



PROYEK AKHIR - VE 180626

**SISTEM KENDALI UNTUK ROBOT MOBIL 4 RODA
MENGUNAKAN MEKANISME SWERVE DRIVE**

Moh Ismarintan Zazuli
NRP 10311600000091

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng.
Fajar Budiman, ST.M.Sc.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019



FINAL PROJECT - VE 180626

***CONTROL SYSTEM FOR 4 WHEEL CAR ROBOT USING
SWERVE DRIVE MECHANISM***

Moh Ismarintan Zazuli
NOR 10311600000091

Supervisor
Ir. Josaphat Pramudijanto, M.Eng.
Fajar Budiman, ST.M.Sc.

***ELECTRICAL AND AUTOMATION ENGINEERING DEPARTMENT
Vocational Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2019***

PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir saya dengan judul “**Sistem Kendali Untuk Robot Mobil 4 Roda Menggunakan Mekanisme Swerve Drive**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2019

Moh Ismarintan Zazuli
NRP 10311600000091

**SISTEM KENDALI UNTUK ROBOT MOBIL 4 RODA
MENGUNAKAN MEKANISME SWERVE DRIVE**

PROYEK AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada
Program Studi Elektro Industri
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui

Dosen Pembimbing 1

Dosen Pembimbing 2

Ir. Josaphat Pramudijanto M.Eng
NIP. 19621005 199003 1 003

Fajar Budiman S.T, M.Sc.
19860707 201404 1 001

Surabaya
JULI, 2019

-----Halaman ini sengaja dikosongkan-----

SISTEM KENDALI UNTUK ROBOT MOBIL 4 RODA MENGUNAKAN MEKANISME SWERVE DRIVE

Nama Mahasiswa : Moh Ismarintan Zazuli
NRP : 10311600000091
Dosen Pembimbing 1 : Ir. Josaphat Pramudijanto, M.Eng
NIP : 19621005 199003 1 003
Dosen Pembimbing 2 : Fajar Budiman S.T, M.Sc.
NIP : 19860707 201404 1 001

ABSTRAK

Robot mobil adalah robot yang dapat berjalan otomatis sesuai dengan jalur yang telah ditentukan. dan dibuat untuk mengangkut barang dari suatu tempat ke tempat tertentu. Untuk dapat bergerak ke segala arah tanpa memutar sudut dari robot maka diperlukan mekanisme roda yang dapat berputar ke segala arah secara independen pada masing-masing roda. Sehingga gerakan dari robot lebih dinamis dan efisien.

Pada mekanisme *Swerve Drive* ini digunakan 4 roda *steering* yang dapat berputar ke segala arah serta masing-masing roda penentuan arah sudut putar roda masing-masing berdasarkan arah yang dituju oleh robot. Untuk memutar *steering* dari roda robot tersebut digunakan motor DC yang dihubungkan dengan gir dan roda, serta digunakan *rotarry encoder* untuk mengetahui nilai sudut pada roda. Sudut dari roda akan ditentukan berdasarkan arah tujuan dari robot.

Berdasarkan hasil pengujian keseluruhan sistem robot dapat bergerak secara *omnidirectional* yaitu bergerak ke segala arah, ke kanan, kiri, depan dan ke belakang tanpa harus memutar arah hadap robot yaitu dengan memutar *steering* roda dimana nilai kesalahan terbesar dari sudut *steering* roda tersebut adalah 1,8 derajat dan nilai rata-rata kesalahan 0,47 derajat terhadap semua arah tujuan.

Kata kunci : Swerve Drive, Abu Robocon, Omni Directional Robot

-----Halaman ini sengaja dikosongkan-----

CONTROL SYSTEM FOR 4 WHEEL CAR ROBOT USING SWERVE DRIVE MECHANISM

Student Name : Moh Ismarintan Zazuli
Registration Number : 10311600000091
Supervisor I : Ir. Josaphat Pramudijanto, M.Eng
ID : 19621005 199003 1 003
Supervisor II : Fajar Budiman S.T, M.Sc.
ID : 19860707 201404 1 001

ABSTRACT

Cars robots are robots that can run automatically in accordance with a predetermined path. and made to transport goods from one place to a certain place. To be able to move in all directions without turning the angle of the robot, the wheel mechanism is added which can rotate in all directions independently on each wheel. So that the robot movement is more dynamic and efficient

In the Swerve Drive mechanism 4 steering wheels are used which can rotate in all directions and each wheel determines the direction of each wheel's turning angle based on the direction of the robot. To rotate the steering wheel of the robot wheel, a DC motor is connected to the gear and wheels, and rotary encoder is used to determine the angle value of the wheel. The angle of the wheel will be determined based on the direction of the direction of the robot

Based on the results of testing the entire system of robots can move omnidirectionally that is moving in all directions, right, left, front and back without having to rotate the direction of the robot by rotating the steering wheel where the error value from the steering wheel angle is 1,8 degrees towards the direction, and the average error value is 0,47 degrees for every direction .

Keywords : Swerve Drive, ABU Robocon, Omni Directional Robot

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Alhamdulillah kami panjatkan kepada Allah Subhanahu Wa Ta'ala, atas limpahan rahmat dan kemudahan dariNya, hingga kami dapat menyelesaikan Proyek Akhir ini dengan baik, begitu pula dengan pembuatan buku Proyek Akhir ini.

Proyek Akhir ini dilakukan untuk memenuhi beban satuan kredit semester (SKS) yang harus ditempuh sebagai persyaratan akademis di Departemen Teknik Elektro Otomasi Fakultas Vokasi Institut Teknologi Sepuluh Nopember Surabaya untuk menyelesaikan program pendidikan Diploma di Teknik Elektro dengan judul :

“Sistem Kendali Untuk Robot Mobil 4 Roda Menggunakan Mekanisme Swerve Drive”

Dengan terselesainya Proyek Akhir ini Penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Kedua orangtua yang senantiasa mendoakan dan memberikan dukungan dengan tulus tiada henti.
2. Bapak Ir. Joko Susila, M.T. selaku Kepala Departemen Teknik Elektro Otomasi.
3. Bapak Ir. Josaphat Pramudijanto, M.Eng. selaku dosen pembimbing 1 dengan segala bimbingan ilmu, moral, dan spiritual.
4. Bapak Fajar Budiman S.T, M.Sc. selaku dosen pembimbing 2 yang telah memberikan bimbingan dan ilmunya.
5. Team ABU Robocon ITS yang membantu dalam proses pembuatan robot.
6. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Proyek Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Proyek Akhir ini. Akhir kata, semoga Proyek Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, Juli 2019

Moh Ismarintan Zazuli

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

HALAMAN

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN PROYEK AKHIR	v
LEMBAR PENGESAHAN	vii
ABSTRAK.....	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xix
 BAB I PENDAHULUAN.....	 1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Tujuan	2
1.4 Sistematika Laporan.....	2
1.5 Relevansi.....	3
 BAB II TEORI DASAR	 5
2.1 Robot <i>Omnidirectional</i>	5
2.2 Mekanisme <i>Swerve Drive</i>	6
2.3 <i>Rotarry Encoder</i>	7
2.4 Motor DC	9
2.5 Mikrokontroler STM32 F4 <i>Discovery</i>	10
2.6 <i>Driver</i> Motor BTS7960B	11
2.7 LCD 20 x 4	13
2.8 Atollic <i>TrueStudio</i>	14
 BAB III PERANCANGAN ALAT	 15
3.1 Perancangan Desain Mekanik	15
3.1.1 Desain Roda <i>Sewerve Drive</i>	16
3.1.2 Desain <i>Base</i> Robot.....	19
3.2 Perancangan Sistem Elektronik.....	19
3.2.1 Konfigurasi Sistem Elektronik Robot Secara Keseluruhan	20
3.2.2 Konfigurasi Pin Elektronik Pada Mikro <i>Slave</i>	21

3.2.3	Konfigurasi Pin Elektronik Pada Mikro <i>Master</i> ...	22
3.3	Perancangan <i>Software</i>	23
3.3.1	Perancangan Program Data <i>Encoder</i> Dan Merubah Derajat Sudut Roda.....	24
3.3.2	Perancangan Program Kendali Roda	25
3.3.3	Perancangan Program Kinematika Robot	28
3.3.4	Perancangan Program Utama Robot.....	30
BAB IV PENGUJIAN DAN ANALISA.....		33
4.1	Pengujian <i>Rotary Encoder</i> Pada <i>Steering</i> Roda.....	33
4.2	Pengujian Motor DC <i>Drive</i>	36
4.3	Pengujian <i>Steering</i> Roda.....	39
4.4	Pengujian Kendali Arah Roda.....	42
4.5	Pengujian Pergerakan Robot	46
BAB V PENUTUP		49
5.1	Kesimpulan	49
5.2	Saran	49
DAFTAR PUSTAKA		51
LAMPIRAN A		A-1
LAMPIRAN B.....		B-1
LAMPIRAN C.....		C-1
DAFTAR RIWAYAT HIDUP		D-1

DAFTAR GAMBAR

HALAMAN

Gambar 2.1 Sistem Pergerakan Roda <i>Standard</i> dan <i>Omnidirectional</i> ..	5
Gambar 2.2 Contoh Roda <i>Swerve Drive</i>	6
Gambar 2.3 <i>Phase Line Rotary Encoder</i>	7
Gambar 2.4 Bentuk Fisik <i>Rotary Encoder</i> Autonic E30S4	8
Gambar 2.5 Motor DC Mabuchi RS775W	10
Gambar 2.6 Mikrokontroler STM32 F4 <i>Discovery</i>	11
Gambar 2.7 Driver Motor BTS7960B	12
Gambar 2.8 Rangkaian <i>Full H-Bridge</i> BTS7960B	12
Gambar 2.9 LCD 20x4.....	13
Gambar 2.10 Tampilan Atollic TrueStudio	14
Gambar 3.1 Alur Perancangan Sistem	15
Gambar 3.2 Desain <i>Base</i> Roda	16
Gambar 3.3 <i>Gearset Steering</i>	17
Gambar 3.4 <i>Gearset Drive</i>	18
Gambar 3.5 Desain Keseluruhan Roda.....	18
Gambar 3.6 Desain <i>Base</i> Robot	19
Gambar 3.7 Penomoran Roda Robot	20
Gambar 3.8 Konfigurasi Elektronik.....	21
Gambar 3.9 Skematik Sistem Roda	23
Gambar 3.10 Konversi Data <i>Encoder</i>	25
Gambar 3.11 Kendali <i>Steering Roda</i>	26
Gambar 3.12 Kendali Pada Motor <i>Drive</i>	27
Gambar 3.13 Arah Vektor Kecepatan Robot	28
Gambar 3.14 Arah Roda Saat Robot Berotasi	29
Gambar 3.15 <i>Flowchart</i> Program Utama.....	31
Gambar 4.1 Arah Putar Roda.....	33
Gambar 4.2 Program PWM <i>Duty Cycle</i> 10% dan 30%	36
Gambar 4.3 Ilustrasi Pengujian Motor DC <i>Drive</i>	36
Gambar 4.4 Tampilan Kecepatan Roda <i>Duty Cycle</i> 30% Beban 4Kg..	37
Gambar 4.5 Penurunan Kecepatan Roda Terhadap Beban	38
Gambar 4.6 Respon <i>Steering</i> Kp 0.2	39
Gambar 4.7 Respon <i>Steering</i> Kp 0.5	40

Gambar 4.8 <i>Respon Steering</i> Kp 0.65.....	40
Gambar 4.9 Respon <i>Steering</i> Kp 0.75	41
Gambar 4.10 Respon <i>Steering</i> Kp 0.8	41
Gambar 4.11 Respon <i>Steering</i> Kp 0.75 dan Kd 0,2	42
Gambar 4.12 Pengujian Arah Roda Dengan Perintah Stik Maju	43
Gambar 4.13 Pergerakan Robot.....	46

