingga Perguruan Tinggi yang terhormat, terima kasih telah memberikan ilmu dan mendidik dengan penuh kesabaran.
11. Serta seluruh rekan-rekan yang penulis kenal dan rekan-rekan yang membaca skripsi ini. Terima kasih untuk segalanya.

**MOTTO**

**Hal-hal yang mudah akan terasa sulit ketika Anda melakukannya dengan enggan (Terenece)**

**Lakukan apa yang belum orang lain lakukan. Maka kau akan menjadi orang hebat (Abdur Rokhim)**



**PERNYATAAN**

Saya yang bertanda tangan dibawah ini:

Nama : Abdur Rokhim

NIM : 131910201010

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul ”Implementasi *Gyrodometry Four Omni-directional Robot* Pada Robot *Hybrid* Kontes Robot ABU Indonesia 2016” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, 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, 22 Mei 2017

Yang menyatakan,

Abdur Rokhim

NIM 131910201010

**SKRIPSI**

**IMPLEMENTASI *GYRODOMETRY FOUR OMNI-DIRECTIONAL ROBOT*  
PADA ROBOT *HYBRID* KONTES ROBOT ABU INDONESIA 2016**

Oleh  
Abdur Rokhim  
NIM 131910201010

Pembimbing :

Dosen Pembimbing Utama : Sumardi, S.T., M.T.

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

**PENGESAHAN**

Skripsi berjudul “**Implementasi Gyrodometry Four Omni-directional Robot Pada Robot Hybrid Kontes Robot ABU Indonesia 2016**” karya Abdur Rokhim telah diuji dan disahkan pada :

Hari : Senin

Tanggal : 12 Juni 2017

Tempat : Fakultas Teknik Universitas Jember

Tim penguji,

Ketua,

Anggota I,

Sumardi, S.T., M.T.  
NIP 196701131998021001

Bambang Supeno, S.T., M.T.  
NIP 196906301995121001

Anggota II,

Anggota III,

Mohamad Agung P N S.T., M.T.  
NIP 198712172012121003

Khairul Anam S.T., M.T., Ph.D.  
NIP 197804052005011002

Mengesahkan  
Dekan,

Dr. Ir. Entin Hidayah, M.U.M.  
NIP 196612151995032001



## RINGKASAN

**Implementasi Gyrodometry Four Omni-directional Robot Pada Robot Hybrid Kontes Robot ABU Indonesia 2016**; Abdur Rokhim; 131910201010; 2017; 98 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Kontes Robot Indonesia (KRI) merupakan salah satu kompetisi robotika tingkat nasional yang diadakan secara rutin setiap tahun oleh Direktorat Jendral Pendidikan Tinggi. Salah satu divisi Kontes Robot Abu Indonesia (KRAI) 2016 bertemakan “Efisiensi Energi Terbarukan” dimana tema ini selaras dengan tema yang telah diusung oleh ABU Robocon 2016 yaitu “*Clean Energy Recharging The World*”. Pada divisi ini terdiri dari dua robot yakni robot otomatis (*eco robot*) dan robot semi otomatis (*hybrid robot*). Robot otomatis bergerak tanpa menggunakan aktuator penggerak, sehingga diperlukan energi penggerak yakni berupa dorongan udara yang bersumber dari robot semi otomatis.

Oleh karena itu, diperlukan kontrol sistem navigasi dan perencanaan jalur sehingga mempermudah dalam pengendalian robot saat kondisi *manual mode* maupun *automatic mode*. Sistem yang digunakan untuk menentukan *trajectory* dari robot dapat menggunakan sensor garis atau sering kita kenal dengan *line follower robot*. Dengan kemajuan teknologi saat ini sistem navigasi dan *trajectory* dari robot kurang efisien menggunakan media dari luar seperti garis atau dinding ruangan, robot harus dapat memperkirakan posisi relatifnya berdasarkan acuan yang terdapat dari dalam robot sendiri. Tujuan dari penelitian ini yaitu: (1) Dapat membuat sistem kontrol navigasi dan *trajectory* sehingga memperoleh tingkat kesesuaian robot dalam bergerak mengikuti plan yang telah ditentukan. (2) Mendapatkan kontrol robot yang lebih stabil dalam menjalankan plan yang diinginkan. Hasil dari penelitian ini diharapkan pergerakan robot dapat berjalan sesuai dengan *planning* yang diharapkan.

Penelitian ini membahas tentang robot KRAI yang menggunakan 4 roda omni sebagai kemudi serta menggunakan metode *gyrodometry* sebagai kontrol navigasi dan *trajectory* robot. Metode *gyrodometry* merupakan metode



pengambilan data dari sensor *gyroscope* serta mengkombinasikannya dengan data yang diperoleh dari *odometry*, sehingga mendapatkan kontrol trayektori robot yang sesuai dengan *plan* yang diinginkan.



## SUMMARY

***Gyrodometry Four Omni-directional Robot Implementation On Robot Hybrid ABU Indonesia Robot Contest 2016***; Abdur Rokhim; 131910201010; 2017; 98 pages; Department of Electrical Engineering Faculty of Engineering, University of Jember.

*Indonesia Robot Contest (KRI) is one of the national-level robotics competitions held regularly every year by the Directorate General of Higher Education. One of the divisions of Indonesia Abu Robot Contest (KRAI) 2016 has the theme of "Renewable Energy Efficiency" where the theme is in line with the theme set by ABU Robocon 2016, "Clean Energy Recharging The World". In this division consists of two robots of the automatic robot (eco robot) and semi-automatic robot (hybrid robot). The robot automatically moves without using the actuator drive, so it takes the driving energy of the form of air stimulation sourced from semi-automatic robot.*

*Therefore, it is necessary to control the navigation system and the planning of the path so as to facilitate the control of the robot when the condition of manual mode or automatic mode. The system used to determine the trajectory of the robot can use a line sensor or often we are familiar with the line follower robot. With the current technological advances of navigation systems and trajectory of robots less efficient using external media such as lines or walls of the room, the robot must be able to estimate its relative position based on the reference contained from within the robot itself. The purpose of this study are: (1) Can create a navigation control system and trajectory so as to obtain the level of robot suitability in moving following a predetermined plan. (2) Get a more stable robot control in executing the desired plan. The results of this study is expected to move the robot can run in accordance with the expected planning.*

*This paper discusses about KRAI robot that uses 4 wheel omni as steering as well as using method of gyrodometry as navigation control and trajectory robot. Gyrodometry method is a method of taking data from the gyroscope sensor and combine it with the data obtained from the odometry, so get the control of the robot trajectory in accordance with the desired plan.*

## PRAKATA

Puji syukur kehadiran Allah SWT yang maha kuasa atas segalanya, karena dengan ridho, hidayah dan petunjukNya, penulis dapat menyelesaikan skripsi yang berjudul “Implementasi *Gyrodometry Four Omni-directional Robot* Pada Robot *Hybrid* Kontes Robot ABU Indonesia 2016”. 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 Sumardi, S.T., M.T. dan Bapak Bambang Supeno, S.T., M.T. selaku dosen pembimbing yang telah membimbing menyelesaikan tugas akhir ini;
6. Bapak Khairul Anam S.T., M.T., Ph.D. dan Bapak Mohamad Agung P N S.T., M.T. selaku dosen penguji yang sudah memberikan saran untuk memperbaiki tugas akhir ini;
7. Keluarga besar INTEL'UJ 2013, terimakasih telah memberikan arti kekeluargaan yang luar biasa.
8. Keluarga besar Civitas Akademia Jurusan Teknik Elektro Universitas Jember. Serta semua pihak yang tidak dapat disebutkan satu per satu, yang telah mendukung dalam penyelesaian skripsi ini.

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, 17 Mei 2017

Penulis



DAFTAR ISI

	Halaman
<b>HALAMAN SAMPUL .....</b>	<b>i</b>
<b>HALAMAN JUDUL.....</b>	<b>ii</b>
<b>HALAMAN PERSEMBAHAN.....</b>	<b>iii</b>
<b>HALAMAN MOTTO.....</b>	<b>iv</b>
<b>HALAMAN PERNYATAAN .....</b>	<b>v</b>
<b>HALAMAN PEMBIMBING .....</b>	<b>vi</b>
<b>HALAMAN PENGESAHAN .....</b>	<b>vii</b>
<b>RINGKASAN.....</b>	<b>viii</b>
<b>SUMMARY .....</b>	<b>x</b>
<b>PRAKATA .....</b>	<b>xi</b>
<b>DAFTAR ISI.....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR .....</b>	<b>xvi</b>
<b>BAB 1. PENDAHULUAN.....</b>	<b>1</b>
<b>1.1. Latar Belakang .....</b>	<b>1</b>
<b>1.2. Rumusan Masalah.....</b>	<b>2</b>
<b>1.3. Batasan Masalah .....</b>	<b>2</b>
<b>1.4. Tujuan .....</b>	<b>3</b>
<b>1.5. Manfaat .....</b>	<b>3</b>
<b>1.6. Sistematika Penelitian.....</b>	<b>3</b>
<b>BAB 2. TINJAUAN PUSTAKA .....</b>	<b>5</b>
<b>2.1. Teori Gyrodometry .....</b>	<b>5</b>
2.1.1 Teori Odometry .....	6
2.1.2 Teori Gyroscope.....	6
<b>2.2. Omni Directional Drive.....</b>	<b>8</b>
<b>2.3. KRAI (Kontes Robot ABU Indonesia) .....</b>	<b>10</b>
2.3.1 Eco-Robot .....	11
2.3.2 Hybrid-Robot .....	11
<b>2.4. Arduino Mega2560.....</b>	<b>11</b>

2.4.1	Spesifikasi Arduino Mega2560.....	12
<b>2.5.</b>	<b>Sensor .....</b>	<b>14</b>
2.5.1	Sensor <i>gyroscope</i> L3G4200D .....	15
2.5.2	<i>Rotary encoder</i> .....	15
<b>2.6.</b>	<b>Motor DC .....</b>	<b>16</b>
<b>BAB 3. METODOLOGI PENELITIAN .....</b>		<b>19</b>
<b>3.1.</b>	<b>Tempat dan Waktu Penelitian .....</b>	<b>19</b>
<b>3.2.</b>	<b>Alat dan Bahan.....</b>	<b>19</b>
3.4.1	Pembuatan dan Pengujian Sensor Posisi.....	21
3.4.2	Pembuatan dan Pengujian Sensor Gyroscope .....	22
<b>3.5.</b>	<b>Pengujian Robot <i>Omni directional</i>.....</b>	<b>23</b>
<b>3.6.</b>	<b>Pengujian Kontrol <i>Gyrodometry</i> .....</b>	<b>24</b>
<b>3.7.</b>	<b>Perancangan Alat .....</b>	<b>24</b>
<b>BAB 4. HASIL DAN PEMBAHASAN .....</b>		<b>31</b>
<b>4.1.</b>	<b>Pengujian Rangkaian Sensor .....</b>	<b>31</b>
4.1.1	Sensor Posisi .....	31
4.1.2	Pengujian Sensor Gyroscope.....	35
<b>4.2.</b>	<b>Pengujian Robot <i>Omni-directional</i>.....</b>	<b>37</b>
<b>4.3.</b>	<b>Pengujian Kontrol <i>Gyrodometry</i> .....</b>	<b>39</b>
4.3.1	Pengujian Robot Menggunakan Kontrol dan Tanpa Menggunakan Kontrol .....	39
4.3.2	Pengujian Pencapaian Target Berdasarkan Koordinat Kartesian .....	42
4.3.3	Pengujian Robot Menggunakan Kontrol dalam <i>Plan</i> .....	42
<b>BAB 5. KESIMPULAN DAN SARAN .....</b>		<b>47</b>
<b>5.1.</b>	<b>Kesimpulan .....</b>	<b>47</b>
<b>5.2.</b>	<b>Saran .....</b>	<b>47</b>
<b>DAFTAR PUSTAKA .....</b>		<b>49</b>
<b>LAMPIRAN .....</b>		<b>51</b>
<b><i>Listing Program Arduino Mega2560</i> .....</b>		<b>52</b>