DAFTAR TABEL

HALAMAN

Tabel 2.1 Spesifikasi <i>Rotary Encoder</i> Autronics E30S4.....	8
Tabel 2.2 Spesifikasi Kaki pada LCD 20x4.....	13
Tabel 3.1 Penomoran Roda.....	20
Tabel 3.2 Tabel Koneksi Pin STM32 F4 <i>Discovery</i>	21
Tabel 3.3 Koneksi Pin Mikro <i>Master</i> STM32	22
Tabel 3.4 Sudut Roda Saat Berotasi.....	29
Tabel 4.1 Hasil Pengujian Putaran Roda Searah Jarum Jam.....	34
Tabel 4.2 Hasil Pengujian Putaran Roda Berlawanan Arah Jarum Jam	34
Tabel 4.3 Perbandingan Data Pengukuran Encoder Roda Dengan Perhitungan	35
Tabel 4.4 Hasil Pengujian Kecepatan Motor DC.....	37
Tabel 4.5 Nilai <i>Encoder</i> Eoda 0 <i>Steering</i> Saat Diberikan Perintah Stik	43
Tabel 4.6 Nilai <i>Encoder</i> Eoda 1 <i>Steering</i> Saat Diberikan Perintah Stik	44
Tabel 4.7 Nilai <i>Encoder</i> Eoda 2 <i>Steering</i> Saat Diberikan Perintah Stik	45
Tabel 4.8 Nilai <i>Encoder</i> Eoda 3 <i>Steering</i> Saat Diberikan Perintah Stik	45
Tabel 4.9 Pengujian Gerakan Robot Jarak 120 cm.....	47
Tabel 4.10 Pengujian Gerakan Robot Jarak 3 m.....	47

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kontes Robot Abu Indonesia merupakan ajang perlombaan nasional yang diadakan setiap tahun. Kontes Robot ABU Indonesia (KRAI) 2019 mengacu pada ABU Asia-Pacific Robot Contest 2019 Mongolia. Dimana robot harus membawa sebuah barang dan harus melewati berbagai rintangan dengan melakukan berbagai macam manuver seperti jalan berkelok-kelok, jalan dengan memutar serta jalan lurus sehingga diperlukan robot berbentuk mobil yang dapat melakukan misi tersebut. Robot Mobil atau *Mobile Robot* adalah konstruksi robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain. Namun apabila menggunakan roda yang memiliki sudut tetap maka gerakan dari robot akan lebih sulit untuk melakukan berbagai macam manuver.

Pada sistem pergerakan konvensional, pergerakan tidak mampu di kontrol pada setiap tingkat kebebasan dalam bergerak secara independen, sehingga hanya mampu bergerak ke beberapa arah yang sudah ditentukan sebelumnya. Untuk itu diperlukan robot mobil yang dapat bergerak kesegala arah tanpa harus mengubah arah hadap robot tersebut atau disebut juga dengan *omnidirectional drive*. Mekanisme tersebut dapat menggunakan berbagai macam solusi seperti menggunakan roda *omni* atau roda *mechanum* namun penggunaan roda tersebut memiliki kelemahan yaitu selip pada roda yang mengakibatkan percepatan pada roda sedikit lambat dibandingkan dengan roda konvensional.

Maka dari itu diperlukan mekanisme roda konvensional namun sudut roda tersebut dapat diputar sesuai dengan arah tujuan atau dinamakan dengan mekanisme *swerve drive*. Sehingga gerakan dari robot lebih cepat dan efisien. Dengan menggunakan mekanisme *swerve drive* tersebut maka robot dapat bergerak secara *omnidirectional* namun tetap memiliki traksi yang kuat pada masing-masing roda dikarenakan tidak banyak terjadi selip pada permukaan roda. Mekanisme *swerve drive* tersebut memanfaatkan *steering* pada masing-masing roda untuk menentukan arah pergerakan dari robot. Dengan menggunakan mekanisme *swerve drive* maka robot tidak perlu memutar untuk

melakukan gerakan berbelok ataupun berbagai macam gerakan manuver lain. Sehingga akan lebih menghemat waktu dan efisiensi robot dalam melakukan *trajectory* yang telah ditentukan.

1.2 Permasalahan

Perumusan masalah yang dibahas dalam pengerjaan Proyek Akhir ini adalah sebagai berikut:

1. Robot mobil dengan roda standar memerlukan mengganti arah hadapnya untuk berbelok ke kanan atau ke kiri.
2. Penggunaan roda omni sebagai mekanisme *omnidirectional* memiliki selip pada rodanya yang mengakibatkan berkurangnya akselerasi pada kecepatannya.
3. Berkurangnya kecepatan saat melakukan gerakan belok pada robot mobil dengan roda standar yang dikarenakan harus menyeimbangkannya kecepatan roda kanan dan kiri.

1.3 Tujuan

Tujuan yang akan dicapai dalam penyelesaian Proyek Akhir ini adalah sebagai berikut:

1. Mampu membuat robot mobil yang dapat bergerak ke segala arah tanpa merubah arah hadapnya.
2. Mampu mengendalikan sudut *steering* pada roda untuk mendapatkan berbagai macam gerakan manuever.
3. Mengetahui cara mendapatkan nilai masing-masing sudut *steering* roda pada berbagai macam gerakan translasi dan rotasi.

1.4 Sistematika Laporan

Pembahasan Proyek Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini membahas teori dasar, teori penunjang dari peralatan-peralatan yang digunakan dalam pembuatan alat.

Bab III Perancangan Sistem

Bab ini membahas mengenai desain perancangan dan pembuatan perangkat keras (*hardware*) yang terdiri dari atas

perancangan rangkaian elektronika, perancangan mekanik serta perangkat lunak (*software*) yang terdiri atas program untuk Perancangan program data *encoder* dan merubah derajat sudut roda, perancangan program kendali *steering* dan kecepatan roda, serta perancangan program kinematika robot.

Bab IV Pengujian dan Analisa Sistem

Bab ini membahas tentang pengukuran, pengujian alat dan analisa data yang didapat dalam pengujian alat.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.5 Relevansi

Pada penyelesaian Proyek Akhir ini diharapkan dapat memberikan manfaat, diantaranya sebagai berikut:

1. Menjadi referensi penelitian dimasa yang akan datang.
2. Menghasilkan robot yang memiliki kelincahan dalam pergerakannya.
3. Membantu tim ABU Robocon ITS dalam perlombaan Kontes Robot Abu Indonesia.

-----Halaman ini sengaja dikosongkan-----

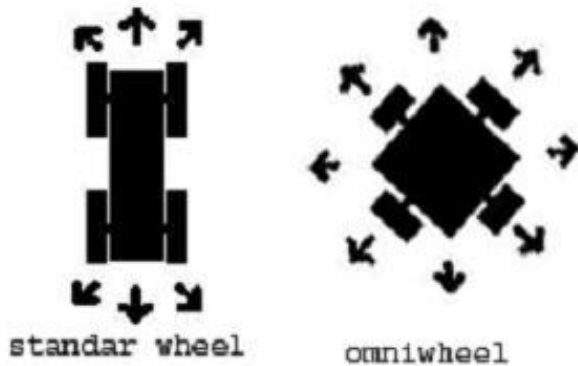
BAB II

TEORI DASAR

Bab ini membahas mengenai teori penunjang dari peralatan yang digunakan dalam Sistem kendali untuk robot mobil 4 roda menggunakan mekanisme *swerve drive* seperti sensor *rotary encoder*, STM32 F4 *Discovery*, Motor DC, LCD 20 x 4, kinematika robot dan lain sebagainya.

2.1 Robot *Omnidirectional* [1]

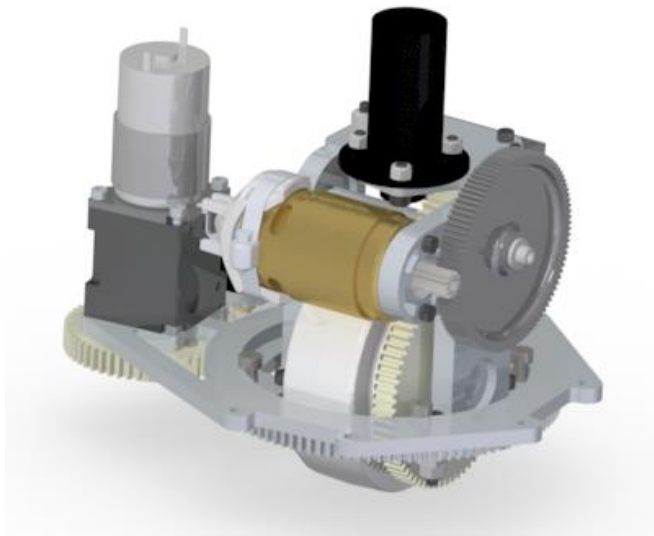
Robot *omnidirectional* adalah robot dengan sistem pergerakan yang secara langsung dapat bergerak kesegala arah dengan konfigurasi apapun. Pada umumnya robot di desain dengan pergerakan yang sudah direncanakan terlebih dahulu. Pada sistem pergerakan konvensional, pergerakan tidak mampu di kontrol pada setiap tingkat kebebasan dalam bergerak secara independen, sehingga hanya mampu bergerak ke beberapa arah yang sudah ditentukan sebelumnya. Ini disebut kendala *non-holomic* yaitu pencegahan roda kemudi dari selip, meskipun pada umumnya mampu menjangkau setiap lokasi dan orientasi dalam ruang 2 dimensi, namun memerlukan manuver dan perencanaan jalan yang rumit dan kompleks. Perbandingan dari sistem gerak roda biasa dan omni digambarkan pada Gambar 2.1.



Gambar 2.1 Sistem Pergerakan Roda Standar dan *Omnidirectional*

2.2 Mekanisme Swerve Drive [2]

Untuk bergerak secara *omnidirectional* dapat digunakan beberapa jenis roda, antara lain roda *omni*, roda *mecanum*, roda *swerve drive* dan lain sebagainya. Mekanisme *swerve drive* sendiri adalah salah satu solusi untuk robot *omnidirectional*. Dimana setiap roda memiliki *steering* yang dapat memutar sudut roda kesegala arah, umumnya terdapat dua aktuator pada masing-masing roda, dimana satu sebagai penggerak dari *steering* roda dan satu sebagai penggerak dari roda sendiri seperti Gambar 2.2.



Gambar 2.2 Contoh Roda Swerve Drive

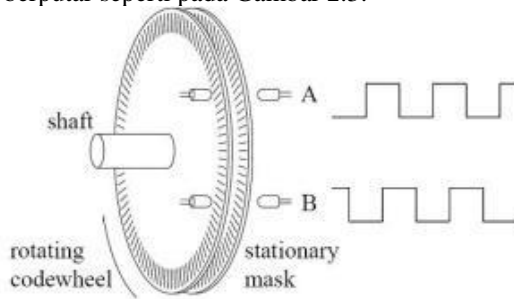
Beberapa kelebihan dari mekanisme roda *swerve drive* adalah sebagai berikut.

- a. Arah *drive* dapat dipisahkan dari arah orientasi *casis*.
- b. Memiliki traksi lebih besar dibanding dengan roda *omnidirectional* lainnya.
- c. Dapat bergerak secara fleksibel ke segala arah tanpa merubah orientasi dari robot.
- d. Efisien untuk gerakan manuver dengan cepat.

2.3 Rotary Encoder [3]

Rotary encoder adalah *device* 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 *device* berputar lainnya yang ingin kita ketahui posisinya, sehingga ketika motor berputar piringan juga akan ikut berputar seperti pada Gambar 2.3.



Gambar 2.3 *Phase Line Rotary Encoder*

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. 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.

Pada robot ini digunakan *rotary encoder tipe incremental*. *incremental encoder* terdiri dari dua *track* atau *single track* dan dua sensor yang disebut *channel A* dan *B*. 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. Bentuk fisik dari *encoder* yang digunakan yaitu Autonic E30S4 adalah seperti pada Gambar 2.4.



Gambar 2.4 Bentuk Fisik *Rotary Encoder* Autonic E30S4

Untuk spesifikasi dari modul sensor ini dapat dilihat pada Tabel 2.1.

Tabel 2.1 Spesifikasi *Rotary Encoder* Autonics E30S4.

Spesifikasi	Keterangan
<i>Resolution (P/R)</i>	200
<i>Working Voltage</i>	12VDC – 24VDC
<i>Output Type</i>	<i>NPN Open Colector Output</i>
<i>Max. Response Frequency</i>	300 KHz
<i>Max. Allowable revolution</i>	5000 rpm
<i>Output Phaze</i>	<i>A,B,Z phaze</i>
<i>Diameter Dimension</i>	30 mm

Spesifikasi	Keterangan
<i>Lenght</i>	28,5 mm
<i>Shaft Dimension</i>	4 mm

2.4 Motor DC [4]

Motor DC adalah suatu perangkat yang mengubah energi listrik menjadi energi kinetik atau gerakan (*motion*). Motor DC ini juga dapat disebut sebagai Motor Arus Searah. Seperti namanya, DC Motor memiliki dua terminal dan memerlukan tegangan arus searah atau DC (*Direct Current*) untuk dapat menggerakkannya. Motor Listrik DC ini biasanya digunakan pada perangkat-perangkat Elektronik dan listrik yang menggunakan sumber listrik DC seperti *Vibrator* Ponsel, Kipas DC dan Bor Listrik DC. Motor Listrik DC atau *DC Motor* ini menghasilkan sejumlah putaran per menit atau biasanya dikenal dengan istilah RPM (*Revolutions per minute*) dan dapat dibuat berputar searah jarum jam maupun berlawanan arah jarum jam apabila polaritas listrik yang diberikan pada Motor DC tersebut dibalik. Motor Listrik DC tersedia dalam berbagai ukuran rpm dan bentuk. Kebanyakan Motor Listrik DC memberikan kecepatan rotasi sekitar 3000 rpm hingga 8000 rpm dengan tegangan operasional dari 1,5V hingga 24V. Apabila tegangan yang diberikan ke Motor Listrik DC lebih rendah dari tegangan operasionalnya maka akan dapat memperlambat rotasi motor DC tersebut sedangkan tegangan yang lebih tinggi dari tegangan operasional akan membuat rotasi motor DC menjadi lebih cepat. Namun ketika tegangan yang diberikan ke Motor DC tersebut turun menjadi dibawah 50% dari tegangan operasional yang ditentukan maka Motor DC tersebut tidak dapat berputar atau terhenti. Sebaliknya, jika tegangan yang diberikan ke Motor DC tersebut lebih tinggi sekitar 30% dari tegangan operasional yang ditentukan, maka motor DC tersebut akan menjadi sangat panas dan akhirnya akan menjadi rusak. Pada Proyek Akhir ini digunakan motor DC dengan Mabuchi dengan seri RS 775W dimana bentuk fisik dari motor tersebut seperti pada Gambar 2.5.

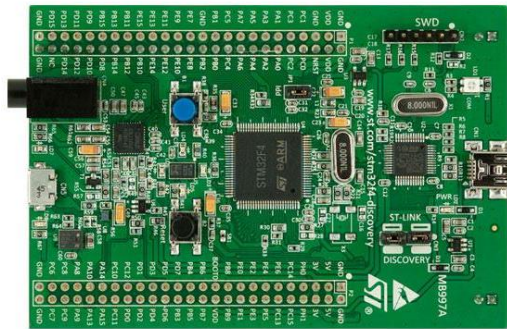


Gambar 2.5 Motor DC Mabuchi RS775W

Pada saat Motor listrik DC berputar tanpa beban, hanya sedikit arus listrik atau daya yang digunakannya, namun pada saat diberikan beban, jumlah arus yang digunakan akan meningkat hingga ratusan persen bahkan hingga 1000% atau lebih (tergantung jenis beban yang diberikan). Oleh karena itu produsen Motor DC biasanya akan mencantumkan *Stall Current* pada Motor DC. *Stall Current* adalah arus pada saat poros motor berhenti karena mengalami beban maksimal.

2.5 Mikrokontroler STM32 F4 Discovery[5]

STM32F4 *Discovery* adalah sebuah modul mikrokontroler yang di dalamnya menggunakan IC STM32F407VGT6 ARM Cortex-M4 yang mempunyai kecepatan sampai dengan 168 MHz, serta mampu mengeksekusi perintah hingga 210 MIPS (*Million Instruction per Second*). STM32F4 *Discovery* merupakan mikrokontroler 32 bit dengan arsitektur ARM. Mikrokontroller ini memiliki kapasitas 1 MByte Flash PEROM (*Flash Programmable and Eraseble Read Only Memory*), 192 Kbyte SRAM. Dilengkapi dengan 100 buah pin *input output* yang mempunyai karakteristik masing-masing yaitu USART, TIMER, ADC dan I2C bentuk fisik dari mikrokontroler STM32 F4 *Discovery* tersebut seperti pada Gambar 2.6.



Gambar 2.6 Mikrokontroler STM32 F4 *Discovery*

Mikrokontroler STM32 F4 *Discovery* memiliki beberapa fitur dan spesifikasi sebagai berikut .

- On-board ST-LINK/V2 yang digunakan untuk proses *upload* file hex ke dalam mikrokontroler.
- ST MEMS *motion* sensor, 3-axis *accelerometer*.
- Audio DAC dengan *driver speaker* kelas D.
- Dua *push button* (*user* dan *reset*).
- USB OTG.

Bahasa pemrograman pada mikrokontroler ini adalah bahasa C. Media pemrograman yang dapat digunakan untuk memprogram STM32F4 *Discovery* yaitu antara lain :

- Altium, TASKING VX-Toolset.
- Atollic TrueSTUDIO.
- IAR *Embedded Workbench for ARM* (EWARM).
- Keil, MDK-ARM.
- Coocox.

2.6 **Driver Motor BTS7960B[6]**

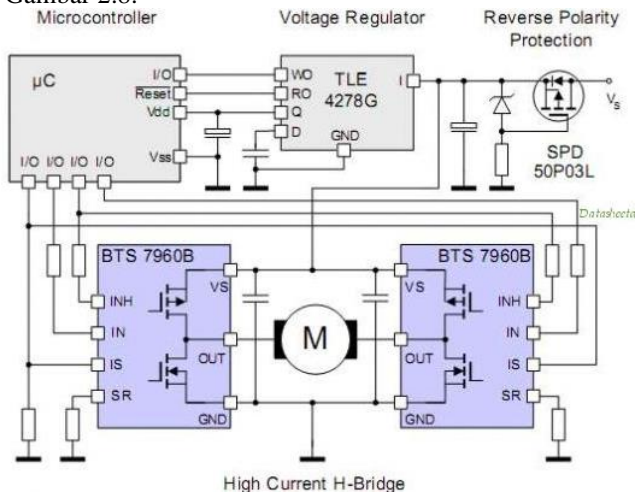
Untuk mengendalikan arah dan kecepatan putaran dari motor DC diperlukan komponen yang disebut dengan *Driver Motor*. Dengan kontrol PWM kita dapat mengatur kecepatan motor dengan memberikan pulsa dengan frekuensi yang tetap ke motor, sedangkan yang digunakan untuk mengatur kecepatan adalah *duty cycle* dari pulsa yang diberikan.

Pada *driver* motor ini terdapat 3 pin kendali, yaitu pin In1 Sebagai Direksi 1, pin In2 sebagai Direksi 2, dan V in sebagai *input* PWM pada *Driver* motor, bentuk fisik dari *driver* motor BTS7960B dapat dilihat pada Gambar 2.7.



Gambar 2.7 *Driver* Motor BTS7960B

Didalam IC BTN7960 terdapat rangkaian Half-Bridge yang berfungsi sebagai kendali dari *driver* motor. Maka untuk membuat rangkaian *full H-Bridge* diperlukan dua IC didalam satu modul tersebut seperti Gambar 2.8.

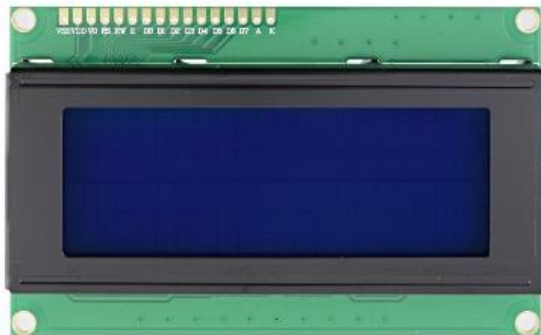


Gambar 2.8 Rangkaian *Full H-Bridge* BTS7960B

2.7 LCD 20 x 4 [7]

Display elektronik adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf maupun grafik. LCD (*Liquid Crystal Display*) adalah suatu jenis display elektronik. LCD merupakan media tampil yang menggunakan kristal cair sebagai penampil utama.

Salah satu jenis LCD (*Liquid Crystal Display*) yang sering digunakan ialah LCD dot matrik dengan jumlah karakter 16x4. LCD sebagai salah satu jenis *display* elektronik bekerja dengan tidak menghasilkan cahaya. LCD memantulkan cahaya yang ada disekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lite*. Spesifikasi kaki LCD 20x4 dapat dilihat pada Tabel 2.2. Dan untuk bentuk fisiknya dapat dilihat pada Gambar 2.9.



Gambar 2.9 LCD 20x4

LCD 20x4 ini memiliki pin yang berfungsi sebagai penghubung kendali dari mikrokontroler dengan lcd dimana pin-pin tersebut mempunyai fungsi masing-masing seperti yang ada pada Tabel 2.2.

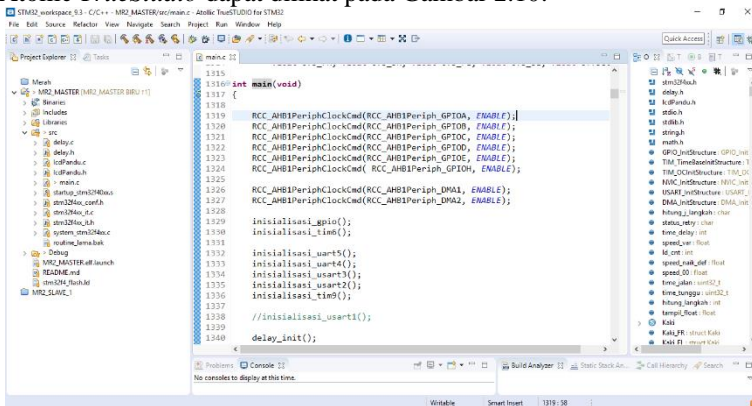
Tabel 2.2 Spesifikasi Kaki pada LCD 20x4

No	Pin LCD	Keterangan
3.1	Pin 1	merupakan <i>Ground</i>
3.2	Pin 2	merupakan VCC (+3V atau +5V)
3.3	Pin 3	merupakan pengatur kontras.

No	Pin LCD	Keterangan
3.4	Pin 4	merupakan RS (<i>Register Select</i>) <i>instruction</i> .
3.5	Pin 5	merupakan R/W (<i>Read/Write</i>) LCD register.
3.6	Pin 6	merupakan EN (<i>Enable</i>).
3.7	Pin 7 - 14	merupakan I/O (<i>Input/Output</i>) data.
3.8	Pin 15	merupakan VCC
3.9	Pin 16	merupakan <i>Ground</i>

2.8 Atollic TrueStudio [8]

Atollic TrueStudio adalah perangkat lunak untuk, memprogram, *debugging*, dan *monitoring* mikrokontroler STM. Atollic TrueStudio Menggunakan bahasa C untuk pemrogramannya. melalui *software* inilah STM32 dilakukan pemrograman untuk melakukan fungsi-fungsi yang diberikan melalui *sintaks* pemrograman, tampilan dari *software* Atollic *TrueStudio* dapat dilihat pada Gambar 2.10.



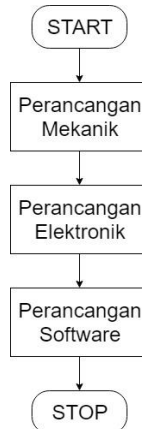
Gambar 2.10 Tampilan Atollic *TrueStudio*

Atollic TrueStudio memiliki beberapa fitur seperti, CPU *fault analyzer*, *Global expressions live watch*, *Real-time data access tracing* (SWV), RTOS *Kernel-aware debugging*, *Exception and Interrupt tracing* (SWV) dan masih banyak lagi.