**DAFTAR TABEL**

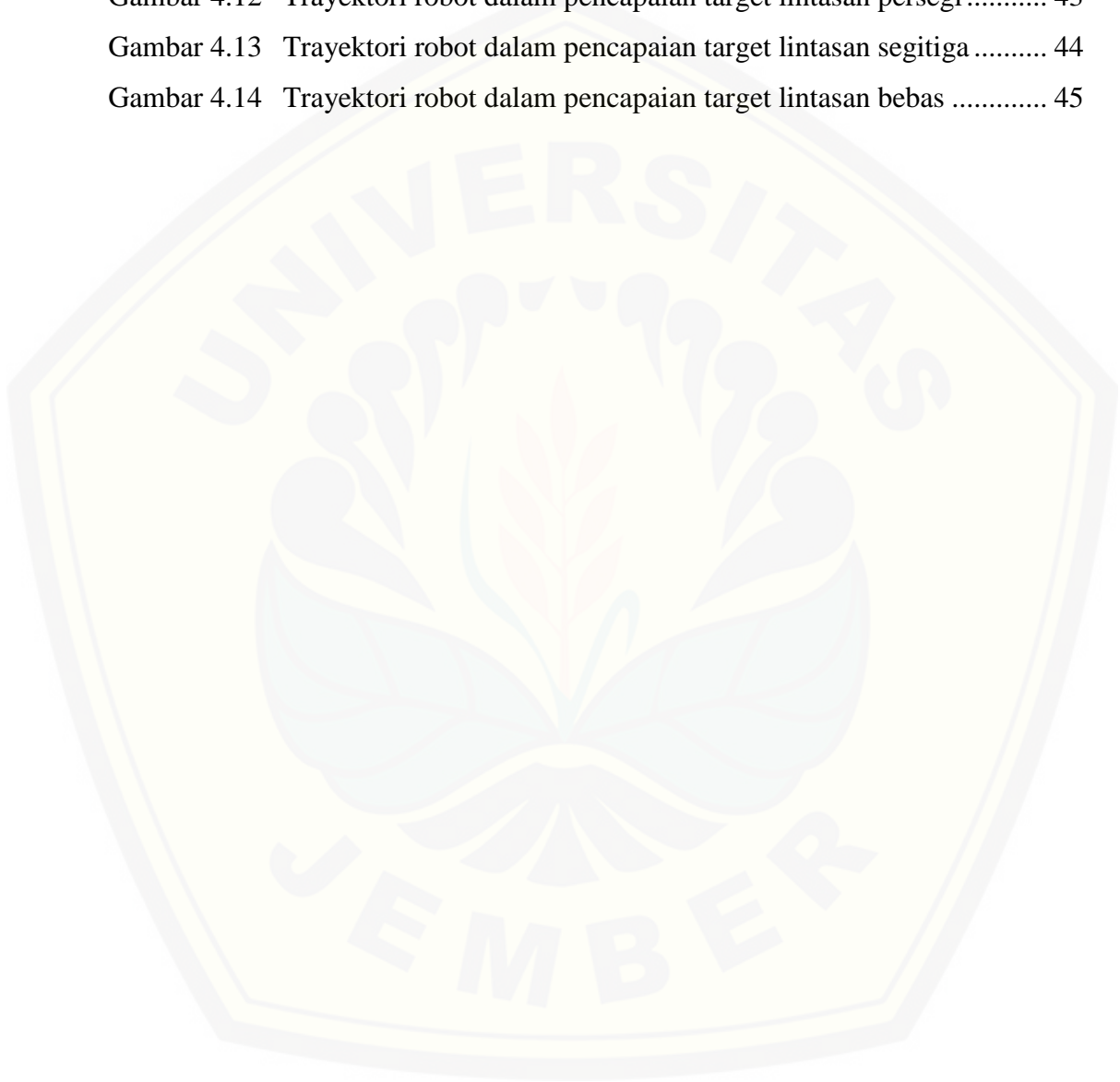
	Halaman
Tabel 3.1 Jadwal Kegiatan Penelitian .....	19
Tabel 3.2 Daftar peralatan yang digunakan .....	20
Tabel 4.1 Hasil pengujian encoder axis X .....	34
Tabel 4.2 Hasil pengujian encoder axis Y .....	35
Tabel 4.3 Hasil pengujian sensor <i>gyroscope</i> .....	36
Tabel 4.4 Hasil pengujian robot <i>omni-directional</i> .....	37
Tabel 4.5 Tabel pengujian robot pada lintasan datar .....	38
Tabel 4.6 Tabel pengujian robot pada lintasan berdebu .....	38
Tabel 4.7 Tabel pengujian robot pada lintasan miring.....	39
Tabel 4.8 Rata-rata simpangan robot dalam variasi PWM maksimal.....	41
Tabel 4.9 Simpangan rata-rata pengujian robot pada lintasan persegi.....	43
Tabel 4.10 Simpangan rata-rata pengujian robot pada lintasan segitiga.....	44
Tabel 4.11 Simpangan rata-rata pengujian robot pada lintasan bebas .....	45



DAFTAR GAMBAR

	Halaman
Gambar 2.1 Grafik hasil pengukuran odometry dan gyro serta hasil dari penggunaan metode gyrodometry. ....	5
Gambar 2.2 Kinematika <i>four wheels omni directional</i> .....	8
Gambar 2.3 Arduino Mega2560.....	12
Gambar 2.4 Diagram Blok L3G4200D .....	15
Gambar 2.5 Prinsip kerja secara umum dari <i>rotary encoder</i> .....	16
Gambar 2.6 Prinsip perputaran motor DC .....	17
Gambar 2.7 Kaidah tangan kiri <i>Fleming</i> .....	18
Gambar 3.1 Diagram Alir Penelitian.....	20
Gambar 3.2 Sensor posisi dan penempatannya .....	21
Gambar 3.3 Rangkaian dan penempatan sensor L3G4200D ke robot .....	22
Gambar 3.4 Sudut roda terhadap sudut target .....	23
Gambar 3.5 Perancangan alat .....	25
Gambar 3.6 Blok Diagram Alat .....	25
Gambar 3.7 Blok diagram sistem kontrol <i>gyrodometry four wheels omni directional</i> .....	26
Gambar 3.8 Blok diagram sistem kontrol <i>four wheels omni directional</i> .....	27
Gambar 3.9 Diagram alir program <i>gyrodometry four wheels omni directional robot</i> . ....	30
Gambar 4.1 Hasil pembacaan sinyal <i>rotary encoder</i> pada <i>oscilloscope</i> saat kondisi <i>increment</i> (V/div = 1 V/div, T/div = 1 ms/div) .....	32
Gambar 4.2 Sampel data sinyal <i>rotary encoder</i> kondisi <i>increment</i> .....	32
Gambar 4.3 Hasil pembacaan sinyal <i>rotary encoder</i> pada <i>oscilloscope</i> saat kondisi <i>decrement</i> (V/div = 1 V/div, T/div = 1 ms/div).....	33
Gambar 4.4 Sampel data sinyal <i>rotary encoder</i> kondisi <i>decrement</i> .....	33
Gambar 4.5 Pengujian sensor posisi.....	34
Gambar 4.6 Pengujian sensor <i>gyroscope</i> .....	36
Gambar 4.7 Grafik pengujian sensor <i>gyroscope</i> .....	37
Gambar 4.8 Trayektori robot dengan kontrol <i>gyrodometry</i> .....	40

Gambar 4.9	Trayektori robot tanpa kontrol <i>gyrodometry</i> .....	40
Gambar 4.10	Trayektori robot dengan kontrol <i>odometry</i> .....	41
Gambar 4.11	Grafik pengujian robot dalam pencapaian target berdasarkan koordinat kartesian .....	42
Gambar 4.12	Trayektori robot dalam pencapaian target lintasan persegi .....	43
Gambar 4.13	Trayektori robot dalam pencapaian target lintasan segitiga .....	44
Gambar 4.14	Trayektori robot dalam pencapaian target lintasan bebas .....	45



## BAB 1. PENDAHULUAN

### 1.1. Latar Belakang

Kontes Robot Indonesia (KRI) merupakan salah satu kompetisi robotika tingkat nasional yang diadakan secara rutin setiap tahun oleh Direktorat Jendral Pendidikan Tinggi. Kompetisi ini dibagi menjadi beberapa divisi yakni Divisi Kontes Robot Abu Indonesia (KRAI), Divisi Kontes Robot Pemadam Api Indonesia (KRPAI) dan Kontes Robot Sepak Bola Indonesia (KRSBI). Masing – masing divisi memiliki aturan, tugas dan arena yang berbeda. Pada Kontes Robot Abu Indonesia (KRAI) peraturan yang digunakan selalu berubah setiap tahun bergantung pada tuan rumah ABU ROBOCON diadakan. (DIKTI, 2016)

Divisi Kontes Robot Abu Indonesia (KRAI) 2016 bertema “Efisiensi Energi Terbarukan” dimana tema ini selaras dengan tema yang telah diusung oleh ABU Robocon 2016 yaitu “Clean Energy Recharging The World”. Pada divisi ini terdiri dari dua robot yakni robot otomatis (*eco robot*) dan robot semi otomatis (*hybrid robot*). Robot otomatis bergerak tanpa menggunakan aktuator penggerak, sehingga diperlukan energi penggerak yakni berupa dorongan udara yang bersumber dari robot semi otomatis. Robot otomatis haruslah dapat melalui medan berupa “*slope and hills*” yang terdiri dari tiga tingkat tanjakan. kemudian berbelok melalui zona “*river*” dan turun melalui zona “*down hill*” sampai mencapai zona “*wind turbin station*” (DIKTI, 2016), sehingga robot semi otomatis harus dapat menyesuaikan posisinya agar dapat memberikan dorongan kepada robot otomatis secara optimal.

Oleh karena itu, diperlukan kontrol sistem navigasi dan perencanaan jalur sehingga mempermudah dalam pengendalian robot saat kondisi *manual mode* maupun *automatic mode*. Sistem yang digunakan untuk menentukan *trajectory* dari robot dapat menggunakan sensor garis atau sering kita kenal dengan *line follower robot*. Dengan kemajuan teknologi saat ini sistem navigasi dan *trajectory*

dari robot kurang efisien menggunakan media dari luar seperti garis atau dinding ruangan, robot harus dapat memperkirakan posisi relatifnya berdasarkan acuan yang terdapat dari dalam robot sendiri (Irfan Affandi, 2014). Acuan ini dapat diperoleh dari sensor yang dapat membaca perpindahan seerta arah kemana robot akan bergerak.

Berdasarkan keadaan di atas maka penulis melakukan penelitian tentang robot KRAI yang menggunakan 4 roda omni sebagai kemudi serta menggunakan metode *gyrodometry* sebagai kontrol navigasi dan *trajectory* robot. Metode *gyrodometry* merupakan metode pengambilan data dari sensor *gyroscope* serta mengkombinasikannya dengan data yang diperoleh dari *odometry* (Irfan Affandi, 2014). Dengan metode ini diharapkan pergerakan robot dapat berjalan sesuai dengan *planning* yang diharapkan.

### 1.2. Rumusan Masalah

Berdasarkan latar belakang diatas maka dapat dirumuskan beberapa masalah sebagai berikut:

1. Bagaimana algoritma *gyrodometry* sistem navigasi dan *trayectory hybrid robot* ?
2. Apakah metode yang digunakan memiliki keunggulan dibandingkan dengan metode yang lain?

### 1.3. Batasan Masalah

Dalam penelitian ini digunakan salah satu tipe *holomic robot* yang menggunakan 4 roda omni dan digerakkan menggunakan motor DC 24 volt, sehingga dalam penelitian ini dilakukan pembatasan permasalahan sebagai berikut :

- a. Sensor yang digunakan yaitu sensor *gyroscope*.
- b. Sistem mikroprosesor yang digunakan berbasis Arduino MEGA 2560.
- c. Motor yang digunakan yakni motor dc 24 volt.
- d. Hanya menggunakan metode *gyrodometry*.
- e. Sistem *trajectory ordometry* hanya menggunakan dua buah *rotary encoder*.

#### 1.4. Tujuan

Berdasarkan dari permasalahan diatas maka tujuan dari penelitian ini adalah sebagai berikut:

- a. Dapat membuat sistem kontrol navigasi dan *trajectory* sehingga memperoleh tingkat kesesuaian robot dalam bergerak mengikuti *plan* yang telah ditentukan.
- b. Mendapatkan kontrol robot yang lebih stabil dalam menjalankan *plan* yang diinginkan.

#### 1.5. Manfaat

Adapun manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

- a. Mempermudah *driver* robot dalam pengontrolan robot saat mode semi otomatis .
- b. Meningkatkan tingkat keoptimalan robot *hybrid* dalam memberikan daya dorong kepada robot otomatis.
- c. Mempersingkat waktu robot dalam menjalankan misi.
- d. Dapat digunakan sebagai referensi dalam sistem kendali robot untuk kontes robot ABU Indonesia (KRAI) maupun robot *omni wheel drive* lainnya.

#### 1.6. Sistematika Penelitian

Secara garis besar penyusunan proposal skripsi ini adalah sebagai berikut:

- a. BAB 1. PENDAHULUAN  
Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika pembahasan.
- b. BAB 2. TINJAUAN PUSTAKA  
Berisi penjelasan tentang teori yang berhubungan dengan penelitian.
- c. BAB 3. METODOLOGI PENELITIAN  
Menjelaskan tentang metode kajian yang digunakan untuk menyelesaikan skripsi.
- d. BAB 4 HASIL DAN PEMBAHASAN  
Berisi hasil penelitian dan analisa hasil penelitian.

e. BAB 5 PENUTUP

Berisi tentang kesimpulan dan saran dari penulis.



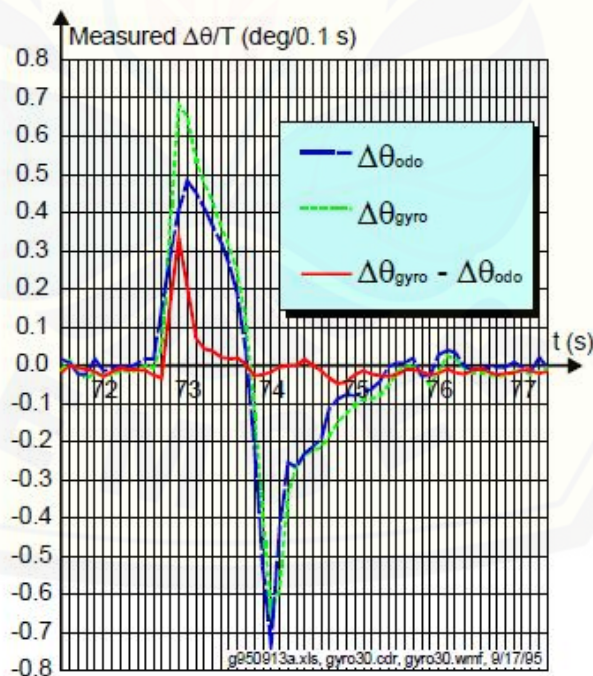


## BAB 2. TINJAUAN PUSTAKA

### 2.1. Teori Gyrodometry

*Gyrodometry* merupakan metode yang menggabungkan data dari sensor *gyro* dengan data yang diperoleh dari odometri. Masalah yang timbul dari penggunaan odometri yakni potensi terjadinya kesalahan *non-systematic* yang diakibatkan adanya benjolan atau permukaan yang tidak rata pada tanah. Robot tidak akan dapat dikalibrasi untuk mengkompensasi *non-systematic error* dan juga tidak dapat memprediksi frekuensi atau besarnya *non-systematic error*. (J. Borenstein and L.Feng, 1996)

Metode *gyrodometry* didasarkan dari hipotesis bahwa perbedaan antara kurva odometri dan kurva *gyro* adalah bernilai tetap dan terjadi pada waktu yang singkat. Pada gambar 2.1 berikut menunjukkan hasil pengukuran odometri dan *gyro* serta hasil dari penggunaan metode *gyrodometry*.