BAB III

PERANCANGAN ALAT

Pada Bab ini akan dijelaskan mengenai perancangan dari robot yang terbagi menjadi tiga bagian yaitu perancangan mekanik, perancangan elektronik, dan perancangan *software*. Perancangan mekanik dilakukan terlebih dahulu kemudian dilanjutkan dengan perancangan elektronik. Perancangan mekanik dan elektronik ini dilakukan pada proses pertama kali dikarenakan perhitungan-perhitungan pada perancangan *software* mengacu pada perancangan mekanik dan elektronik, sehingga perancangan elektronik dan mekanik dilakukan pertama kali seperti pada Gambar 3.1.



Gambar 3.1 Alur Perancangan Sistem

3.1 Perancangan Desain Mekanik

Perancangan mekanik terdiri dari desain roda *swerve drive* dan desain base robot. Pada proses desain robot mengacu pada peraturan ABU Asia Pasific Robot Contest 2019 yaitu ukuran maksimal saat robot pertama di garis *start* adalah panjang 1000mm x lebar 1000mm dan saat permainan di mulai dimensi maksimal adalah 1500 mm untuk keseluruhan panjang, lebar, dan tinggi.

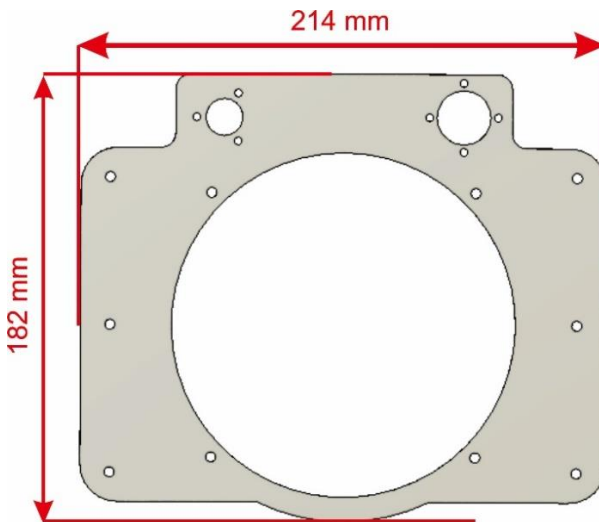
3.1.1 Desain Roda *Sewerve Drive*

Sistem mekanik roda pada robot didesain supaya robot dapat bergerak secara *omnidirectional*. Sehingga robot dapat bergerak secara bebas ke segala namun juga memiliki kecepatan yang tinggi dalam melakukan berbagai gerakan manuver. Roda *swerve drive* sendiri adalah roda yang mempunyai dua bagian penting yakni bagian *steering* sebagai penggerak arah sudut roda, dan bagian *Drive* sebagai penggerak dari putaran roda.

Pada bagian-bagian tersebut terdapat beberapa komponen penting yakni *gear*, *base* roda, *mounting* motor dc , dan *mountinting rotary encoder*.

3.1.1.1 Base Roda

Fungsi dari *base* roda ini adalah sebagai penopang komponen-komponen roda seperti motor dc, *rotary encoder*, dan *gear*, spesifikasi dari *Base* Roda seperti ditampilkan pada Gambar 3.2.



Gambar 3.2 Desain *Base* Roda

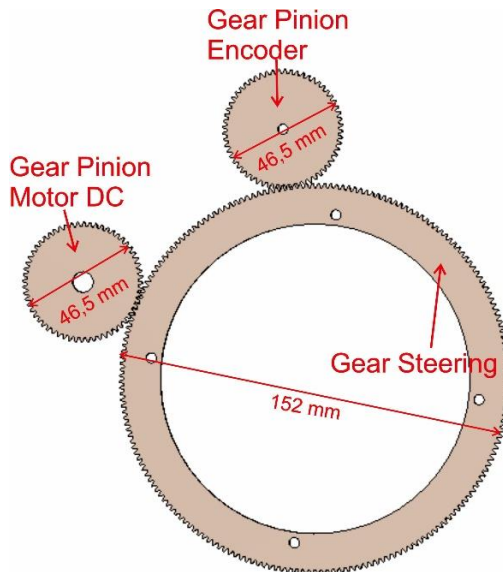
Base roda ini terbuat dari plat aluminium yang memiliki tebal 3 mm dan panjang 214 mm dan lebar 182 mm.

3.1.1.2 Desain Gear Set Roda

Gear set roda berfungsi sebagai penghubung antara motor dc dan roda serta *rotary encoder*, masing-masing *gear* memiliki bentuk dan rasio sendiri-sendiri.

a. *Gear Steering*

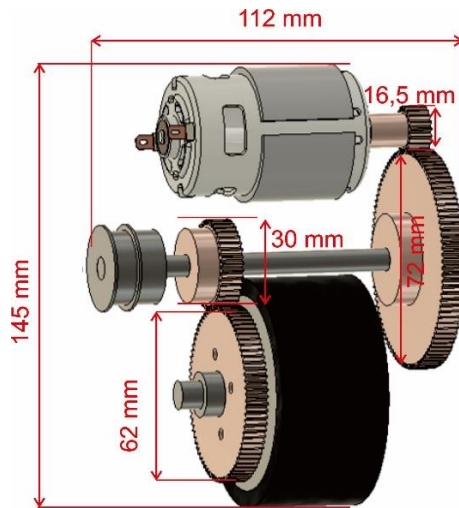
Rasio *gear pinion* motor dc : *gear pinion* encoder : *gear steering* adalah 3,3 : 3,3 : 1 atau dapat diartikan jika *gear steering* berputar 1 kali putaran maka *gear pinion* motor dc dan *gear pinion encoder* telah berputar 3,3 kali putaran. Dimensi dan desain dari *gear* tersebut adalah seperti Gambar 3.3.



Gambar 3.3 *Gearset Steering*

b. *Gear Drive*

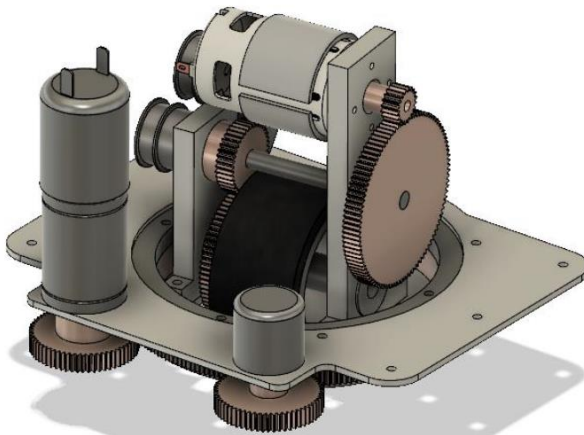
Gear drive berfungsi untuk menghubungkan motor dc *drive* dengan roda dan *rotary encoder*, masing – masing *gear* tersebut memiliki ukuran dimensi seperti pada Gambar 3.4.



Gambar 3.4 *Gearset Drive*

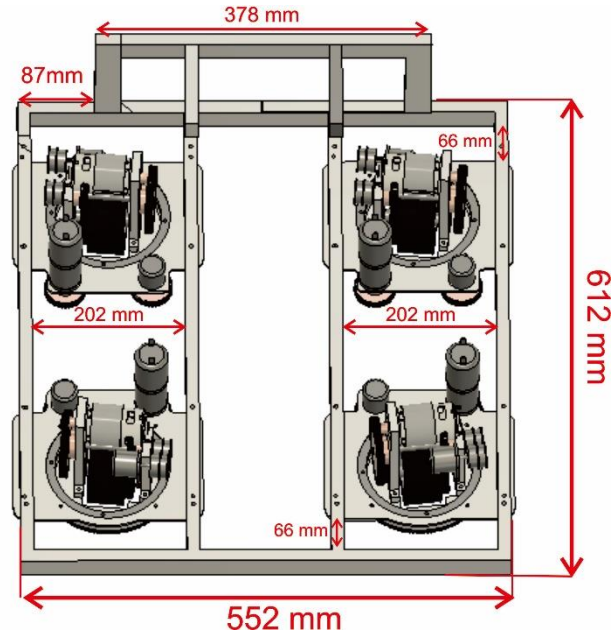
3.1.1.3 Desain Keseluruhan Roda

Seluruh bagaian dari roda disatukan hingga membentuk roda yang dapat berputar serta memiliki *steering* 360° seperti Gambar 3.5.



Gambar 3.5 Desain Keseluruhan Roda

3.1.2 Desain *Base Robot*



Gambar 3.6 Desain *Base Robot*

Pada Gambar 3.6 tersebut adalah spesifikasi desain *Base robot* berfungsi sebagai penopang dari keempat roda supaya roda dapat menggerakkan robot untuk berjalan dan juga sebagai penopang dari aktuator lain. *Base* dari robot ini terbuat dari aluminium.

3.2 Perancangan Sistem Elektronik

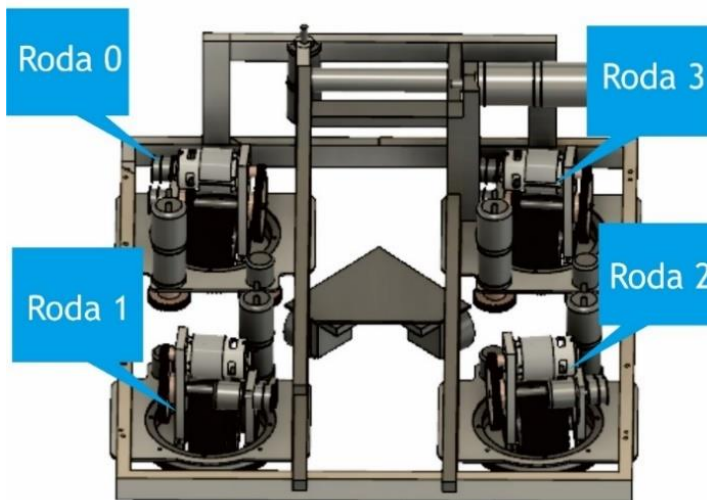
Pada perancangan sistem elektronik terbagi menjadi beberapa bagian yaitu bagian konfigurasi umum, Konfigurasi pin elektronik pada mikro *slave*, dan konfigurasi pin elektronik pada mikro *master*, mikro *slave* berfungsi membaca data *encoder steering* dan mengirimkan data tersebut ke mikro *master*, sedangkan mikro *master* berfungsi sebagai program kendali utama dari robot.

3.2.1 Konfigurasi Sistem Elektronik Robot Secara Keseluruhan

Untuk memudahkan dalam proses *wiring* elektronik dilakukan penomoran pada motor dc dan *encoder*. Penomoran tersebut adalah seperti Tabel 3.1 dan Gambar 3.7.

Tabel 3.1 Penomoran Roda

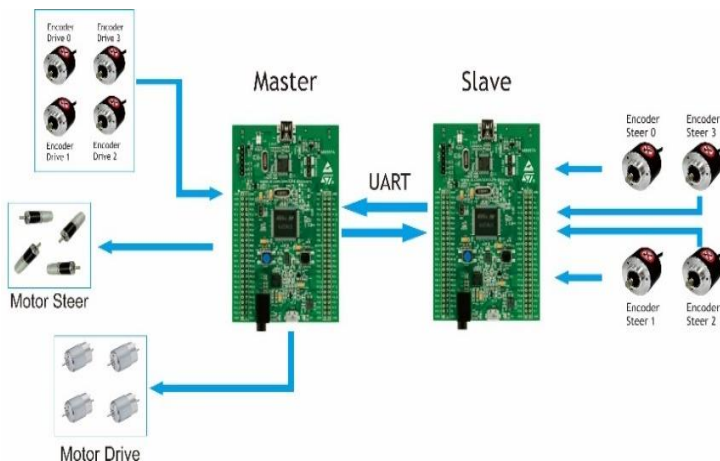
Nomor Roda	Keterangan
0	Roda Kanan Belakang
1	Roda Kanan Depan
2	Roda Kiri Depan
3	Roda Kiri Belakang



Gambar 3.7 Penomoran Roda Robot

Sistem elektronik untuk mengendalikan sistem dari pergerakan robot digunakan dua 2 mikrokontroler STM32 F4 *Discovery* dimana mikrokontroler 1 berfungsi untuk membaca nilai data *encoder* pada roda *steering* dan mengirimkan data tersebut ke mikrokontroler 2. Sedangkan

mikrokontroler 2 berfungsi sebagai membaca data *encoder* motor *drive* dan kendali dari keseluruhan sistem. Kedua mikrokontroler ini berkomunikasi menggunakan UART (*Universal Asynchronous Receiver/Transmitter*). Desain tersebut seperti Gambar 3.8.



Gambar 3.8 Konfigurasi Elektronik

3.2.2 Konfigurasi Pin Elektronik Pada Mikro *Slave*

Pada mikro *slave* dilakukan pembacaan empat data *encoder* yaitu *encoder steer 0*, *encoder steer 1*, *encoder steer 2* dan *encoder steer 3*. untuk melakukan pembacaan nilai *encoder* hanya dapat dilakukan pada pin tertentu saja yaitu pin yang terhubung pada *timer* di STM32F4 *Discovery*. Berikut adalah koneksi pin mikro STM32F4 *Discovery* dengan *rotary encoder*.

Tabel 3.2 Tabel Koneksi Pin STM32 F4 *Discovery*

No	Pin STM	Koneksi Pin
1	PE11	<i>Encoder Steer 0 Chanel A</i>
2	PE9	<i>Encoder Steer 0 Chanel B</i>
3	PB3	<i>Encoder Steer 1 Chanel A</i>
4	PA15	<i>Encoder Steer 1 Chanel B</i>
5	PB4	<i>Encoder Steer 2 Chanel A</i>
6	PB5	<i>Encoder Steer 2 Chanel B</i>

No	Pin STM	Koneksi Pin
7	PB7	<i>Encoder Steer 3 Chanel A</i>
8	PB6	<i>Encoder Steer 3 Chanel B</i>
9	PC12	UART5 Rx Master
10	PD2	UART5 Tx Master
11	PB0	LCD RS
12	PB1	LCD Enable
13	PE7	LCD D4
14	PE8	LCD D5
15	PE10	LCD D6
16	PE12	LCD D7
17	5V	Power Supply 5V
18	GND	Power Supply GND

3.2.3 Konfigurasi Pin Elektronik Pada Mikro Master

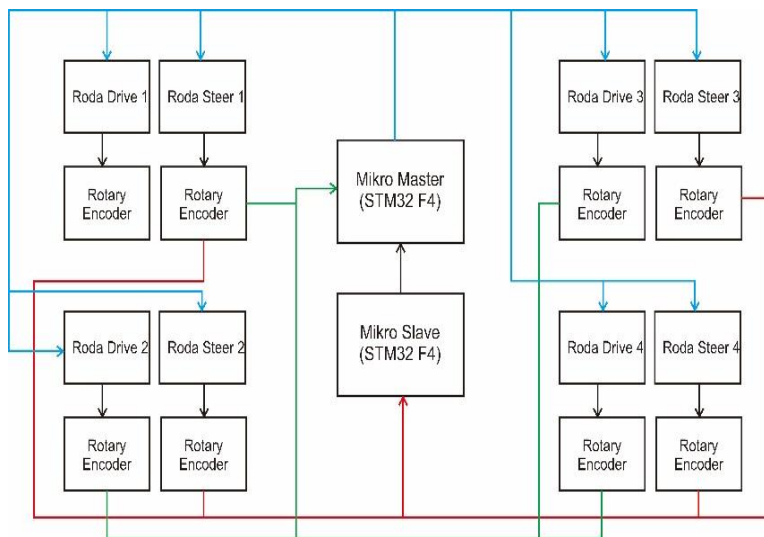
Pada mikro *master* berfungsi mengakses data *encoder drive 0*, *encoder drive 1*, *encoder drive 2*, dan *encoder drive 3*. serta mengendalikan seluruh motor dc yang ada di robot baik itu motor dc *drive* dan motor dc *steer* serta menerima data UART dari mikro *slave*, rincian pin-pin tersebut seperti pada Tabel.3.3.

Tabel 3.3 Koneksi Pin Mikro Master STM32

No	Pin STM	Koneksi Pin
1	PE11	<i>Encoder Drive 0 Chanel A</i>
2	PE9	<i>Encoder Drive 0 Chanel B</i>
3	PB3	<i>Encoder Drive 1 Chanel A</i>
4	PA15	<i>Encoder Drive 1 Chanel B</i>
5	PB4	<i>Encoder Drive 2 Chanel A</i>
6	PB5	<i>Encoder Drive 2 Chanel B</i>
7	PB7	<i>Encoder Drive 3 Chanel A</i>
8	PB6	<i>Encoder Steer 3 Chanel B</i>
9	PC12	UART5 Rx <i>Slave</i>
10	PD2	UART5 Tx <i>Slave</i>
11	PB0	LCD RS
12	PB1	LCD Enable
13	PE7	LCD D4
14	PE8	LCD D5
15	PE10	LCD D6
16	PE12	LCD D7

No	Pin STM	Koneksi Pin
17	5V	<i>Power Supply 5V</i>
18	GND	<i>Power Supply GND</i>
19	PC10	UART Rx Stik
20	PC11	UART Tx Stik

3.3 Perancangan *Software*



Gambar 3.9 Skematik Sistem Roda

Pada sistem seperti Gambar 3.9 tersebut satu set roda berisikan dua motor dc dimana satu motor dc digunakan sebagai roda penggerak robot dan satu motor dc digunakan sebagai roda *steering* pada roda robot. Masing – masing roda memiliki *rotary encoder* yang digunakan sebagai sensor kecepatan dan sensor posisi pada roda robot. *Rotarty encoder* dihubungkan dengan mikrokontroler STM32 F4 *Discovery* sebagai kendali utama robot. Mikrokontroler akan mengendalikan arah putaran dari motor dc *drive* dan juga motor dc *steering* yang dimana masing-masing motor dc tersebut dihubungkan dengan *gear* dengan roda.

Putaran pada masing-masing roda dikonversi oleh masing-masing *rotary encoder* menjadi pulsa dimana pulsa tersebut akan dibaca oleh mikrokontroler STM32 F4 Discovery dan dikonversi menjadi satuan posisi dan kecepatan masing-masing roda. Dimana hasil dari sensor tersebut digunakan sebagai *feedback* dari kontrol posisi dan kecepatan pada masing-masing roda. Sedangkan untuk kontroler digunakan stik PS3 yang akan memberikan nilai arah dan kecepatan robot.

Pada perancangan *software* pada sistem ini terdapat beberapa bagian adalah sebagai berikut.

1. Perancangan program pembacaan data *encoder* dan merubah dalam skala derajat pada *encoder steer*.
2. Perancangan program kendali *steering roda* dan program kendali *drive* roda.
3. Perancangan program kinematika roda.
4. Perancangan program utama.

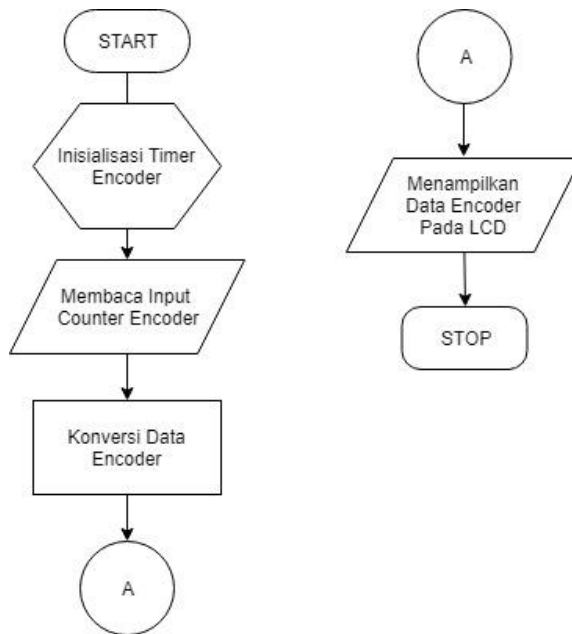
3.3.1 Perancangan Program Data *Encoder* Dan Merubah Derajat Sudut Roda

Pada spesifikasi *encoder* yang digunakan memiliki ppr 200 yaitu 200 pulsa dalam satu putaran. Karena *encoder* yang digunakan memiliki dua *channel output*, maka dapat digunakan persamaan data sebagai berikut untuk mengonversi data ke derajat sudut roda

$$\text{Sudut Roda} = \frac{\text{data } encoder \text{ terbaca} \times 360}{\text{ppr encoder} \times 4 \times \text{Rasio gear roda terhadap encoder}} \dots\dots(3.1)$$

$$\text{Sudut Roda} = \frac{\text{data } encoder \text{ terbaca} \times 360}{2640} \dots\dots\dots(3.2)$$

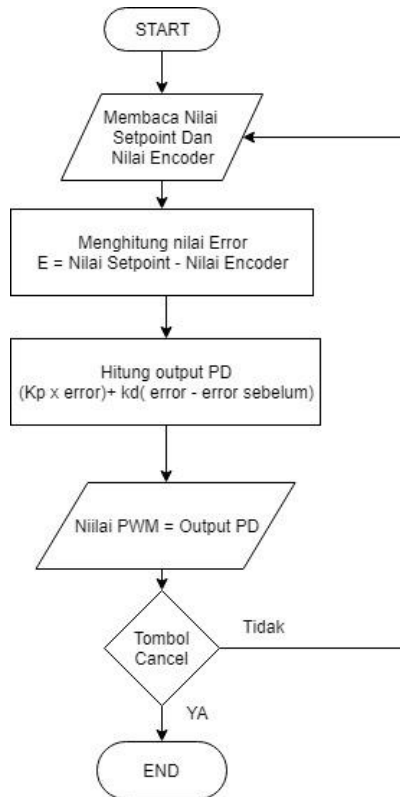
Setelah didapat persamaan untuk merubah data *encoder* menjadi sudut roda maka digunakan algoritma seperti pada Gambar 3.10.



Gambar 3.10 Konversi Data *Encoder*

3.3.2 Perancangan Program Kendali Roda

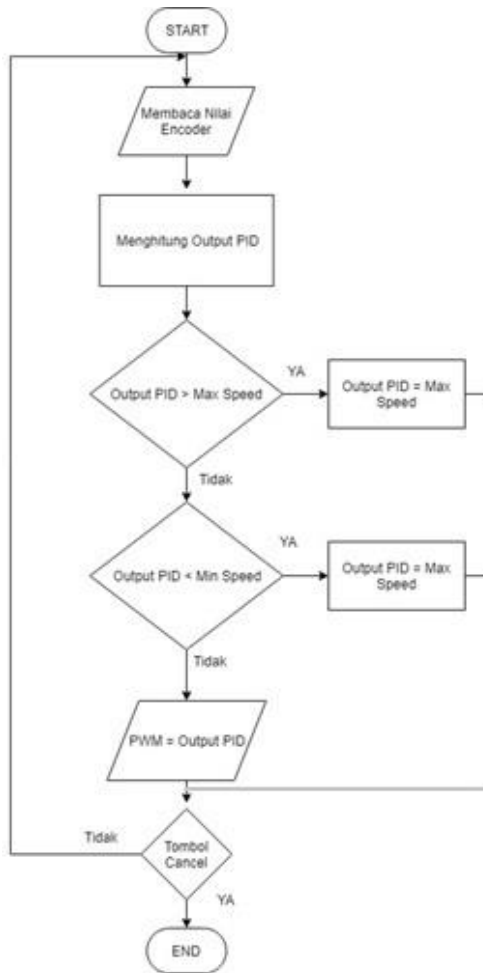
Untuk mengendalikan arah dari sudut roda dilakukan dengan cara membuat program kontrol posisi pada *steering* roda dan digunakan metode kontrol PD (*Proportional-Derivative*). Metode tersebut dipilih dikarenakan memiliki respon yang cepat terhadap perubahan dari *input* kontrol dan untuk menentukan nilai dari k_p , dan k_d dilakukan metode *trial error*. Dimana algoritma dari program tersebut ditunjukkan seperti pada Gambar 3.11.