Gambar 2.1 Grafik hasil pengukuran odometry dan gyro serta hasil dari penggunaan metode gyrodometry.  
(J. Borenstein and L.Feng, 1996)



### 2.1.1 Teori *Odometry*

*Odometry* adalah metode kontrol yang menggunakan data dari sensor posisi untuk memperkirakan perubahan posisi dari waktu ke waktu. *odometry* digunakan untuk memperkirakan posisi robot terhadap posisi awal robot bergerak pada koordinat kartesian. (Slamet Basori, 2014)

*Odometry* didasarkan pada asumsi bahwa revolusi roda dapat diterjemahkan ke dalam perpindahan linear relatif ke lantai. Asumsi ini hanya berlaku terbatas. Salah satu contohnya adalah roda selip, jika satu roda itu terpeleset maka encoder pada roda itu akan menghitung revolusi roda meskipun revolusi ini tidak akan sesuai dengan perpindahan linier roda. Selain kasus ekstrem ini total selip, ada beberapa lainnya, alasan yang lebih halus untuk ketidakakuratan dalam terjemahan bacaan roda encoder menjadi gerak linier. Semua sumber kesalahan masuk ke dalam salah satu dari dua kategori yakni kesalahan sistematis dan kesalahan *nonsystematic*. (J. Borenstein, 1996)

Kesalahan sistematis meliputi diameter roda yang tidak sama, roda tidak berbentuk lingkaran sempurna, ketidak pastian titik gesekan pada roda, resolusi encoder yang terbatas, dan juga *sampling rate* yang terbatas. Kesalahan Non-sistematis meliputi lantai yang tidak rata, roda selip dan juga terganjal benda yang tidak terduga.

### 2.1.2 Teori *Gyroscope*

Berdasarkan pada KBBI (Kamus Besar Bahasa Indonesia), kata *gyroscope* memiliki arti sebuah alat berupa piringan yang sumbunya berputar diantara dua titik dan tetap dalam posisinya apabila tidak ada pengaruh dari gaya luar.

Prinsip dasar yang digunakan oleh *gyroscope* untuk mengukur orientasi yakni momentum sudut. Prinsip rotor *gyro* adalah kelakuan dalam ruang atau inersia giroskopik. Hukum pertama Newton menyatakan jika gaya total suatu benda sama dengan nol, maka benda akan diam atau dalam keadaan setimbang. Rotor yang berputar dalam sebuah *gyroscope* akan mempertahankan kondisinya selama tidak ada gaya luar yang mempengaruhi gerakannya. Tingkat stabilitas akan meningkat jika rotor memiliki kecepatan dan massa yang besar.

Karakteristik lain dari *gyroscope* yakni presesi. Presesi adalah gerakan kemiringan atau berputar terhadap sumbu *gyroscope* sebagai akibat gaya yang diterimanya. Dalam persamaan (2.1) berikut menunjukkan perlakuan *gyroscope*

$$\tau = \frac{dL}{dt} = \frac{d(I\omega)}{dt} = I\alpha \dots\dots\dots (2.1)$$

dengan :

= torsi

L = momentum sudut

I = momentum inersia

= kecepatan sudut

= percepatan sudut

Dengan menggunakan prinsip inilah sensor *gyroscope* bekerja. *Gyroscope* mengukur kecepatan sudut dengan acuan inersia benda itu sendiri. Sudut orientasi berupa gerak guling (*roll*), angguk (*pitch*) dan geleng (*yaw/rotation*) diperoleh dengan cara mengintegrasikan kecepatan sudut. (Mochammad Rif'an, dkk 2012)

Berbeda dengan *accelerometer* yang memberikan pengukuran sudut yang akurat ketika sistem dalam keadaan diam (statis) serta sensor kompas yang mengukur gerakan linier dengan acuan medan magnet bumi yang hanya dapat menentukan arah mata angin, keluaran dari *gyro* berupa kecepatan sudut. Kecepatan sudut merupakan besaran vektor yang menunjukkan nilai frekuensi sudut suatu benda terhadap sumbu putarnya. Besar nilai dari kecepatan sudut kemudian dilakukan integrasi terhadap waktu untuk menghitung besar sudut orientasi. Besar nilai sudut orientasi dapat diketahui menggunakan persamaan (2.2)

$$\theta = \theta_0 + \omega t \dots\dots\dots (2.2)$$

dengan :

$\theta$  = sudut yang terbentuk ( $^{\circ}$ )

$\theta_0$  = sudut awal yang terbentuk ( $^{\circ}$ )

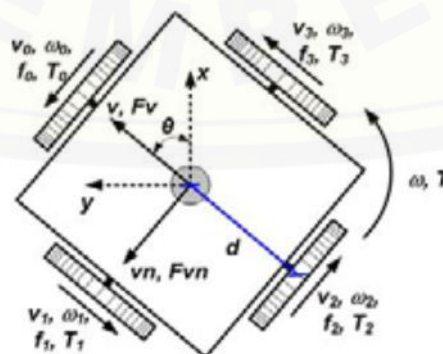
$\omega$  = kecepatan sudut ( $^{\circ}/s$ )

t = waktu tempuh (s)

## 2.2. Omni Directional Drive

Robot *omni directional* telah banyak dipelajari dan dikembangkan secara ekstensif selama sepuluh tahun terakhir di sejumlah laboratorium robotika di seluruh dunia. Robot tersebut ditandai dengan kemampuan untuk bergerak ke samping dan berputar di tempat (Efendi, 2007). Salah satu sistem pergerakan robot yang sedang berkembang pesat adalah robot dengan *four omni-directional drive*. Robot ini memiliki bentuk persegi dengan roda omni ditempatkan pada tiap usung sisi robot. *Omni wheels robot* adalah robot yang dapat berpindah ketempat lain yang dirancang dengan menggunakan roda omni. Roda omni yakni rancangan roda yang tidak hanya mempunyai roda utama, tetapi banyak roda dalam satu roda inti. Terdapat sebuah roda utamayang berukuran lebih besar dan bagian tepi terdapat banyak roda bebas yang berukuran lebih kecil yang mempunyai sumbu tegak lurus terhadap sumbu roda inti. *Omni wheels robot* termasuk tipe *holonomic*, yaitu dapat bergerak bebas kesegala arah. Untuk mencapai pergerakan yang bebas tersebut maka sebuah penelitian haruslah membutuhkan perhitungan kinematika dari *omni wheels robot*.

Untuk memahami dasar dari *omni-directional* robot yang dikembangkan, kita dapat meneliti sistem kinematika dari robot tersebut. Untuk mendapatkan perpindahan robot dari titik awal robot bergerak, kita harus mengetahui variabel-variabel yang berkaitan sehingga dapat dilakukan kontrol seperti posisi gerak dari robot dan sudut masing-masing roda. Perhitungan kinematik pada robot digunakan untuk menentukan perubahan posisi antar koordinat global dan posisi koordinat internal dari robot itu sendiri.



Gambar 2.2 Kinematika *four wheels omni directional*  
(Helder P. Oliveira, dkk 2009)

Dimana :

- $x, y, \theta$  : posisi robot dalam koordinat  $(x, y)$  dan  $\theta$  merupakan sudut hadap robot.
- $d$  (m) : jarak roda terhadap titik tengah robot;
- $v_{(0,1,2,3)}$  (m/s) : kecepatan linier roda;
- $\omega_{(0,1,2,3)}$  (rad/s) : kecepatan sudut roda;
- $f_{(0,1,2,3)}$  (N) : arah gaya roda;
- $T_{(0,1,2,3)}$  (Nm) : arah torsi roda;
- $v, v_n$  (m/s) : kecepatan linier robot;
- $\omega$  (rad/s) : kecepatan sudut robot;
- $F_v, F_{v_n}$  (N) : arah gaya robot terhadap  $v$  dan  $v_n$ ;
- $T$  (Nm) : torsi robot.

Untuk mendapatkan permodelan dari sebuah robot yang bergerak pada sebuah permukaan bidang datar, haruslah mengidentifikasi posisi sebagai  $(x, y, \theta)$

dan juga kecepatan linier masing masing sumbu yakni  $v_x(t) = \frac{dx(t)}{dt}$ ,  $v_y(t) = \frac{dy(t)}{dt}$ ,  $\omega(t) = \frac{d\theta(t)}{dt}$ ,  $v_x(t) = \frac{dx(t)}{dt}$ ,  $v_y(t) = \frac{dy(t)}{dt}$ ,  $\omega(t) = \frac{d\theta(t)}{dt}$ ,  $v_x(t) = \frac{dx(t)}{dt}$ ,  $v_y(t) = \frac{dy(t)}{dt}$ ,  $\omega(t) = \frac{d\theta(t)}{dt}$ .

Pada gambar 2.2 menunjukkan posisi arah depan dan menunjukkan arah  $v$  dan arah ortogonal  $v_n$  dari sumbu robot. Berikut merupakan persamaan yang digunakan untuk merubah kecepatan linear pada kondisi statis menjadi kecepatan linear terhadap sumbu robot.

$$\begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & \sin(\theta(t)) & 0 \\ -\sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_x(t) \\ v_y(t) \\ \omega(t) \end{bmatrix} \dots\dots\dots (2.3)$$

Sedangkan hubungan antara kecepatan linear sumbu robot terhadap kecepatan roda di rumuskan dalam persamaan berikut:

$$\begin{bmatrix} v_0(t) \\ v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & d \\ -1 & 0 & d \\ 0 & -1 & d \\ 1 & 0 & d \end{bmatrix} \cdot \begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} \dots\dots\dots (2.4)$$



Persamaan diatas memungkinkan untuk mendapatkan persamaan untuk menentukan kecepatan robot dengan kecepatan roda meskipun matriks tidak dalam bentuk persegi. Hal ini dikarenakan sistem dalam keadaan redundant (berlebihan).

Berikut persamaan yang diperoleh

$$v(t) = \left(\frac{1}{2}\right) \cdot (v_3(t) - v_1(t)) \dots\dots\dots (2.5)$$

$$v(t) = \left(\frac{1}{2}\right) \cdot (v_0(t) - v_2(t)) \dots\dots\dots (2.6)$$

$$\omega(t) = \left(\frac{1}{4.d}\right) \cdot (v_0(t) + v_1(t) + v_2(t) + v_3(t)) \dots\dots\dots (2.7)$$

(Helder P. Oliveira, dkk 2009).

### 2.3. KRAI (Kontes Robot ABU Indonesia)

Didunia robotika kita sering mengetahui banyak kontes-kontes robot tingkat pelajar maupun mahasiswa. Salah satunya Kontes Robot Indonesia (KRI), kontes ini merupakan salah satu kompetisi robotika mahasiswa perguruan tinggi tingkat nasional yang diadakan secara rutin setiap tahun oleh DIKTI (Direktorat Jendral Pendidikan Tinggi). Kompetisi ini terbagi menjadi beberapa kategori perlombaan yakni kategori KRAI (Kontes Robot Abu Indonesia), kategori KRPAI (Kontes Robot Pemadam Api Indonesia) dan kategori KRSBI (Kontes Robot Sepak Bola Indonesia). Masing-masing kategori memiliki peraturan, misi dan medan pertandingan yang berbeda. Pada kategori Kontes Robot ABU Indonesia (KRAI) peraturan yang digunakan selalu berubah setiap tahun bergantung pada dimana ABU ROBOCON dilaksanakan. (DIKTI, 2016)

Tema dan peraturan yang diusung oleh Kontes Robot ABU Indonesia 2016 berpedoman pada ABU *Asia-Pacific Robot Contest* (ABU ROBOCON) 2016 yang diselenggarakan di Bangkok. Kontes Robot ABU Indonesia (KRAI) 2016 bertemakan "Efisiensi Energi Terbarukan", dimana tema tersebut berpedoman pada tema ABU Robocon 2016, yaitu "Clean Energy Recharging The World". Pada Kontes Robot ABU Indonesia (KRAI) ini terdapat dua robot, yaitu *eco-robot* atau robot otomatis tanpa aktuator penggerak roda dan *hybrid-robot* yakni robot semi otomatis yang dikendalikan oleh seorang *driver*, kedua robot bekerja sama untuk menempatkan baling-baling ke *box* turbin angin. (DIKTI, 2016)

### 2.3.1 *Eco-Robot*

*Eco-robot* bertugas sebagai robot yang membawa *propeller* yang akan dipasang oleh *hybrid-robot* pada *wind turbine*. *Eco-robot* tidak memiliki aktuator penggerak, *eco-robot* hanya menerima dorongan dari *hybrid-robot*. *Eco robot* hanya memiliki satu aktuator yang digunakan sebagai kemudi atau pengatur direksi robot dan sensor yang digunakan untuk melacak jalan yang harus dilewati. (DIKTI, 2016)

*Eco-robot* haruslah dapat melalui medan berupa “*slope and hills*” yang terdiri dari tiga tingkat tanjakan. kemudian berbelok melalui zona “*river*” dan turun melalui zona “*down hill*” sampai mencapai zona “*wind turbine station*”. (DIKTI, 2016)

### 2.3.2 *Hybrid-Robot*

*Hybrid-robot* bergerak dengan sistem semi otomatis maupun *full* otomatis. *Hybrid-robot* bertugas memberikan dorongan secara tidak langsung kepada *eco-robot*, dorongan dapat berupa tiupan angin, gaya magnetik dan lain-lain. *Hybrid-robot* juga bertugas meletakkan *propeller* yang dibawa oleh *eco-robot* dan memasang *propeller* pada *wind turbine*. (DIKTI, 2016)

## 2.4. Arduino Mega2560

Arduino Mega2560 adalah *board* mikrokontroler yang menggunakan ATmega2560 *processor* utamanya. Arduino Mega2560 memiliki 15 pin dapat digunakan sebagai *output* PWM, 16 pin sebagai *input* analog, dan 4 pin sebagai UART (*port serial hardware*) hingga total jumlah pin sebanyak 54 pin *digital input/output*, 16 MHz kristal osilator, koneksi USB, *jack power*, *header ICSP*, dan tombol *reset*. Komponen diatas merupakan *device* pendukung mikrokontroler. Untuk mengaktifkan arduino Mega2560 cukup dengan menghubungkannya ke komputer melalui kabel USB atau *power* yang dihubungkan dengan adaptor AC-DC atau baterai. Sebagian besar *shield* yang dirancang untuk Arduino Duemilanove atau Arduino Diecimila kompatibel dengan Arduino Mega2560.