Gambar 3.11 Kendali *Steering* Roda

Nilai *output* dari kontrol PD akan dimasukkan pada nilai PWM motor dc *steering* untuk menggerakakan *steering* pada posisi yang diinginkan.

Pada bagian kontrol kecepatan roda *drive* digunakan metode kontrol PID (*Proportional – Integral – Derivative*). Metode tersebut dipilih dikarekan sistem membutuhkan kontrol *continue* yang memiliki *output* yang stabil. Untuk algoritma dari program tersebut seperti pada Gambar 3.12.

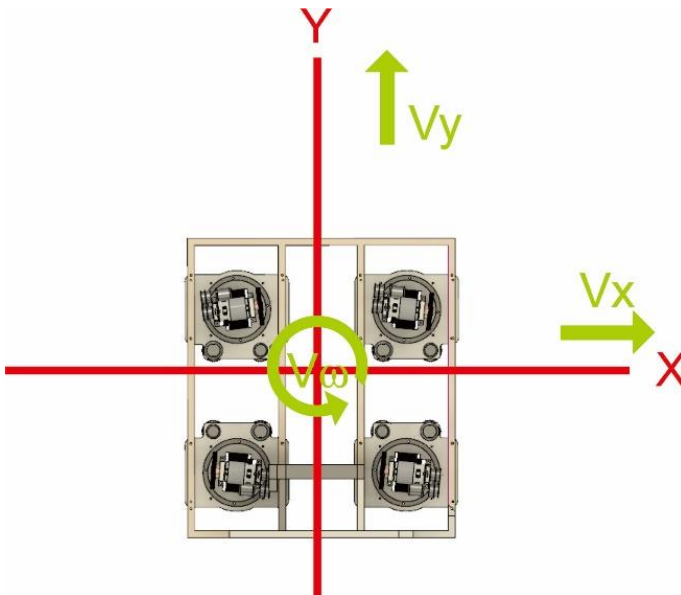


Gambar 3.12 Kendali Pada Motor *Drive*

Nilai *output* dari PID akan dijadikan sebagai nilai PWM untuk menggerakakan motor *drive* yang akan memutar roda sesuai dengan *setpoint* yang telah ditentukan, serta diharapkan nilai putaran roda tetap konsisten dalam kondisi beban yang berbeda-beda.

3.3.3 Perancangan Program Kinematika Robot

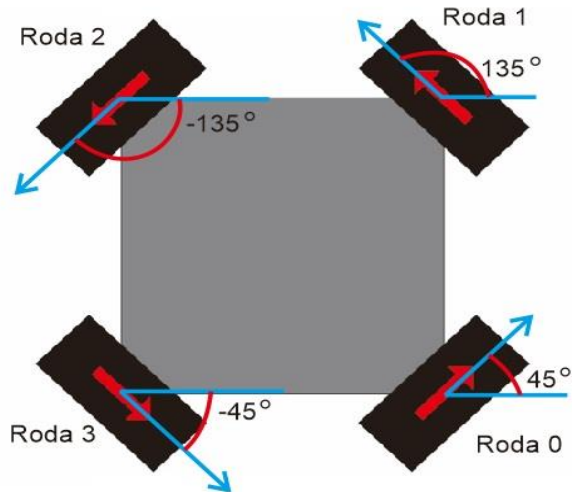
Pada Proyek Akhir ini, kinematika merupakan sebuah cara untuk menggerakan robot. Dalam kasus mekanisme roda *swerve drive* terdapat tiga *input* kecepatan yakni kecepatan x adalah kecepatan gerak robot terhadap sumbu x , kecepatan y adalah kecepatan gerak robot terhadap sumbu y , serta kecepatan ω adalah kecepatan rotasi robot seperti tergambar pada Gambar 3.13.



Gambar 3.13 Arah Vektor Kecepatan Robot

Dari Gambar 3.13 tersebut V_y pada robot ini adalah arah depan dan belakang dimana V_y positif adalah ke arah depan dan V_y negatif adalah arah ke belakang, sedangkan pada V_x positif adalah arah ke kanan dan V_x negatif adalah arah ke kiri, untuk V_ω positif adalah rotasi berlawanan arah jarum jam dan V_ω negatif adalah rotasi berlawanan arah jarum jam.

Untuk mendapatkan nilai sudut dan kecepatan pada roda berdasarkan kecepatan V_x , V_y , dan V_ω maka diperlukan perhitungan kinematik pada masing-masing roda. Nilai tersebut didapatkan dengan cara seperti pada Gambar 3.14.



Gambar 3.14 Arah Roda Saat Robot Berotasi

Dari Gambar 3.14 dapat diketahui sudut masing-masing roda saat robot berotasi yakni tertulis dalam Tabel 3.4.

Tabel 3.4 Sudut Roda Saat Berotasi

No.	Roda	Sudut Saat Rotasi
1	Roda 0	45°
2	Roda 1	135°
3	Roda 2	-135°
4	Roda 3	-45°

Dari nilai-nilai sudut roda saat rotasi tersebut dapat digunakan sebagai acuan dalam perhitungan kinematik robot sebagai berikut.

$$\begin{aligned}
V_x \text{ Roda } 0 &= V_x \text{ Input} + V\omega \times \cos (45) & \dots\dots\dots (3.3) \\
V_x \text{ Roda } 1 &= V_x \text{ Input} + V\omega \times \cos (135) & \dots\dots\dots (3.4) \\
V_x \text{ Roda } 2 &= V_x \text{ Input} + V\omega \times \cos (-135) & \dots\dots\dots (3.5) \\
V_x \text{ Roda } 3 &= V_x \text{ Input} + V\omega \times \cos (-45) & \dots\dots\dots (3.6) \\
V_y \text{ Roda } 0 &= V_y \text{ Input} + V\omega \sin (45) & \dots\dots\dots (3.7) \\
V_y \text{ Roda } 1 &= V_y \text{ Input} + V\omega \sin (135) & \dots\dots\dots (3.8) \\
V_y \text{ Roda } 2 &= V_y \text{ Input} + V\omega \sin (-135) & \dots\dots\dots (3.9) \\
V_y \text{ Roda } 3 &= V_y \text{ Input} + V\omega \sin (-45) & \dots\dots\dots (3.10)
\end{aligned}$$

Setelah didapat nilai V_x dan V_y pada masing-masing roda maka dapat dihitung nilai sudut dan kecepatan pada roda dengan perhitungan sebagai berikut.

$$\text{Sudut Roda } 0 = \tan^{-1} \left(\frac{V_y \text{ Roda } 0}{V_x \text{ Roda } 0} \right) \dots\dots\dots (3.11)$$

$$\text{Kecepatan Roda } 0 = \sqrt{V_x \text{ roda } 0^2 + V_y \text{ Roda } 0^2} \dots\dots\dots (3.12)$$

$$\text{Sudut Roda } 1 = \tan^{-1} \left(\frac{V_y \text{ Roda } 1}{V_x \text{ Roda } 1} \right) \dots\dots\dots (3.13)$$

$$\text{Kecepatan Roda } 1 = \sqrt{V_x \text{ roda } 1^2 + V_y \text{ Roda } 1^2} \dots\dots\dots (3.14)$$

$$\text{Sudut Roda } 2 = \tan^{-1} \left(\frac{V_y \text{ Roda } 2}{V_x \text{ Roda } 2} \right) \dots\dots\dots (3.15)$$

$$\text{Kecepatan Roda } 2 = \sqrt{V_x \text{ roda } 2^2 + V_y \text{ Roda } 2^2} \dots\dots\dots (3.16)$$

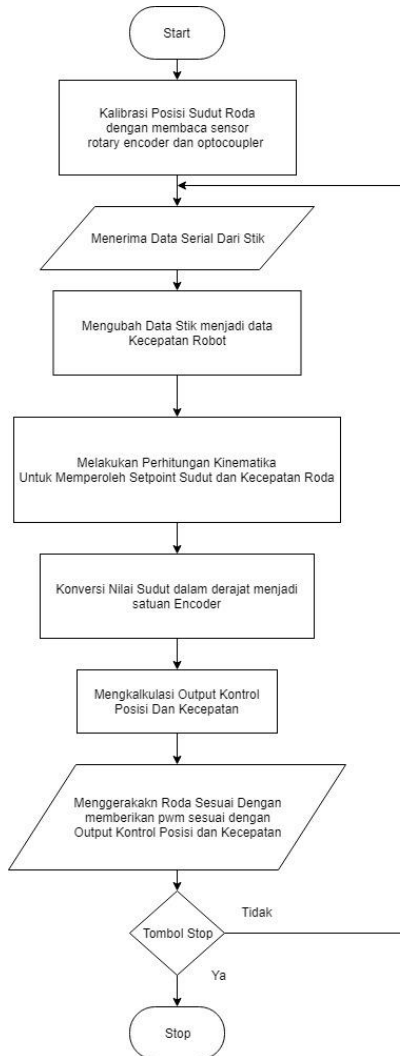
$$\text{Sudut Roda } 3 = \tan^{-1} \left(\frac{V_y \text{ Roda } 3}{V_x \text{ Roda } 3} \right) \dots\dots\dots (3.17)$$

$$\text{Kecepatan Roda } 3 = \sqrt{V_x \text{ roda } 3^2 + V_y \text{ Roda } 3^2} \dots\dots\dots (3.18)$$

Setelah didapat masing-masing nilai sudut dan kecepatan pada roda maka nilai tersebut digunakan sebagai *setpoint* dari kontrol posisi dan kecepatan pada roda.

3.3.4 Perancangan Program Utama Robot

Pada program utama robot dilakukan pembacaan arah gerak robot berdasarkan *input* arah dari stik. Kemudian nilai dari stik akan digunakan sebagai nilai *input* kecepatan V_x , V_y , dan $V\omega$ yang akan dijadikan sebagai *input* kecepatan pada robot.



Gambar 3.15 Flowchart Program Utama

-----Halaman ini sengaja dikosongkan-----

BAB IV

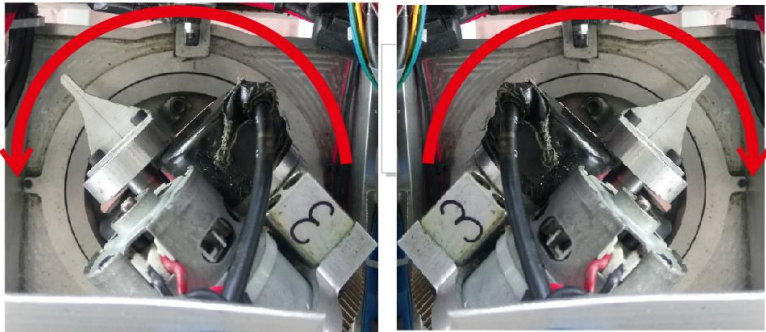
PENGUJIAN DAN ANALISA

Untuk mengetahui apakah tujuan dari pembuatan sistem ini telah tercapai atau tidak, maka perlu diadakannya sebuah pengujian dan analisa terhadap alat yang telah dibuat. Dan sebagai acuan yang tidak terpisahkan adalah adanya proses evaluasi sehingga akan dapat dilakukan langkah-langkah positif guna membawa alat ini kearah yang lebih baik. Adapun pengujian yang dilakukan dengan dua tahap yaitu pengujian pada tiap sub sistem dan secara keseluruhan. Untuk pengujian sub sistem terdiri dari pengujian gerakan motor dc *drive*, gerakan motor dc *steer*. Dan arah dari roda dalam menuju target tujuan posisi tertentu.

4.1 Pengujian Rotary Encoder Pada Steering Roda

Sensor *Rotary Encoder* digunakan untuk mengetahui nilai sudut dari *steering* roda nilai yang ditampilkan berupa *counter* pulsa dimana *counter* tersebut dapat dirubah menjadi besaran derajat. Pengujian pada *rotary encoder* ini dilakukan dengan cara memutar roda pada sudut tertentu dan membaca berapa nilai dari *counter rotary encoder* pada masing-masing sudut tersebut

Pada pengujian ini dilakukan dua kondisi yaitu kondisi diputar searah jarum jam dan kondisi diputar berlawanan arah jarum jam seperti pada Gambar 4.1.



Gambar 4.1 Arah Putar Roda

Dari kegiatan tersebut dapat diketahui hasilnya pada Tabel 4.1 berikut.

Tabel 4.1 Hasil Pengujian Putaran Roda Searah Jarum Jam

No	Nilai Encoder	Keterangan
1.	-334 pulsa	Roda Diputar 45°
2.	-663 pulsa	Roda Diputar 90°
3.	-1011 pulsa	Roda Diputar 135°
4.	-1343 pulsa	Roda Diputar 180°
5.	-1693 pulsa	Roda Diputar 225°
6.	-2016 pulsa	Roda Diputar 270°
7.	-2340 pulsa	Roda Diputar 315°
8.	-2670 pulsa	Roda Diputar 360°

Tabel 4.2 Hasil Pengujian Putaran Roda Berlawanan Arah Jarum Jam

No	Nilai Encoder	Keterangan
1.	338 pulsa	Roda Diputar 45°
2.	663 pulsa	Roda Diputar 90°
3.	1004 pulsa	Roda Diputar 135°
4.	1342 pulsa	Roda Diputar 180°
5.	1683 pulsa	Roda Diputar 225°
6.	2010 pulsa	Roda Diputar 270°
7.	2343 pulsa	Roda Diputar 315°
8.	2665 pulsa	Roda Diputar 360°

Dari hasil pengujian tersebut dapat diketahui bahwa apabila roda diputar searah jarum jam maka nilai dari *rotary encoder* akan semakin kecil sebaliknya apabila roda diputar berlawanan arah jarum jam maka nilai dari *rotary encoder* akan semakin besar. Dari hal tersebut maka dapat diketahui bahwa sudut negatif adalah searah jarum jam, sedangkan sudut positif adalah berlawanan arah jarum jam.

Apabila dihitung berdasarkan perbandingan rasio *gear encoder* dan *gear* roda maka dapat digunakan persamaan sebagai berikut.

$PPR\ Encoder = 200$

Rasio *Gear* = 1:3,3 maka satu putaran roda sama dengan 3,3 putaran dari *rotary encoder* maka dapat dituliskan 3,3.

$$\text{Nilai Encoder} = \frac{\text{Sudut Roda}}{360} \times \text{PPR encoder} \times 4 \times \text{rasio gear} \dots (4.1)$$

$$\text{Nilai Encoder} = \frac{\text{Sudut Roda}}{360} \times 200 \times 4 \times 3,3 \dots (4.2)$$

$$\text{Nilai Encoder} = \frac{\text{Sudut Roda}}{360} \times 2640 \dots (4.3)$$

$$\text{Nilai Encoder} = \frac{\text{Sudut Roda}}{360} \times \text{PPR encoder} \times 4 \times \text{rasio gear} \dots (4.4)$$

$$\text{Nilai Encoder} = \frac{\text{Sudut Roda}}{360} \times 200 \times 4 \times 3,3 \dots (4.5)$$

$$\text{Nilai Encoder} = \frac{\text{Sudut Roda}}{360} \times 2640 \dots (4.6)$$

Dengan persamaan diatas maka dapat dibandingkan data hasil pengukuran langsung dengan *encoder* dengan data hasil perhitungan.

Tabel 4.3 Perbandingan data pengukuran *encoder* roda dengan perhitungan

No	Keterangan	Nilai Encoder (Pengukuran)	Nilai Encoder (Perhitungan)	Margin Error
1	Diputar -45°	-334 pulsa	-330 pulsa	0,15 %
2	Diputar -90°	-663 pulsa	-660 pulsa	0,11 %
3	Diputar -135°	-1011 pulsa	-990 pulsa	0,80%
4	Diputar -180°	-1343 pulsa	-1320 pulsa	0,87%
5	Diputar -225°	-1693 pulsa	-1650 pulsa	1,62%
6	Diputar -270°	-2016 pulsa	-1980 pulsa	1,36%
7	Diputar -315°	-2340 pulsa	-2310 pulsa	1,14%
8	Diputar -360°	-2670 pulsa	-2640 pulsa	1,14%
9	Diputar 45°	338 pulsa	330 pulsa	0,30 %
10	Diputar 90°	663 pulsa	660 pulsa	0,11 %
11	Diputar 135°	1004 pulsa	990 pulsa	0,53%
12	Diputar 180°	1342 pulsa	1320 pulsa	0,83%
13	Diputar 225°	1683 pulsa	1650 pulsa	1,25 %
14	Diputar 270°	2010 pulsa	1980 pulsa	1,14%
15	Diputar 315°	2343 pulsa	2310 pulsa	1,25%
16	Diputar 360°	2665 pulsa	2640 pulsa	0,95%

Dari tabel 4.3 dapat dilihat bahwa nilai kesalahan paling besar adalah 1,63% yaitu pada saat roda diputar -225° nilai rata-rata kesalahannya adalah 0,84%. nilai kesalahan tersebut dikarenakan adanya *backlash* dari *gear encoder* dan *gear* roda.

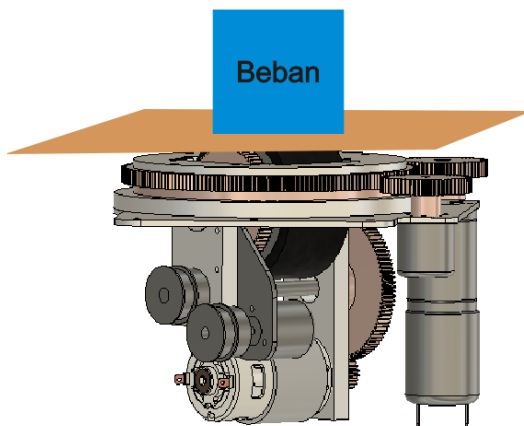
4.2 Pengujian Motor DC Drive

Pada pengujian ini dilakukan dengan cara memberikan pwm dengan *duty cycle* mulai dari 10% sampai 100% pada *driver* motor, cara untuk memberikan nilai *duty cycle* tersebut dengan memberikan nilai pada program seperti pada Gambar 4.2.

<pre><u>duty = 10; // 10% duty cycle</u> pwm = (duty/100)*999; HAL_GPIO_WritePin(portDir1Motor4_GPIO_Port, portDir1Motor4_Pin, GPIO_PIN_SET); HAL_GPIO_WritePin(portDir2Motor4_GPIO_Port, portDir2Motor4_Pin, GPIO_PIN_RESET); TIM12->CCR1 = pwm;</pre>	<pre><u>duty = 30; // 30% duty cycle</u> pwm = (duty/100)*999; HAL_GPIO_WritePin(portDir1Motor4_GPIO_Port, portDir1Motor4_Pin, GPIO_PIN_SET); HAL_GPIO_WritePin(portDir2Motor4_GPIO_Port, portDir2Motor4_Pin, GPIO_PIN_RESET); TIM12->CCR1 = pwm;</pre>
--	--

Gambar 4.2 Program PWM *Duty Cycle* 10% dan 30%

Setelah roda berputar pada pwm *duty cycle* yang telah ditentukan maka roda akan diberikan beban mulai dari 1kg, 3kg, 4kg, 6kg dan 7kg seperti pada ilustrasi Gambar 4.3.



Gambar 4.3 Ilustrasi Pengujian Motor DC Drive

Nilai dari kecepatan roda akan ditampilkan pada lcd seperti pada Gambar 4.4.



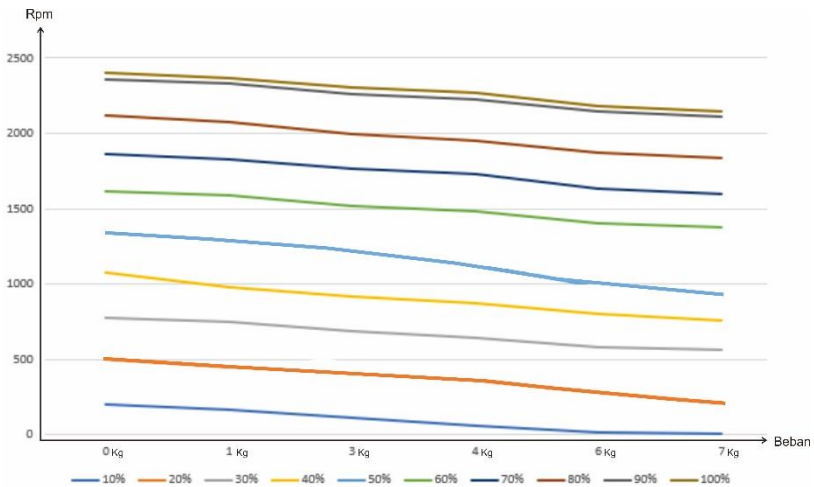
Gambar 4.4 Tampilan Kecepatan Roda *Duty Cycle* 30% Beban 4Kg

Dari pengujian tersebut didapatkan data yang dituliskan pada Tabel 4.4 berikut.

Tabel 4.4 Hasil Pengujian Kecepatan Motor DC

No	Nilai <i>Duty Cycle</i> PWM	Kecepatan					
		Tanpa Beban (RPM)	Beban 1 Kg (RPM)	Beban 3 Kg (RPM)	Beban 4 Kg (RPM)	Beban 6 Kg (RPM)	Beban 7 Kg (RPM)
1	10 %	206	170	111	60	20	5
2	20 %	500	450	405	360	280	210
3	30 %	780	750	690	645	584	568
4	40 %	1075	980	920	873	806	756
5	50 %	1340	1302	1226	1140	982	948
6	60 %	1615	1590	1515	1482	1403	1378
7	70 %	1866	1832	1764	1728	1638	1596
8	80 %	2120	2080	1992	1954	1876	1838
9	90 %	2360	2335	2265	2228	2142	2108
10	100 %	2430	2371	2306	2274	2184	2145

Dari data pengujian diatas nilai kecepatan maksimum dari roda yaitu saat *duty cycle* 100% bernilai pada kecepatan 2400 rpm saat tanpa beban dan 2145 rpm saat diberi beban maksimal 7 Kg. dari data pengujian kecepatan motor dapat dibuat grafik yaitu pada Gambar 4.5.



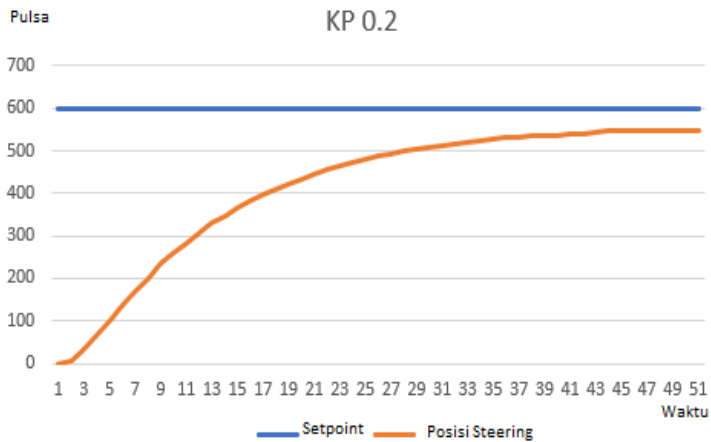
Gambar 4.5 Penurunan Kecepatan Roda Terhadap Beban

Dari grafik pada Gambar 4.5 terlihat bahwa kecepatan dari roda mengalami penurunan terhadap beban yang diberikan dimana semakin berat beban yang diberikan maka semakin banyak juga penurunan kecepatan pada roda. Penurunan kecepatan pada *duty cycle* 10% sampai dengan *duty cycle* 50% terlihat lebih banyak saat beban lebih dari 4 kg dibandingkan pada beban kurang dari 4 kg hal ini dikarenakan pada *duty cycle* 10% sampai dengan 50% tegangan yang diberikan pada motor dc lebih kecil dibandingkan dengan *duty cycle* lebih dari 50%, yaitu kurang dari 50% tegangan maksimal motor dc yakni kurang dari 12 V sehingga mengakibatkan *power* dari motor dc juga kurang dari 50% , sedangkan pada *duty cycle* lebih dari 60 % penurunan kecepatan dari roda mulai dari beban 1 kg sampai dengan 7 kg lebih stabil dikarenakan pada *duty cycle* 60% sampai dengan 100% tegangan dari motor dc lebih dari 14 V atau 14 V sampai dengan 24 V sehingga *power* dari motor dc lebih besar dibandingkan dengan *duty cycle* kurang dari 50%. Pada *duty cycle* 100% kecepatan pada motor dc tidak berbeda jauh dengan *duty cycle* 90% hal ini dikarenakan pada *duty cycle* 90% kecepatan dari motor dc sudah hampir mencapai kecepatan maksimalnya yaitu 2400 rpm maka apabila ditambah *duty cycle* lagi kecepatan dari motor dc tidak akan bertambah secara signifikan lagi seperti pada *duty cycle* 10% sampai dengan *duty cycle* 90 %.