Gambar 2.3 Arduino Mega2560. (Wahyu Hidayat, 2016)

#### 2.4.1 Spesifikasi Arduino Mega2560

##### a. Power Supply

Power supply merupakan bagian penting dari Arduino Mega2560, Arduino Mega 2560 tidak akan aktif jika tidak adanya ketersediaan power supply yang memadai. Pada dasarnya dalam board arduino Mega2560 terdapat dua pin power supply yang dapat dipergunakan untuk menghubungkan arduino dengan power supply, yakni pin VIN dan pin 5V. Pin VIN dapat dihubungkan dengan powersupply eksternal seperti baterai maupun adaptor, batas yang diperbolehkan yakni sebesar 7 hingga 12 volt. Pin VIN juga dapat diakses melalui jack socket pada board. Pin 5V yakni pin yang terhubung dengan regulator 5 volt yang sudah terpasang pada board. Pin ini dapat diakses sebagai output apabila pin VIN mendapat supply atau USB port terhubung dengan komputer, untuk dijadikan sebagai input sumber batas yang diperbolehkan yakni 5 volt. Selain itu juga ada pin 3V3 dimana dapat menghasilkan tegangan 3,3 volt yang berasal dari keluaran regulator 3,3 volt yang terdapat pada *board* dengan arus maksimum sebesar 50 mA. Kemudian terdapat pin IOREF, pin ini memiliki fungsi sebagai nilai referensi untuk input output pin pada arduino. Yang terakhir adalah pin GND dimana pin ini terhubung dengan semua ground pada board, pin ini difungsikan sebagai referensi 0 volt pada sistem.

##### b. Memori Arduino Mega2560

Memori merupakan salah satu bagian penting dari arduino, terdapat tiga memori pada arduino Mega 2560 yakni 256 KB *flash memory* untuk menyimpan kode yang telah dibuat dalam Arduino IDE dengan 8 KB memori yang telah digunakan untuk menyimpan *bootloader*. Kemudian Arduino Mega2560 memiliki



8 KB SRAM dan serta 4 KB EEPROM yang apabila Arduino Mega2560 tidak diprogram ulang memori ini tidak akan terhapus berbeda dengan flash memori yang dapat terhapus saat Arduino Mega2560 diprogram ulang, memori ini dapat diakses dengan *library* EEPROM.

c. Input dan Output

Semua pin analog dan digital pada Arduino Mega2560 dapat difungsikan sebagai pin IO dan beroperasi pada tegangan 5 volt dengan arus maksimum sebesar 40 mA serta memiliki resistor pullup virtual sebesar 20 hingga 50 kOhm. Namun ada juga pin yang memiliki fungsi khusus sebagai serial pin yakni pin 0, 19, 17 dan 15 sebagai RX serta pin 1, 18, 16 dan 14 sebagai TX. Kemudian pin yang berfungsi sebagai eksternal interrupt yakni pin 2, 3, 18, 19, 20 dan 21, pin ini dapat diatur untuk memicu sebuah interupsi pada nilai yang rendah, dari logika LOW menuju HIGH atau *rising* dan dari logika HIGH menuju LOW atau *falling*, serta perubahan nilai atau *change*. Setelah itu terdapat analog pin yang digunakan sebagai voltage translator untuk tegangan dari 0 volt hingga 5 volt dengan rentang data sebesar 10 bit.

d. Komunikasi Arduino Mega2560

Di dalam Arduino Mega2560 terdapat perangkat yang digunakan untuk berkomunikasi dengan perangkat lain. Pada Arduino mega2560 terdapat empat *hardware* komunikasi *serial* UART TTL. Kemudian terdapat komunikasi *serial peripheral interface* (SPI) yang terhubung dengan pin 50, 51, 52 dan 53 yang dapat diakses menggunakan *library* SPI. Arduino Mega2560 juga didukung oleh komunikasi *two wire interface* (TWI) yang terhubung pada pin 20 dan pin 21 dan diakses menggunakan *library* Wire.

e. Pemrograman

Terdapat banyak pilihan bahasa pemrograman arduino yang membedakan yakni cara uploading dan komunikasi yang digunakan. Pada umumnya Arduino diprogram dengan menggunakan software Arduino IDE yang open source. Pada Arduino Mega2560 sudah tersedia *preburned* dengan *bootloader* yang memungkinkan pengguna untuk memasukkan program baru tanpa menggunakan *programmer hardware* eksternal dengan menggunakan komunikasi *serial*. Hal ini

karena komunikasi yang terjadi menggunakan protocol STK500. Pengguna juga dapat melakukan *bypass bootloader* memasukkan program melalui pin *header ICSP (In-Circuit Serial Programming)*.

## 2.5. Sensor

Sensor adalah *hard device* yang digunakan untuk merubah suatu nilai tertentu menjadi satuan sinyal sehingga dapat diterjemahkan oleh rangkaian elektronik tertentu. Sensor sendiri merupakan komponen utama dari suatu transduser, dimana kita ketahui transduser merupakan sistem yang melengkapi agar sensor tersebut mempunyai keluaran sesuai yang kita inginkan dan dapat langsung diketahui nilainya.

Sensor adalah jenis transduser yang digunakan untuk mengubah besaran mekanik, kimia, suhu, cahaya, dan magnetis menjadi tegangan dan arus listrik. Sensor sering dipergunakan untuk pendeteksian saat melakukan pengukuran atau kontrol suatu sistem. Dalam bidang sistem kendali dan robotika, sensor memberikan kesamaan yang menyerupai mata, pendengaran, hidung, lidah yang kemudian akan diolah oleh *processor* sebagai otaknya (Petruzella, 2001).

Sensor dalam teknik pengukuran dan pengaturan secara elektronik berfungsi mengubah tegangan fisika menjadi besaran listrik yang linier. Sensor dalam teknik pengukuran dan pengaturan ini harus memenuhi persyaratan-persyaratan berikut yakni :

a. tingkat linieritas sensor

Konversi harus benar-benar linier, sehingga karakteristik konversi harus menjadi nilai yang linier.

b. Tidak dipengaruhi oleh temperatur

*Output* penguat tidak boleh bergantung pada besarnya temperatur disekelilingnya, kecuali sensor suhu.

c. Tingkat sensitifitas sensor

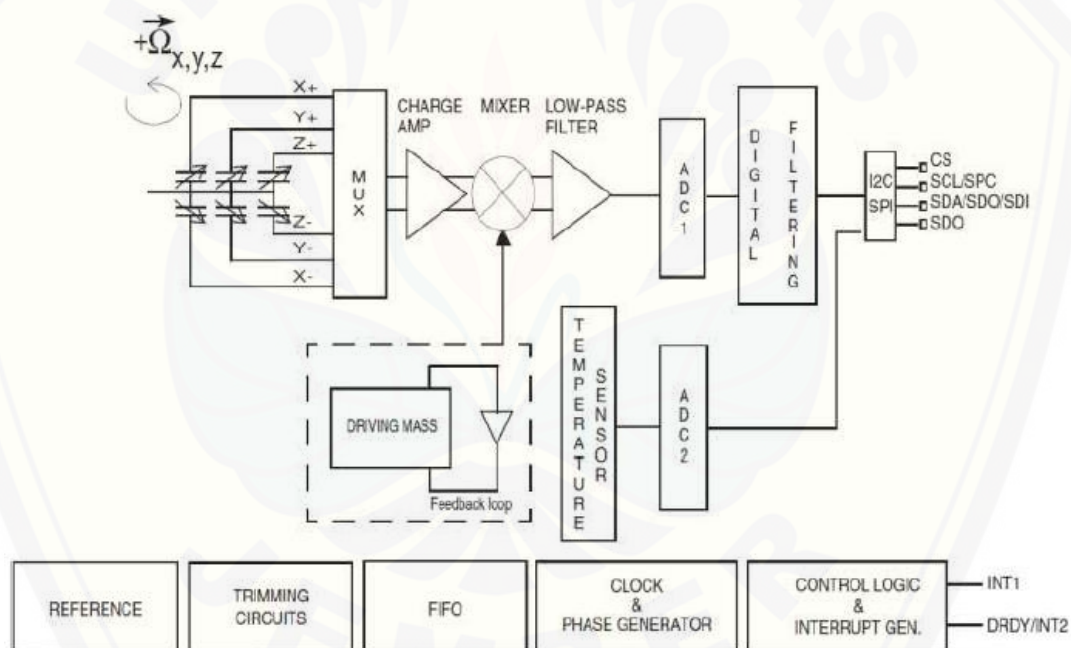
Tingkat sensitifitas sensor harus dipilih sedemikian rupa sehingga pada nilai-nilai *input* yang diperoleh, dapat diolah menjadi tegangan listrik dengan keluaran yang cukup besar.

#### d. Waktu tanggapan (*respon time*)

Waktu tanggapan (*respon time*) adalah waktu yang diperlukan *output* sensor untuk mencapai nilai akhirnya ketika nilai *input* berubah secara mendadak. Nilai keluaran sensor harus dapat berubah cepat bila nilai *input* pada sistem tersebut berubah. (Arifur Anwar, 2015)

#### 2.5.1 Sensor *gyroscope* L3G4200D

Sensor *gyroscope* L3G4200D dapat digunakan untuk mengukur tiga koordinat kartesian yaitu sumbu x, y dan z. Modul sensor ini mampu bekerja hingga percepatan 10000 kali percepatan gravitasi bumi, dan mampu membaca kecepatan sudut dengan tiga pilihan skala penuh yakni 250 dps, 500 dps dan 2000 dps.

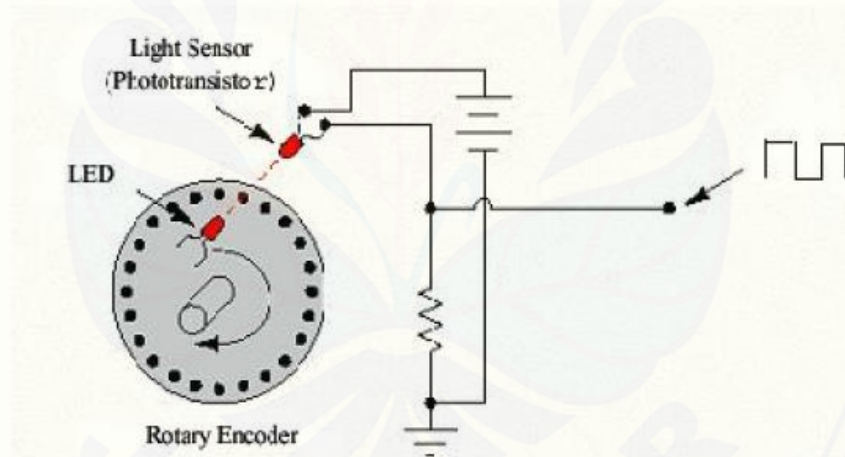


Gambar 2.4 Diagram Blok L3G4200D (Mochammad Rif'an dkk, 2012)

*Output* sensor adalah data kecepatan sudut dalam bentuk digital 16 bit dan data temperatur 8 bit. Tegangan catu daya yang dibutuhkan 2,4 volt sampai 3,6 volt.

#### 2.5.2 Rotary encoder

*Rotary encoder* adalah komponen elektronika yang bekerja secara mekanik dan menggunakan sensor optik maupun kontak yang menghasilkan pulsa-pulsa yang dapat diubah kedalam suatu nilai posisi atau arah. Pada Gambar 2.10 menunjukkan prinsip kerja secara umum dari *rotary encoder*, dimana sebuah *disk* tipis dan LED yang ditempatkan sedemikian rupa sehingga cahayanya tetap terfokus pada *disk* tersebut. Sebuah *phototransistor* ditempatkan pada sisi lain dari *disk* sehingga dapat mendeteksi cahaya dari LED. *disk* tersebut diposisikan pada *shaft* yang bergerak. Dimana pergerakan *disk* tersebut sesuai dengan pergerakan *shaft*, sehingga ketika *shaft* berputar, maka *disk* ikut berputar. Ketika *disk* yang di atasnya ditempatkan LED yang memancarkan cahaya yang terfokus terhadap *phototransistor*, *phototransistor* akan berada dalam keadaan saturasi yang keluarannya berupa pulsa gelombang kotak. (Rizal Banodin, 2006)



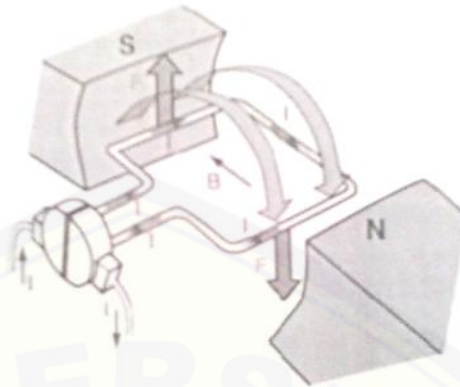
Gambar 2.5 Prinsip kerja secara umum dari *rotary encoder* (Rizal Banodin, 2006)

## 2.6. Motor DC

Motor adalah mesin listrik yang merubah energi listrik menjadi energi mekanis. Pada motor arus searah energi listrik yang dirubah adalah energi DC yang berasal dari sumber tegangan arus searah. Dimana sumber tegangan ini dihubungkan dengan rangkaian dan rangkaian jangkar dari motor. (Ardian Widyatama, 2013)



Prinsip kerja dari sebuah motor DC dapat dijelaskan dengan gambar berikut,



Gambar 2.6 Prinsip perputaran motor DC (Agung Teguh Pribadi, 2015)

Pada motor terdapat dua buah medan magnet permanen yang menimbulkan fluksi yang melingkar dari utara menuju selatan. Saat coil pada motor terhubung dengan sumber tegangan, coil menghasilkan fluksi magnet yang memotong fluksi magnet permanen dan mengakibatkan perubahan kerapatan antar fluksi magnet sehingga menciptakan sebuah torsi. Besarnya fluksi magnet pada coil bergantung pada besarnya arus yang melewati coil tersebut. Arah gaya yang ditimbulkan dapat ditentukan dengan menggunakan “Kaidah Tangan Kiri *Fleming*” yang berbunyi: apabila tangan kiri terbuka diletakkan di antar kutub U dan S, sehingga garis-garis gaya yang keluar dari kutub utara menembus telapak tangan kiri dan arus di dalam kawat mengalir searah dengan arah keempat jari, maka kawat itu akan mendapat gaya yang arahnya sesuai dengan arah ibu jari sebagaimana yang ditunjukkan pada gambar 2.7.

Besarnya gaya dapat dirumuskan  $F = B \cdot I \cdot l \cdot \sin$ , dikarenakan arus jangkar (I) tegak lurus dengan arah induksi magnet (B), maka besar gaya yang dihasilkan oleh arus yang melewati konduktor jangkar yang ditempatkan pada suatu medan magnet adalah :

$$F = B \cdot I \cdot L \dots\dots\dots (2.8)$$

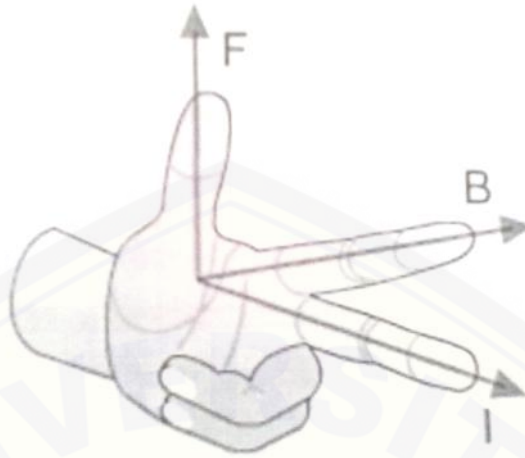
Keterangan :

F = Gaya *lorenz* (Newton)

B = kerapatan fluk (Weber/m<sup>2</sup>)

I = arus yang mengalir pada konduktor (ampere)

$L$  = panjang konduktor (m).



Gambar 2.7 Kaidah tangan kiri *Fleming* (Agung Teguh Pribadi, 2015)

**BAB 3. METODOLOGI PENELITIAN**

**3.1. Tempat dan Waktu Penelitian**

Adapun tempat dan waktu penelitian - analisis dilakukan secara umum dilakukan di :

Tempat: Laboratorium Elektronika Terapan Fakultas Teknik, Universitas Jember

Alamat: Jalan Slamet Riyadi no 62 Patrang, Kecamatan Patrang, Kabupaten Jember

Waktu : Oktober 2016 – Maret 2016

Tabel 3. 1 Jadwal Kegiatan Penelitian

No	Kegiatan	Bulan					
		I	II	III	IV	V	VI
1	Studi literatur	■					
2	Perancangan dan pemodelan sistem		■	■	■	■	
3	Pengambilan data			■			
4	Analisa data				■		
5	Pembuatan laporan				■	■	■

**3.2. Alat dan Bahan**

Dalam penelitian ini digunakan peralatan-peralatan sebagaimana diuraikan dalam tabel 3.2 Penjelasan lebih lengkap untuk beberapa komponen adalah sebagaimana dijelaskan dalam Bab 2.

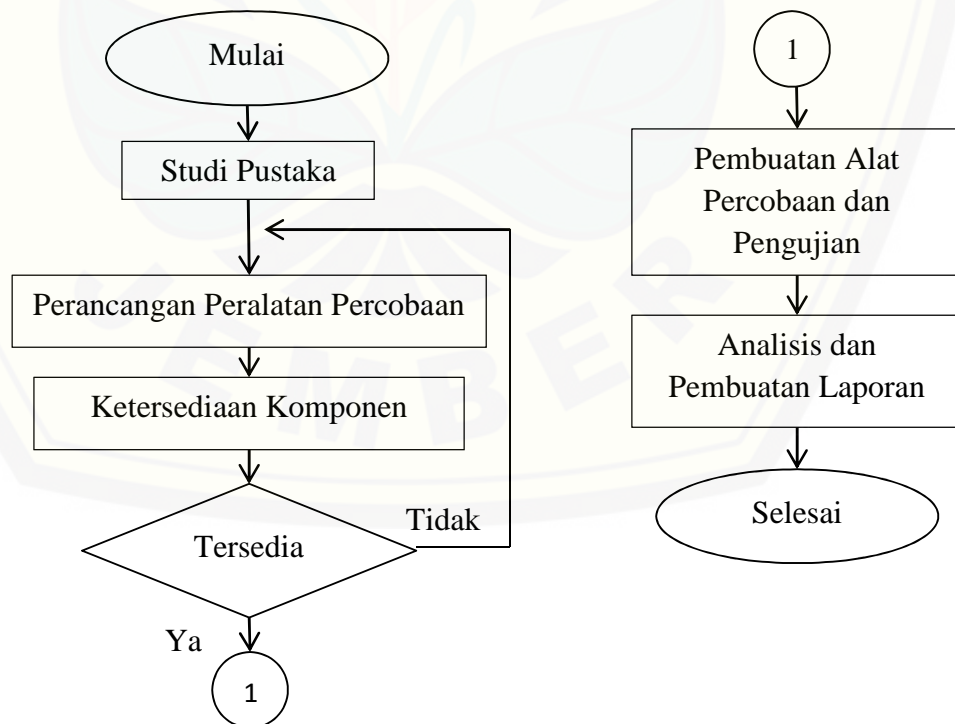


Tabel 3.2 Daftar peralatan yang digunakan

No	Nama Alat	Spesifikasi Teknis dan penjelasan
1.	Robot	<b>Robot manual KRAI 2016</b> , 4 omni wheels 120mm, 600 x 600 x 200mm.
2.	Baterai	<b>ZIPPY Flightmax 5000mAh 3S1P 20C</b> , 5000mAh / 11.1v Discharge: 20C Constant / 30C Burst, Dimensi: 145x52x25mm, Balance Plug: JST-XH
3.	Rotary encoder	<b>KY-040</b> , Incremental Encoder, 20 CPR Material: PCB + Brass, Size: 32 x 19 x 30mm
4.	Sensor <i>Gyroscope</i>	<b>L3G4200D modules (three-axis gyro module)</b> .3-5V. Modes of communication : IIC communication protocol / spi. Measurement range: 250/500/2000 dps.
5.	LCD	<b>LCD 20x4.5V</b> 16 pin. 20 karakter axis x dan 4 karakter axis y
6.	ARDUINO Controller	<b>Arduino MEGA 2560-r3</b> . <b>Atmega2560</b> microcontroller, input voltage - 7-12v, 5v electric current : 500ma, 3.3v electric current : 50ma, 54 digital i/o pins (16 pwm outputs), 16 analog inputs
7.	Aksesoris rangkaian	Kabel, selang udara, resistor, dll.
8.	Peralatan kerja bengkel	Obeng, kunci, bor dll.

### 3.3. Diagram Alir Penelitian

Langkah-langkah dalam penelitian dan penyusunan skripsi ini dapat dinyatakan dalam diagram alir sebagaimana terlihat dalam Gambar 3.1



Gambar 3.1 Diagram Alir Penelitian

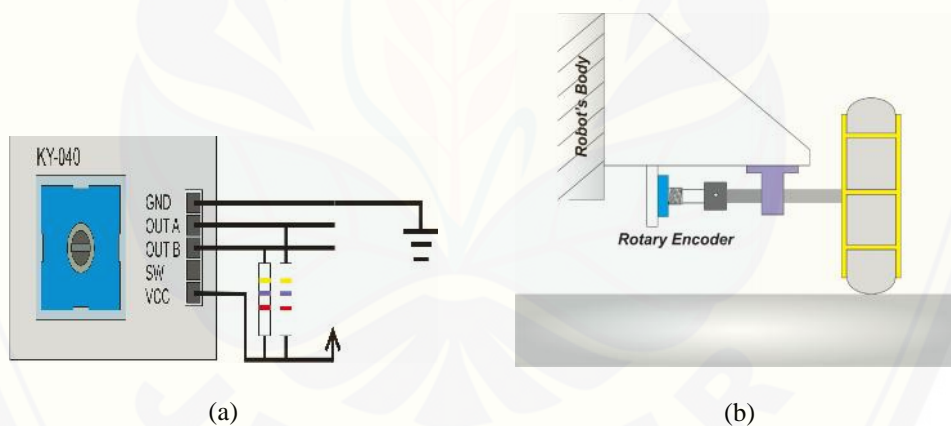
### 3.4. Pembuatan dan Pengujian Sensor

Dalam bagian ini akan dibahas langkah-langkah pengujian sensor-sensor yang diperlukan dan dibuat dalam penelitian ini. Hasil pengujian disajikan dan dibahas dalam BAB 4.

#### 3.4.1 Pembuatan dan Pengujian Sensor Posisi

Sensor posisi diperlukan untuk mendapatkan posisi robot pada koordinat cartesian dan juga untuk memperoleh besar kecepatan pada sumbu linier, maka dari itu sensorposisi termasuk sistem yang dibuat dan diuji terlebih dahulu dalam penelitian ini.

Sensor ini dibuat dengan berbasis pada komponen *rotary encoder* dan roda omni 50mm seperti yang dijelaskan dalam sub-sub 2.5.3 point 2.5.3.1. Komponen tambahan yang diberikan adalah sebuah resistor yang ditempatkan pada kaki digital OUT dan +VCC. Gambar 3.2 menunjukkan penempatan resistansi 4.7 k $\Omega$  dan penempatan roda omni dan *rotary encoder* sensor pada *body* robot.



Gambar 3.2 Sensor posisi dan penempatannya

*Output* dari sensor dihubungkan dengan digital *input* pada Arduino MEGA 2560 -R3 dan hasil dan hasil pengujian di lihat dengan menghubungkan port USB Arduino ke PC.

Pengujian pertama yakni pengujian untuk mengetahui sinyal output dari *rotary encoder*. Pada pengujian ini, *rotary encoder* dihubungkan dengan *oscilloscope* dan diputar searah jarum jam serta berlawanan dengan arah jarum jam. Sinyal ditampilkan pada *oscilloscope* dan dilakukan analisa.

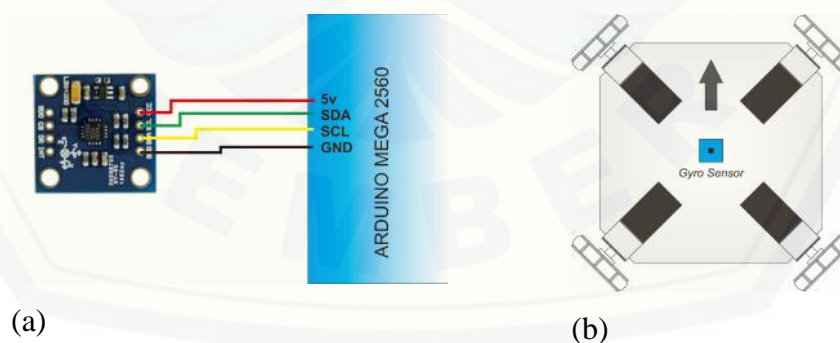
Pada saat pengujian kedua ini dilakukan, *shaft rotary encoder* sudah terkopel dengan roda omni 50mm dan pengambilan data dilakukan dengan cara menjalankan robot pada lintasan lurus dengan cara didorong dengan jarak 100 cm. Hasil pengujian ini dibandingkan dengan hasil pengukuran menggunakan penggaris. Hasil pengujiannya sebagaimana ditabelkan dalam tabel 4.1 dan tabel 4.2.

### 3.4.2 Pembuatan dan Pengujian Sensor Gyroscope

Sensor *gyroscope* dibuat dengan berbasis pada komponen L3G4200D yang berkomunikasi dengan Arduino MEGA 2560 menggunakan I2C. Gambar 3.3 menunjukkan penempatan sensor ini ke robot.

Sensor ditempatkan di titik tengah robot terhadap sumbu x dan y pada koordinat kartesian. Hal ini dimaksudkan agar sensor dapat memperoleh arah hadap robot saat bergerak.

Untuk dapat menguji dan melakukan kalibrasi, pada percobaan ini digunakan media gambar sudut yang sudah disesuaikan dengan penggaris busur untuk membandingkan hasil pengukuran sistem sensor L3G4200D. Hasil pengukuran ini dibandingkan dengan nilai yang tertera pada media kalibrasi yang ditampilkan pada PC melalui Arduino MEGA 2560-R3. Hasil pengujian disajikan dalam sub-sub Tabel 4.3.

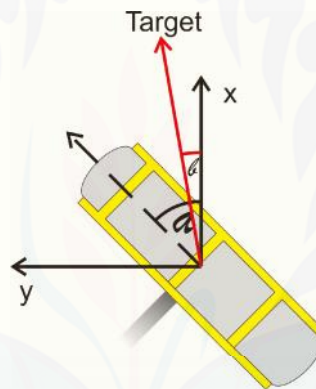


Gambar 3.3 Rangkaian dan penempatan sensor L3G4200D ke robot

### 3.5. Pengujian Robot *Omni directional*

Robot yang digunakan merupakan robot manual yang digunakan oleh tim robotika Universitas Jember dalam Kontes Robot ABU Indonesia 2016. Dimana robot memiliki spesifikasi menggunakan 4 roda omni dengan diameter 120 mm. Bagian robot yang digunakan pada penelitian yakni bagian pergerakan robot bawah, sehingga robot hanya memiliki dimensi 60cm x 60cm x 20cm.

Pengujian robot dilakukan dengan cara menentukan arah gerak robot untuk mengetahui arah pergerakan roda robot. Dengan berpatokan pada sudut masing-masing roda dan arah pergerakan robot serta nilai PWM maksimal maka dapat diperoleh besar nilai PWM masing-masing motor. Berikut rumus yang digunakan dalam mendapatkan besar nilai PWM tiap motor:



Gambar 3.4 Sudut roda terhadap sudut target

$$m1 = PWMmax \times \cos(\text{sudut target} - 45) \dots\dots\dots (3.1)$$

$$m2 = PWMmax \times \cos(\text{sudut target} - 135) \dots\dots\dots (3.2)$$

$$m3 = PWMmax \times \cos(\text{sudut target} - 225) \dots\dots\dots (3.3)$$

$$m4 = PWMmax \times \cos(\text{sudut target} - 315) \dots\dots\dots (3.4)$$

dari sudut target dikurangi dengan sudut roda akan mendapatkan resultan sudut dan jika dikalikan dengan PWM maksimal akan memperoleh besar nilai resultan PWM untuk masing-masing motor. Pada pengujian ini, robot dalam keadaan diam diatas kursi putar kemudian robot dihadapkan kearah tertentu dengan sudut yang telah ditentukan.

Pengujian selajutnya yakni pengujian robot pada bidang datar, berdebu dan bidang miring. Pada pengujian ini robot diatur untuk berjalan sejauh 100 cm

kemudian data yang tercatat oleh robot dibandingkan dengan jarak sesungguhnya yang diukur menggunakan penggaris.

Media untuk pengujian menggunakan laptop untuk melakukan *uploading* program dan keadaan yang diperoleh akan dicatat. Hasil pengujian disajikan dan dibahas dalam Bab 4.

### **3.6. Pengujian Kontrol Gyrodometry**

Dalam bagian ini akan dibahas tentang langkah-langkah pengujian robot dengan menggunakan kontrol *gyrodometry* yang diperlukan dan dibuat dalam penelitian ini.

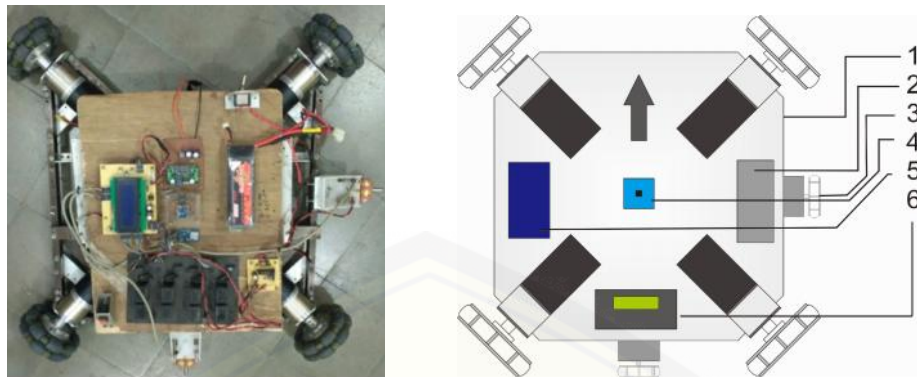
Setelah melakukan pengujian dan kalibrasi sensor-sensor serta melakukan pengujian sistem *omni directional* maka penerapan kontrol *gyrodometry* dapat dilakukan pada robot. Pengujian dilakukan dengan memberikan nilai koordinat target dan besar nilai PWM yang di inputkan secara manual menggunakan sistem *general user interface* (GUI). Data yang diperoleh akan di-*record* dalam data *logger*, *data logger* dalam penelitian ini hanyalah sebuah media pengambilan data sehingga tidak dibahas secara mendalam dalam laporan ini. Data yang diperoleh akan dibandingkan dengan data robot ketika tanpa menggunakan kontrol dan dengan menggunakan kontrol *odometry* serta pengujian robot pada lintasan yang telah ditentukan.

Variabel yang dirubah dalam percobaan ini yakni posisi target dan besar nilai PWM yang diberikan yang berpengaruh pada kecepatan robot. Hasil pengujian disajikan dan dibahas dalam Bab 4.

### **3.7. Perancangan Alat**

Dari penelitian yang saya lakukan dapat dilihat perancangan alat yang akan saya desain seperti gambar 3.5 di bawah ini.

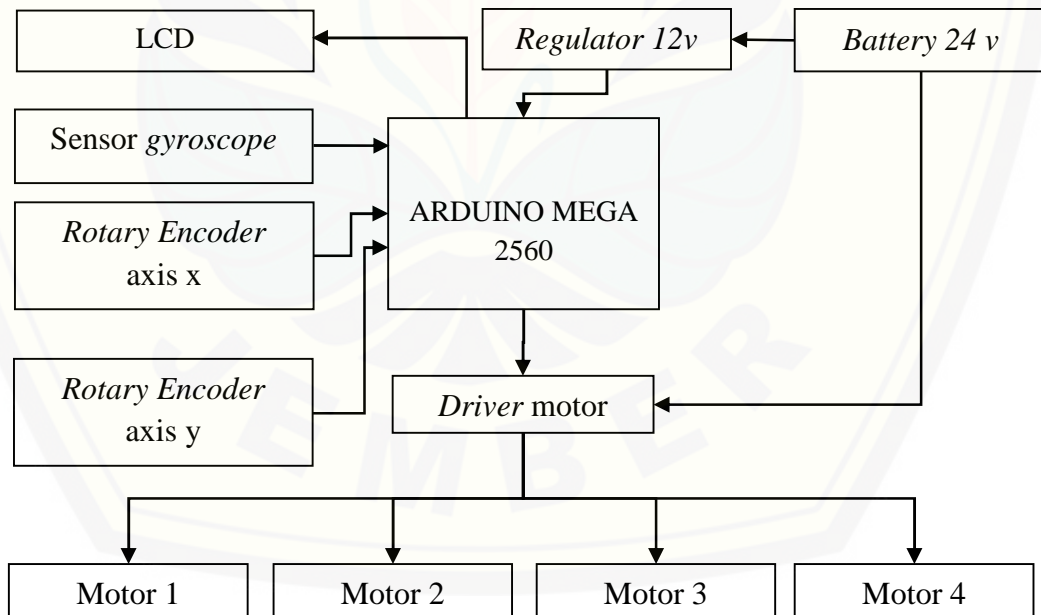




Gambar 3.5 Perancangan alat

Keterangan gambar:

1. Robot *Omni directional drive* KRAI 2016
2. *Driver motor*
3. *Rotary encoder*
4. *Sensor gyroscope*
5. Baterai 12 volt
6. Rangkaian kontrol robot



Gambar 3.6 Blok Diagram Alat

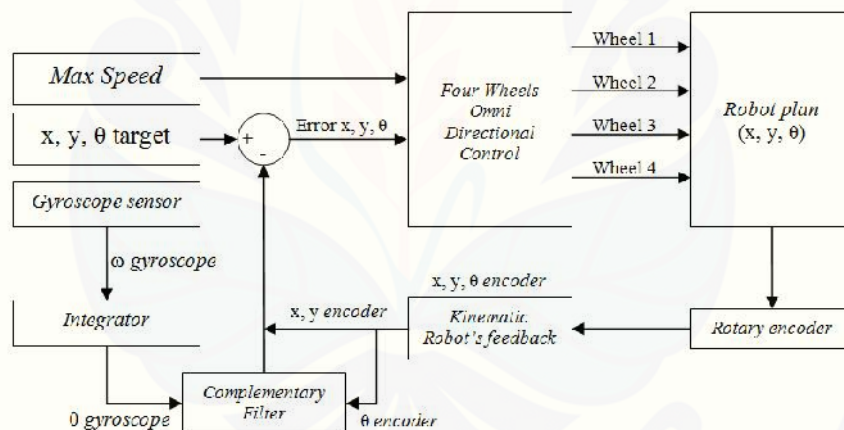
Pada proses pembuatan proyek akhir ini penyusun menggunakan rangkaian sensor *gyroscope* sebagai masukan *angular velocity* yang akan dirubah menjadi besaran sudut ke Arduino yang di-*supply* oleh daya 12 volt yang berasal dari sumber



berupa baterai 24 volt yang telah direduksi oleh regulator 12 volt, kemudian keluaran dari Arduino digunakan untuk mengaktifkan rangkaian *driver* motor untuk menggerakkan 4 buah motor penggerak robot *four wheels omni directional* dimana semua data akan ditampilkan di LCD. Arduino juga mendapatkan masukan berupa data *odometry* yang berasal dari dua buah *rotary encoder* untuk axis x dan axis y.

Prinsip kerja sistem yakni ketika robot bergerak menggunakan sistem *four wheels omni directional*, Arduino akan memperoleh data masukan dari rotary encoder sebagai data *odometry* dan juga sensor *gyroscope* berupa sudut yang digunakan sebagai *feedback* untuk mengoreksi arah hadap robot, penggabungan dua data dari *ordometry* dan *gyroscope* inilah yang dinamakan metode *gyrodometry*.

Berikut adalah blok diagram kontrol *gyrodometry four wheels omni directional* ang akan diterapkan pada robot *hybrid* Kontes Robot ABU Indonesia,



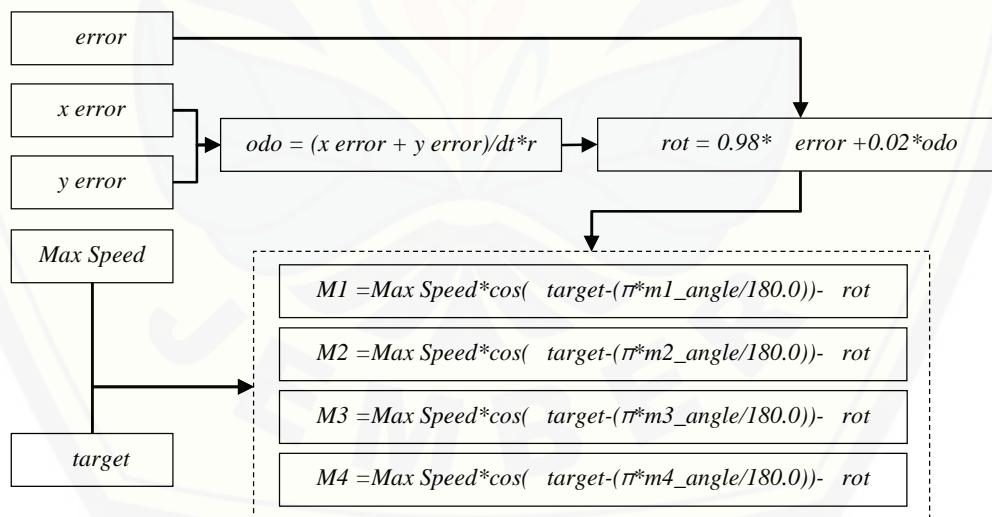
Gambar 3.7 Blok diagram sistem kontrol *gyrodometry four wheels omni directional*

(Irfan Afandi dkk, (Tanpa tahun))

dari blok diagram diatas dapat dijelaskan bahwa *input* dari kontrol robot *four wheels omni directional* berupa kecepatan dan *error* koordinat x, y, dan  $\theta$ . *Error* koordinat x, y dan  $\theta$  diperoleh dari perbandingan nilai pembacaan sensor dengan nilai target. Nilai-nilai tersebut kemudian menjadi *input four wheels omni directional control*, dalam blok *four wheels omni directional control* terdiri dari konversi *inverse kinematik* yang merubah dari besarnya sudut yang diperoleh dari koordinat dan

*input* kecepatan maksimum menjadi nilai kecepatan untuk tiap motor dalam bentuk nilai PWM. *Output* dari kontrol berupa nilai PWM atau kecepatan sehingga robot dapat bergerak sesuai *plan* yang telah dibuat. Saat robot bergerak, diperoleh nilai pembacaan sensor *rotary encoder* yakni nilai  $x$ ,  $y$  dan  $\theta$  serta pembacaan sensor *gyroscope* yakni berupa nilai kecepatan sudut ( $\dot{\theta}$ ), kemudian dilakukan integrasi terhadap nilai kecepatan sudut hingga diperoleh nilai  $\theta$  dari sensor *gyroscope*. Nilai  $x$  dan  $y$  dari *rotary encoder* digunakan sebagai *feedback* dan dibandingkan dengan nilai target, untuk nilai  $\theta$  dari *rotary encoder* dan *gyroscope* dilakukan pemfilteran sehingga didapatkan nilai  $\theta$  *real* dari sistem *gyrodometry* dan digunakan sebagai *feedback* serta dibandingkan dengan nilai  $\theta$  target.

Pada gambar 3.7 terdapat blok *four wheels omni directional control* dimana blok ini berisikan proses merubah nilai error posisi terhadap target dengan mengubah arah gerak dan kecepatan. Hal ini akan dijelaskan melalui blok diagram dibawah ini;

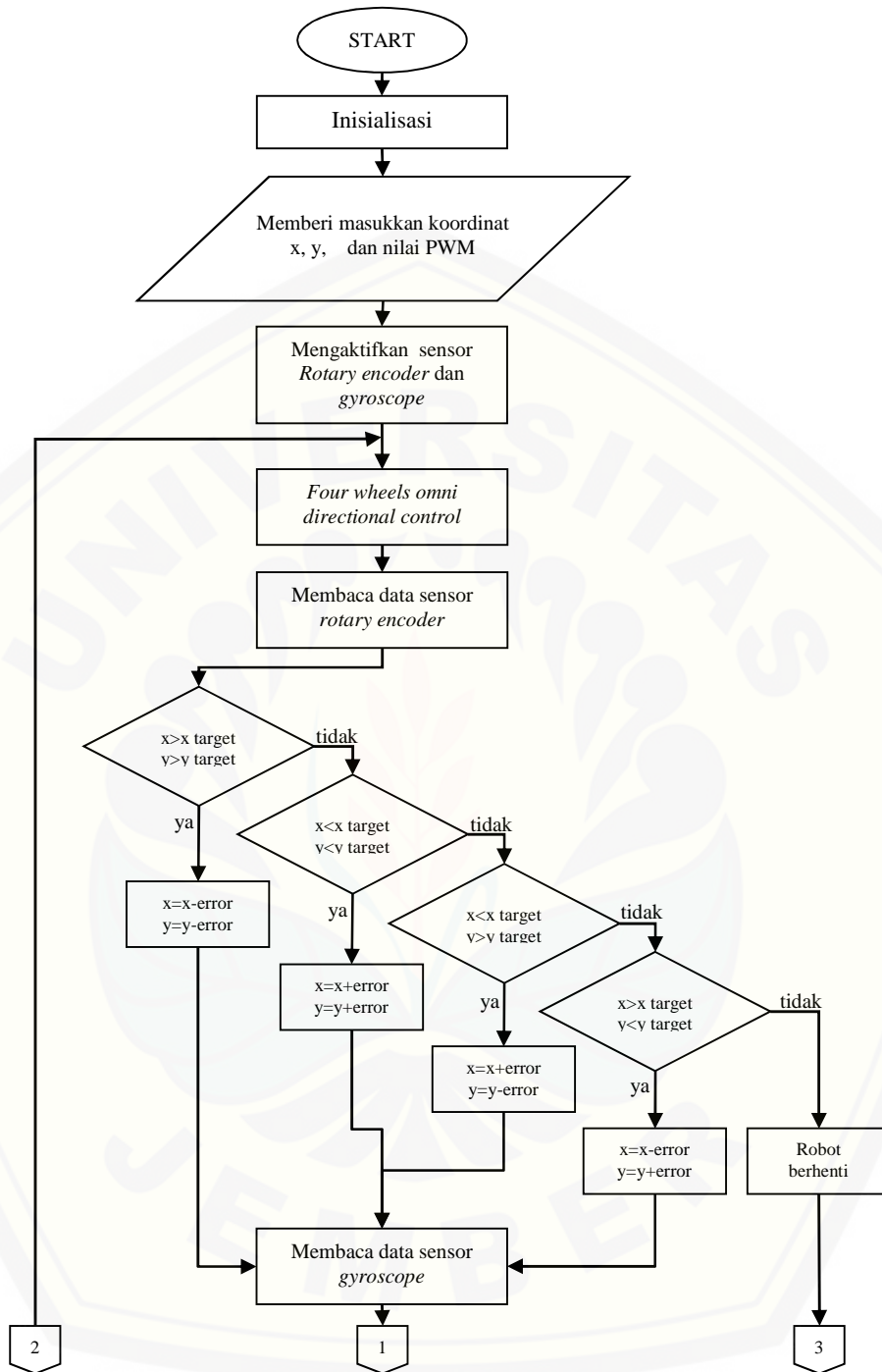


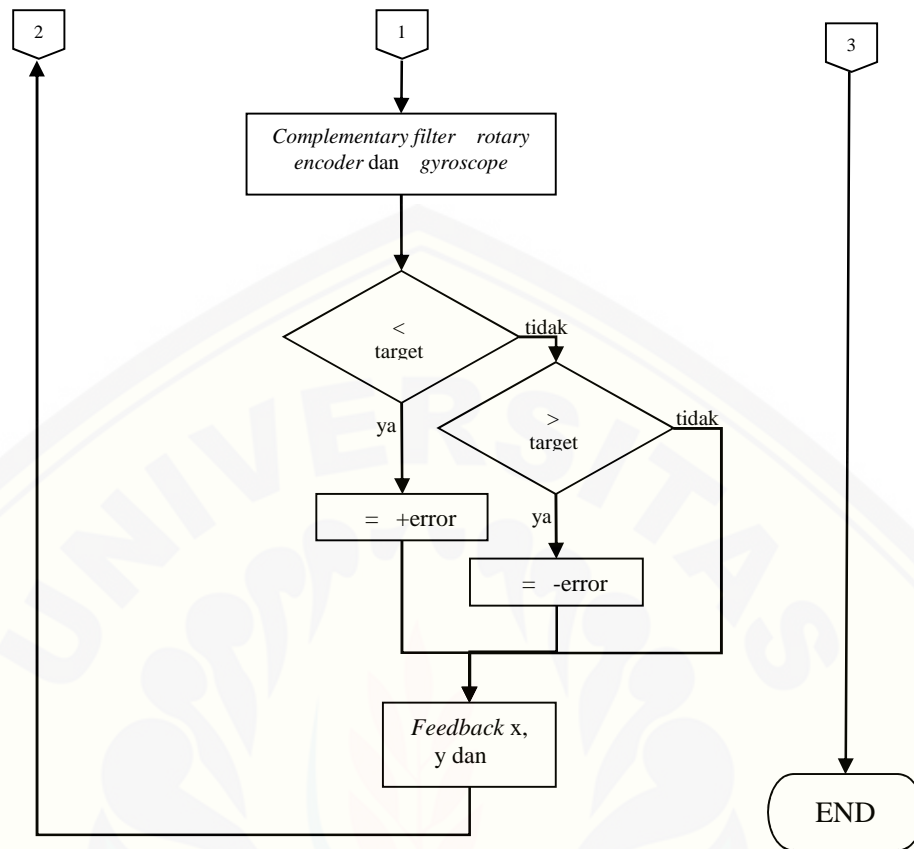
Gambar 3. 8 Blok diagram sistem kontrol *four wheels omni directional*

Dari gambar 3.8 diatas dijelaskan bahwa nilai  $x\ error$  dan  $y\ error$  dirubah menjadi kecepatan sudut terlebih dahulu kemudian dilakukan *fusion* menggunakan *complementary filter* barulah diperoleh nilai sudut rotasi. Untuk mengontrol

kecepatan masing masing motor yakni menggunakan nilai maksimum *speed* dikalikan radian sudut masing-masing roda dikurangi dengan sudut rotasi yang diinginkan.

Pada gambar 3.9 berikut merupakan gambar diagram alir dari robot *four wheels omni directional* dengan menggunakan sistem *gyrodometry*. Pada diagram alir tersebut dijelaskan pertama robot melakukan inisialisasi pin mikrokontroler dan variabel-variabel yang digunakan, setelah itu robot mendapatkan input berupa koordinat  $x$ ,  $y$  dan  $\theta$  serta besar nilai PWM untuk pergerakan robot sebagai variabel nilai target. Kemudian robot mengaktifkan sensor *rotary encoder* sebagai *input feedback* dari *odometry* dan sensor *gyroscope* sebagai *input feedback* untuk mengompensasi nilai *error* yang ditimbulkan oleh sistem *odometry*. Selanjutnya robot bergerak menuju target yang telah dimasukkan dan dilakukan pembacaan sensor *rotary encoder* untuk mengetahui posisi robot. Dari sensor *rotary encoder* diperoleh nilai  $x$ ,  $y$  dan  $\theta$  *real* dari robot, jika  $x$  lebih besar dari  $x$  target maka  $x$  dikurangi besarnya *error*, jika  $x$  lebih kecil dari  $x$  target maka  $x$  dijumlah dengan besarnya nilai *error*. Perlakuan yang sama juga dilakukan kepada nilai  $y$  yang dibandingkan dengan nilai  $y$  target. Namun jika  $x$  dan  $y$  sesuai dengan nilai target robot akan berhenti bergerak, hal ini menunjukkan bahwa robot telah mencapai titik koordinat yang telah ditentukan. Untuk nilai  $\theta$  *real* robot akan dibandingkan dengan pembacaan sensor *gyroscope* dengan menggunakan *complementary filter* dan diperoleh nilai  $\theta$ , nilai  $\theta$  ini diberikan perlakuan yang sama dengan nilai  $x$  dan  $y$  sebelumnya dengan cara membandingkan dengan nilai  $\theta$  target. Setelah itu diperoleh nilai  $x$ ,  $y$  dan  $\theta$  yang digunakan sebagai *feedback* untuk sistem *gyrodometry* dan proses akan terus berulang hingga robot mencapai target yang diinginkan.





Gambar 3.9 Diagram alir program *gyrodometry four wheels omni directional robot*.



## BAB 5. KESIMPULAN DAN SARAN

### 5.1. Kesimpulan

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

1. Sistem kontrol *gyrodometry* yang digunakan sebagai kontrol navigasi dan *trajectory* memiliki tingkat kesesuaian yang cukup bagus yakni dengan simpangan terbesar 0 cm pada saat nilai PWM maksimal sebesar 24 %, hal ini ditunjukkan pada tabel 4.8 halaman 43.
2. Robot *four wheels omni-directional drive* pada penelitian ini menggunakan kontrol *gyrodometry* yang dapat dikatakan cukup baik. Berdasarkan gambar 4.12 robot berjalan sesuai *plan* namun dari lima percobaan yang ditampilkan pada gambar grafik, terjadi simpangan dimana simpangan rata-rata simpangan terbesar *axis x* yakni sebesar 3 cm dengan besar nilai deviasi sebesar 1.399 cm dan pada *axis y* nilai simpangan terbesar yakni sebesar 3.55 cm dengan nilai deviasi sebesar 3.67 cm.

### 5.2. Saran

Dengan harapan penelitian ini dapat membuahkan hasil yang maksimal dan menjadi lebih baik lagi di kemudian hari, untuk penulis itu memberikan saran sebagai berikut:

1. Dalam penelitian ini disarankan untuk menggunakan *rotary encoder* yang memiliki tingkat kepresisian tinggi atau juga dapat menggunakan generator yang nantinya dapat dikonversikan sebagai sensor kecepatan dan diintegrasikan terhadap waktu sehingga menjadi jarak. Hal ini *dikarenakan rotary encoder* pada sistem *gyrodometry* berperan sangat vital, jika pembacaan *rotary encoder* kurang baik maka kontrol *gyrodometry* tidak dapat berjalan secara optimal.
2. Desain penempatan *rotary encoder* juga harus diperhatikan sehingga rotari dapat menapak pada medan dengan sempurna.



3. Dalam penelitian selanjutnya diharapkan dapat menggunakan *driver motor* mosfet dikarenakan respon kontak *relay* yang kurang cepat mengakibatkan robot kurang responsif dalam berjalan berubah arah.
4. Pengujian sistem ini masih belum bisa maksimal dikarenakan tempat pengujian yang terlalu sempit. Untuk penelitian selanjutnya disarankan menggunakan tempat pengujian yang lebih luas seperti di aula atau lain sebagainya.



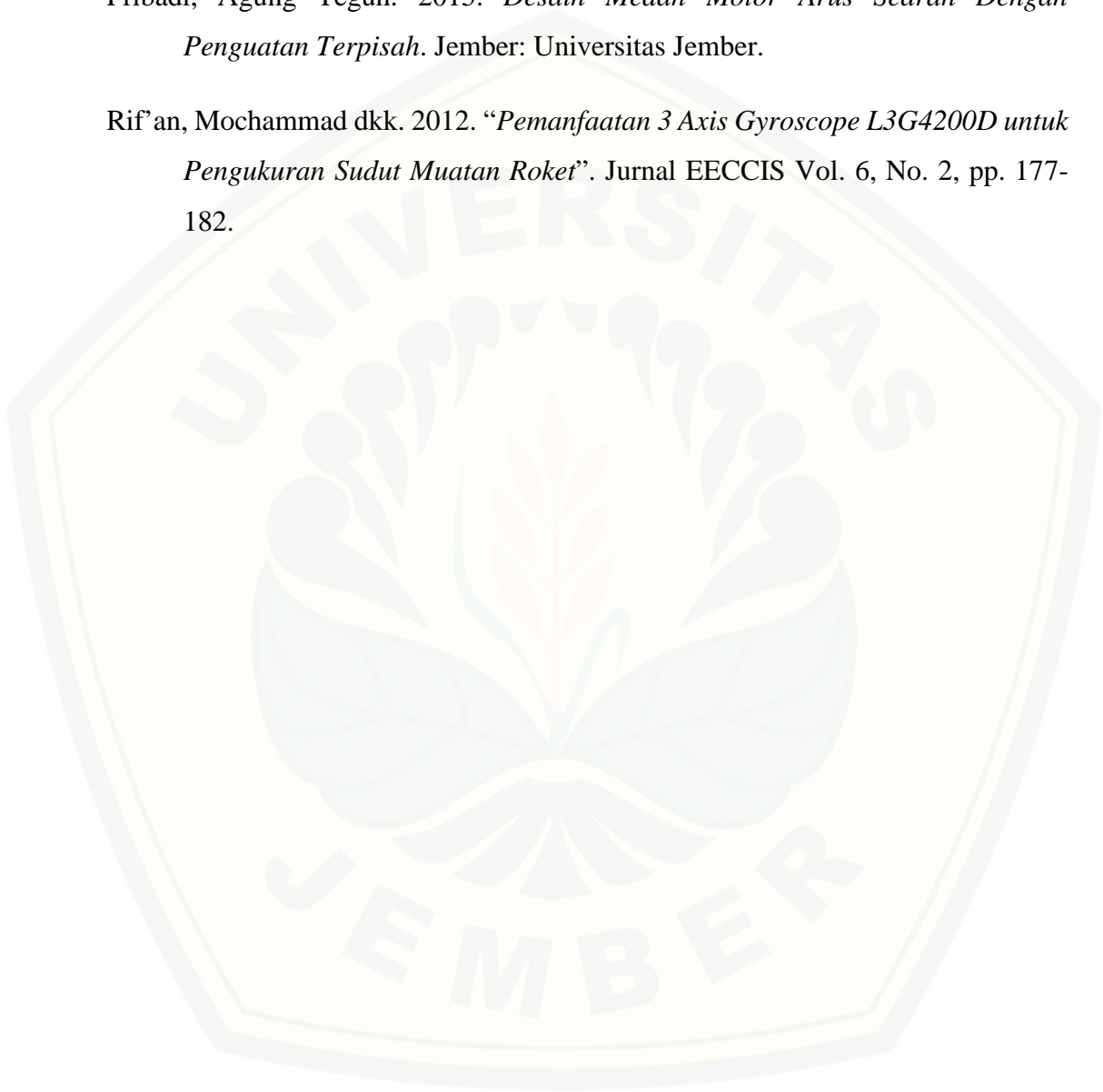
**DAFTAR PUSTAKA**

- Affandi, Irfan, Indra Adji Sulistijono dan Fernando Ardilla. (Tanpa tahun). *Kontrol Posisi Robot Omni-Directional Menggunakan Metode Gyrodometry*. Surabaya: Politeknik Elektronika Negeri Surabaya.
- Banodin, Rizal. (Tanpa tahun). *Alat Penunjuk Arah Angin Dan Pengukur Kecepatan Angin Berbasis Mikrokontroler AT89C51*. Semarang: Universitas Diponegoro.
- Basori, Slamet, Nanang Sulistiyanto dan Mochammad Rif'an. 2014. *Implementasi odometry pada robot otomatis kntes robot ABU Indonesia*. Malang: Universitas Brawijaya.
- Borenstein, J. dan L. Feng. 1996. *Gyrodometry: "A New Method for Combining Data from Gyros and Odometry in Mobile Robots"*. Proceedings of the 1996 IEEE International Conference on Robotics and Automation. Minneapolis, Apr. 22-28, pp. 423-428.
- Borenstein, J.; dkk. (Tanpa tahun). *"Mobile Robot And Positioning – Sensor And Techniques"*. Jurnal Of Robotic System, Special Issue On Mobile Robots. Vol. 14, No. 4, pp. 231-249.
- Dikti. 2016. *Panduan Kontes Robot ABU Indonesia 2016*. Jakarta: DIKTI.
- Efendi, Jefri Bin Moh Salih, 2007, *Design And Development Of Autonomous Omni-Directional Mobile Robot With Mecanum Wheel*, thesis Universiti Malaysia Perlis.
- Hidayat, Wahyu. 2016. *Rancang Bangun Robot Avoider Segala Medan Berbasis ARDUINO MEGA 2560*. Jember: Universitas Jember.

Oliveira, Helder P.; dkk. 2009. Modelling And Assessing of Omni-directional Robot with Three and Four Wheels. CONTEMPORARY ROBOTICS – Challenges and Sollutions, No. 12, pp. 207-229.

Pribadi, Agung Teguh. 2015. *Desain Medan Motor Arus Searah Dengan Penguatan Terpisah*. Jember: Universitas Jember.

Rif'an, Mochammad dkk. 2012. “*Pemanfaatan 3 Axis Gyroscope L3G4200D untuk Pengukuran Sudut Muatan Roket*”. Jurnal EECCIS Vol. 6, No. 2, pp. 177-182.





# LAMPIRAN

**Listing Program Arduino Mega2560**

```
#include <Wire.h>//library rokhim

#include <L3G4200D.h>//library rokhim

#include <SPI.h>//library rokhim

#include <LiquidCrystal.h>//library rokhim

#include <SD.h>//library rokhim

L3G4200D gyroscope;
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

//          {maju, mundur, speed}
const int MT2[]={30, 28, 7};
const int MT3[]={34, 32, 6};
const int MT4[]={38, 36, 5};
const int MT1[]={42, 40, 4};
const int T[]  ={31, 33, 35};

const int chipSelect = 53;

int xSet[]={0,0,0,0};
int ySet[]={0,0,0,0};
int sSet[]={0,0,0,0};
// rotary
int xflag=0x20, yflag=0x08;
int encoder1a = 18;//3;
int encoder1b = 19;//2;
int encoder2a = 3; //18;
int encoder2b = 2; //19;
volatile int lastEncoded1 = 0;//0
volatile int lastEncoded2 = 0;//1
volatile long encoderValue1 = 0;//0/
volatile long encoderValue2 = 0;//2
// Timers
unsigned long timer = 0;
float timeStep = 0.01;
```



```
//kalman filter
float G=1.0;
float varG=pow(0.01,2);
float varM=pow(0.5,3); //0.425565996; //
float K=1.0;
float Kalman=0.0;

// Pitch, Roll and Yaw values
float pitch = 0; //pitch value
float yaw = 0; //yaw value
float roll = 0; //roll value

float sudut; //init sudut var
float fc=1.00 ; //faktor pengali

float r_head,r_rot,m1,m2,m3,m4;
int m2_angle=135, m1_angle=45,
    m3_angle=225, m4_angle=315;
float xValue=0.00;
float yValue=0.00;

double pulsax=0;
double pulsay=0;
int vx=0;
int vy=0;

float odo=0.0;

ISR(TIMER5_OVF_vect){
    TCNT5H=0xC2;
    TCNT5L=0xF7;

    vx=(float)pulsax*60/20;
    vy=(float)pulsay*60/20;
    odo = ((vx+vy)/37)*1000;
    //saveData();
    pulsax=0;
    pulsay=0;
}

void initTimer5(){
```

```
TCCR5A=0x00; //menonaktifkan TCCR5 bagian A
TCCR5B=0x05; //meaktifkan TCCR5 bagian B dengan nilai
05
TCNT5H=0xC2; //meaktifkan TCNTH bagian dengan nilai
C2
TCNT5L=0xF7; //meaktifkan TCNTL bagian dengan nilai
f7

ICR5H=0x00; //menonaktifkan TCCR5 bagian A
ICR5L=0x00; //menonaktifkan TCCR5 bagian A
OCR5AH=0x00; //menonaktifkan TCCR5 bagian A
OCR5AL=0x00; //menonaktifkan TCCR5 bagian A
OCR5BH=0x00; //menonaktifkan TCCR5 bagian A
OCR5BL=0x00; //menonaktifkan TCCR5 bagian A
TIMSK5=0x01; //menonaktifkan TCCR5 bagian A

sei();
}

ISR(INT2_vect) {
    pulsax++;
    yflag=0x08;
    updateEncoder();
}
ISR(INT3_vect) {
    yflag=0x04;
    updateEncoder();
}
ISR(INT4_vect) {
    pulsay++;
    xflag=0x20;
    updateEncoder();
}
ISR(INT5_vect) {
    xflag=0x10;
    updateEncoder();
}

float erSud=0; //7.558;
int enable=0;
```

```
void setup() {
  for(int x=0; x<3; x++){
    pinMode(MT1[x],OUTPUT);
    pinMode(MT2[x],OUTPUT);
    pinMode(MT3[x],OUTPUT);
    pinMode(MT4[x],OUTPUT);
    pinMode(T[x],INPUT_PULLUP);
  }
  Serial.begin(9600);
  lcd.begin(20, 4);
  /*if (!SD.begin(chipSelect)) { //xx
  //xxxx sdcard tidak masuk
    lcd.setCursor(0,0);
    lcd.print("check SD wiring");
    lcd.clear();
    return;
  }
  */
  while(!gyroscope.begin(L3G4200D_SCALE_2000DPS,
  L3G4200D_DATARATE_400HZ_50))// isialisasi sensor giro
  {
    Serial.println("Couldn't find a L3G4200D sensor,
  please check wiring!");
    delay(500);
  }

  pinMode(encoder1a,INPUT_PULLUP);
  pinMode(encoder1b,INPUT_PULLUP);
  pinMode(encoder2a,INPUT_PULLUP);
  pinMode(encoder2b,INPUT_PULLUP);
  EICRA=0x50;//0b01010000
  EICRB=0x05;//0b00000101
  EIMSK=0x14;//0b00010100
  EIFR=0x14;//0b00010100
  initTimer5();
  gyroscope.calibrate(1000);
  gyroscope.calibrate(1000);
}

void loop(){
```

```
    jalan(0,0,0);
    if(!digitalRead(T[0])){
//test(100.0, 0.001, 100);
    menu();
    lcd.clear();
    }
    if(!digitalRead(T[1])){
        test(xSet[0], ySet[0], sSet[0]);
        jalan(0,0,0);
        test(xSet[1], ySet[1], sSet[1]);
        jalan(0,0,0);
        test(xSet[2], ySet[2], sSet[2]);
        jalan(0,0,0);
        test(xSet[3], ySet[3], sSet[3]);
        jalan(0,0,0);
    }
}
void menu(){
    delay(1000);
    lcd.clear();
    int sel=0;
    while(1){
        lcd.setCursor(0,0);
        lcd.print("Set Plan "); lcd.print(sel);
        if(!digitalRead(T[1])){
            delay(1000);
            int num;
            while(num==0){
                num =
                (!digitalRead(T[0])*1)+(!digitalRead(T[1])*2)+(!digital
                Read(T[2])*4);
            }
            switch(num){
                case 1 : settingX(sel); break;
                case 2 : settingY(sel); break;
                case 3 : settingS(sel); break;
            }
        }
        if(!digitalRead(T[0])){ sel++;
        delay(300);if(sel>3){sel=0;}}
        if(!digitalRead(T[2])){ break;}
```

```
    }  
  }  
void settingX(int s){  
  delay(1000);  
  int inc;  
  while(1){  
    if(!digitalRead(T[0])){inc++;}  
    if(!digitalRead(T[2])){inc--;}  
    if(!digitalRead(T[1])){break;}  
    xSet[s]= inc;  
    lcd.setCursor(0,1);  
    lcd.print("X = "); lcd.print(xSet[s]);lcd.print("  
");  
  }  
}  
void settingY(int s){  
  delay(1000);  
  int inc;  
  while(1){  
    if(!digitalRead(T[0])){inc++;}  
    if(!digitalRead(T[2])){inc--;}  
    if(!digitalRead(T[1])){break;}  
    ySet[s]= inc;  
    lcd.setCursor(0,2);  
    lcd.print("Y = "); lcd.print(ySet[s]);lcd.print("  
");  
  }  
}  
void settingS(int s){  
  delay(1000);  
  int inc;  
  while(1){  
    if(!digitalRead(T[0])){inc++;}  
    if(!digitalRead(T[2])){inc--;}  
    if(!digitalRead(T[1])){break;}  
    sSet[s]= inc;  
    lcd.setCursor(0,3);  
    lcd.print("S = "); lcd.print(sSet[s]);lcd.print("  
");  
  }  
}
```



```
void test(float xtar, float ytar, int spd){
float Rtar = sqrt((xtar*xtar)+(ytar*ytar));
float arah = atan2(xtar,ytar)*180/PI;
while(1){
    float tetha = (0.98*Kalman*fc)+(0.02*odo);
    jalan(spd, arah, tetha);
    float Rvalue= sqrt((xValue*xValue)+(yValue*yValue));
    if(Rvalue>=Rtar){break;}
}
}

void jalan(int kec, int heading, int rot){
    baca_gyro();
    if(heading>=0){r_head=((heading-erSud)/180.0)*PI;}
    else{r_head=(((360+heading)-erSud)/180.0)*PI;}
    r_rot=(rot/180.0)*PI;
    m1= kec*cos(r_head-(PI*m1_angle/180.0))-r_rot;
    m2= kec*cos(r_head-(PI*m2_angle/180.0))-r_rot;
    m3= kec*cos(r_head-(PI*m3_angle/180.0))-r_rot;
    m4= kec*cos(r_head-(PI*m4_angle/180.0))-r_rot;
    motor(m1,m2,m3,m4);
}

void motor(int sm1, int sm2, int sm3, int sm4){

    if(sm1 > 0){digitalWrite(MT1[0],HIGH);
digitalWrite(MT1[1],LOW);}
    else if(sm1 < 0){digitalWrite(MT1[1],HIGH);
digitalWrite(MT1[0],LOW);}
    else{digitalWrite(MT1[0],LOW);
digitalWrite(MT1[1],LOW);}
    if(sm2 > 0){digitalWrite(MT2[0],HIGH);
digitalWrite(MT2[1],LOW);}
    else if(sm2 < 0){digitalWrite(MT2[1],HIGH);
digitalWrite(MT2[0],LOW);}
    else{digitalWrite(MT2[0],LOW);
digitalWrite(MT2[1],LOW);}
    if(sm3 > 0){digitalWrite(MT3[0],HIGH);
digitalWrite(MT3[1],LOW);}
    else if(sm3 < 0){digitalWrite(MT3[1],HIGH);
digitalWrite(MT3[0],LOW);}
    else{digitalWrite(MT3[0],LOW);
digitalWrite(MT3[1],LOW);}
```

```
        if(sm4 > 0){digitalWrite(MT4[0],HIGH);
digitalWrite(MT4[1],LOW);}
        else if(sm4 < 0){digitalWrite(MT4[1],HIGH);
digitalWrite(MT4[0],LOW);}
        else{digitalWrite(MT4[0],LOW);
digitalWrite(MT4[1],LOW);}
        analogWrite(MT1[2], abs(sm1));
        analogWrite(MT2[2], abs(sm2));
        analogWrite(MT3[2], abs(sm3));
        analogWrite(MT4[2], abs(sm4));
    }
void saveData(){
    String dataString = "";
    float dataK= Kalman*fc;
    float dataY= yaw*fc;
    dataString +=
String(yValue)+"\t"+String(xValue)+"\t";
    dataString += String(dataK);
    dataString += "\t";
    dataString += String(dataY);
    File dataFile = SD.open("datalog.txt", FILE_WRITE);

    if (dataFile) {
        dataFile.println(dataString);
        dataFile.close();
    }
    else {
        lcd.setCursor(0,1);
        lcd.print("SDcard Error");
    }
}
void baca_gyro() {
    timer = millis();
    Vector norm = gyroscope.readNormalize();
    yaw = yaw + norm.ZAxis * timeStep;
    G=G+varG;
    K=G/(G+varM);
    Kalman=Kalman+(K*(yaw-Kalman));
    G=(1-K)*G;

    lcd.setCursor(0,3);
```

```
lcd.print(Kalman*fc);
delay((timeStep*1000) - (millis() - timer));
}

void updateEncoder(){

int encoded1 = (digitalRead(encoder1a) << 1) |
digitalRead(encoder1b);
int encoded2 = (digitalRead(encoder2a) << 1) |
digitalRead(encoder2b);

int sum1 = (lastEncoded1 << 2) | encoded1;
int sum2 = (lastEncoded2 << 2) | encoded2;
switch (sum1){
case 0b1101 : {encoderValue1++;break;}
case 0b0100 : {encoderValue1++;break;}
case 0b0010 : {encoderValue1++;break;}
case 0b1011 : {encoderValue1++;break;}
case 0b1110 : {encoderValue1--;break;}
case 0b0111 : {encoderValue1--;break;}
case 0b0001 : {encoderValue1--;break;}
case 0b1000 : {encoderValue1--;break;}
}
switch (sum2){
case 0b1101 : {encoderValue2++;break;}
case 0b0100 : {encoderValue2++;break;}
case 0b0010 : {encoderValue2++;break;}
case 0b1011 : {encoderValue2++;break;}
case 0b1110 : {encoderValue2--;break;}
case 0b0111 : {encoderValue2--;break;}
case 0b0001 : {encoderValue2--;break;}
case 0b1000 : {encoderValue2--;break;}
}
xValue=(float)encoderValue2*0.865592;
yValue=(float)encoderValue1*0.865592;
lastEncoded1 = encoded1;
lastEncoded2 = encoded2;
EICRA=0x50;
EICRB=0x05;
EIMSK= xflag | yflag;
EIFR= xflag | yflag;}
```