4.3 Pengujian *Steering* Roda

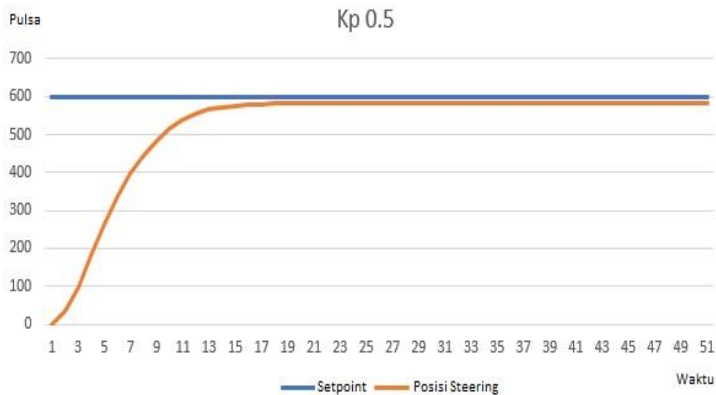
Pada *steering* roda diberikan program kontrol posisi menggunakan kontroler *Proportional*, pada pengujian ini dilakukan dengan memberikan *input setpoint* posisi pada *steering* roda dan merekam data posisi *steering* tiap 20 milisekon dan membuat grafiknya.

Pada pengujian pertama menggunakan K_p 0.2, dari hasil yang didapatkan respon dari *steering* roda lambat dan belum bisa mencapai target *setpoint* seperti yang ditunjukkan oleh Gambar 4.6.



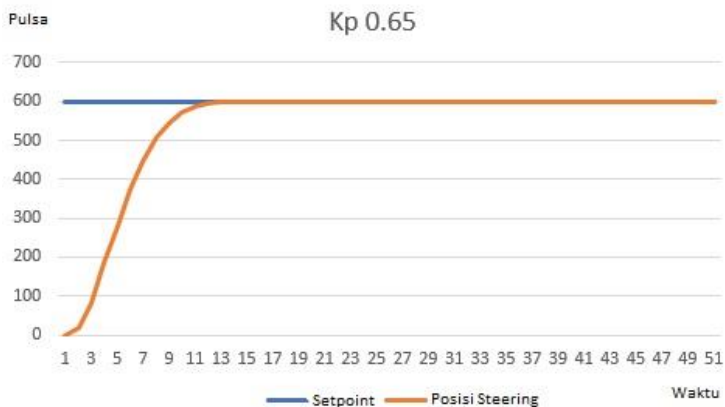
Gambar 4.6 Respon *Steering* K_p 0.2

Pada pengujian kedua menggunakan K_p 0.5, dari hasil yang didapatkan respon dari *steering* roda juga masih belum bisa mencapai nilai *setpoint* seperti yang ditunjukkan oleh Gambar 4.7.



Gambar 4.7 Respon *Steering* K_p 0.5

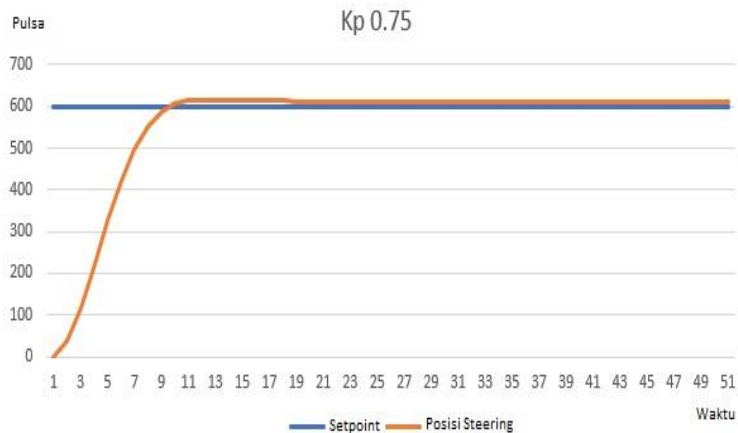
Pada pengujian ketiga menggunakan K_p 0.65, dari hasil yang didapatkan respon dari *steering* roda sudah bisa mencapai nilai *setpoint* pada waktu mencapai 11 milidetik seperti yang ditunjukkan oleh Gambar 4.8.



Gambar 4.8 Respon *Steering* K_p 0.65

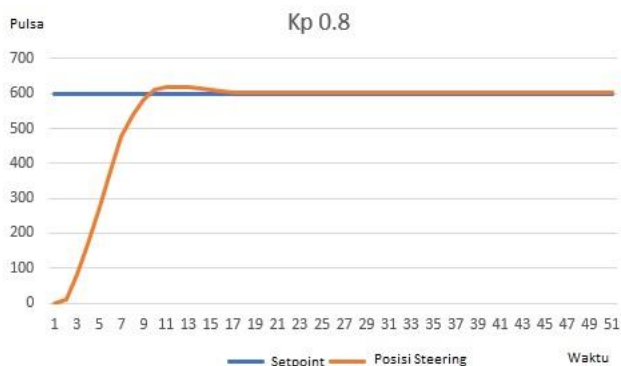
Pada pengujian keempat menggunakan K_p 0.75, dari hasil yang didapatkan respon dari *steering* roda sudah bisa mencapai nilai *setpoint*

pada waktu mencapai 9 milidetik namun terdapat sedikit *overshoot* di awal seperti yang ditunjukkan oleh Gambar 4.9.



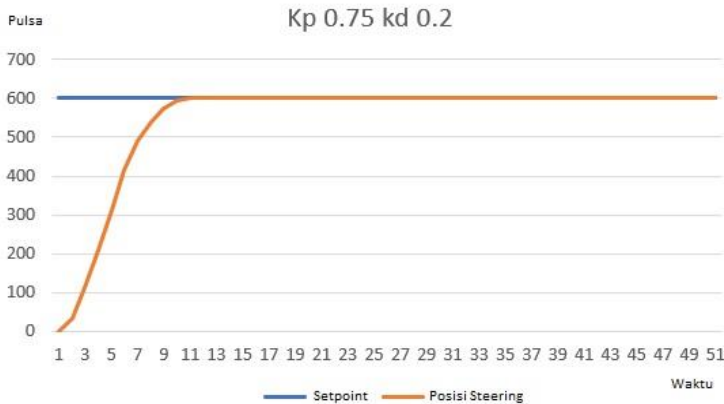
Gambar 4.9 Respon *Steering* Kp 0.75

Pada pengujian kelima menggunakan Kp 0.8, dari hasil yang didapatkan respon dari *steering* roda sudah bisa mencapai nilai *setpoint* pada waktu mencapai 7 milidetik namun terdapat *overshoot* di awal seperti yang ditunjukkan oleh Gambar 4.10.



Gambar 4.10 Respon *Steering* Kp 0.8

Pada pengujian keenam menggunakan K_p 0.75 dan K_d 0.2, dari hasil yang didapatkan respon dari *steering* roda sudah bisa mencapai nilai *setpoint* pada waktu mencapai 9 milidetik seperti yang ditunjukkan oleh Gambar 4.11.



Gambar 4.11 Respon *Steering* K_p 0,75 dan K_d 0,2

Dari pengujian yang dilakukan didapatkan nilai pada K_p 0,2 dan K_p 0,5 sistem belum bisa mencapai *setpoint*, sedangkan pada K_p 0,65 sistem mencapai *setpoint* pada saat waktu mencapai 11 milidetik pada K_p 0,75 dan 0,8 sistem mengalami *overshoot*, sedangkan pada K_p 0,75 dan K_d 0,2 sistem mencapai *setpoint* pada saat waktu mencapai 9 milidetik, oleh karena itu K_p 0,75 dan K_d 0,2 akan dipilih dalam kontroler karena sistem mampu mencapai *setpoint* dengan waktu tercepat dibanding dengan yang lain yaitu pada waktu 9 milidetik.

4.4 Pengujian Kendali Arah Roda

Pengujian ini dilakukan dengan membuat program untuk membaca *input* dari stik kemudian nilai dari stik tersebut dikonversi menjadi *input* kecepatan V_x dan V_y pada robot yang akan digunakan untuk menggerakkan roda sesuai dengan arah yang diberikan oleh stik sehingga arah putaran roda akan sesuai dengan arah yang diberikan oleh stik seperti pada Gambar 4.12.



Gambar 4.12 Pengujian Arah Roda Dengan Perintah Stik Maju

Berdasarkan hasil pengujian roda akan berputar sesuai dengan arah yang diberikan oleh stik dengan nilai sudut dan *encoder* seperti pada Tabel 4.5 – Tabel 4.8 berikut.

Tabel 4.5 Nilai *Encoder* Eoda 0 *Steering* Saat Diberikan Perintah Stik

No	Arah Stik	Sudut Hadap Roda	Nilai Encoder Roda (pulsa)	Nilai Target Encoder (pulsa)	Nilai Error Encoder (pulsa)	Nilai Error Dalam Derajat
1	Maju	90°	663	660	3	0,4
2	Serong Kanan Atas	45 °	322	330	8	1
3	Kanan	0 °	1	0	1	0,1
4	Serong Kanan Bawah	-45 °	-330	-330	0	0
5	Mundur	-90 °	-663	660	3	0,4

No	Arah Stik	Sudut Hadap Roda	Nilai Encoder Roda (pulsa)	Nilai Target Encoder (pulsa)	Nilai <i>Error Encoder</i> (pulsa)	Nilai <i>Error</i> Dalam Derajat
6	Serong Kiri Bawah	45 °	332	330	2	0,2
7	Kiri	0 °	5	0	5	0,7
8	Serong Kiri Atas	45 °	323	330	7	0,9
9	Rotasi Ke Kiri	45 °	330	330	0	0
10	Rotasi Ke Kanan	-45 °	-331	-330	1	0,1

Berdasarkan hasil dari pengujian arah kendali roda terdapat nilai kesalahan paling besar yaitu 8 pulsa *encoder* atau jika dikonversikan dalam satuan derajat sudut roda yaitu 1° dan nilai kesalahan rata-ratanya adalah 0,33°.

Tabel 4.6 Nilai *Encoder* Roda 1 *Steering* Saat Diberikan Perintah Stik

No	Arah Stik	Sudut Hadap Roda	Nilai Encoder Roda (pulsa)	Nilai Target Encoder (pulsa)	Nilai <i>Error Encoder</i> (pulsa)	Nilai <i>Error</i> Dalam Derajat
1	Maju	90	665	660	5	0.7
2	Serong Kanan Atas	45	317	330	13	1.8
3	Kanan	0	3	0	3	0.4
4	Serong Kanan Bawah	-45	-334	-330	4	0.5
5	Mundur	-90	-665	-660	5	0.7
6	Serong Kiri Bawah	45	327	330	3	0.4
7	Kiri	0	2	0	2	0.3
8	Serong Kiri Atas	45	325	330	5	0.7
9	Rotasi Ke Kiri	-45	-331	-330	1	0.1
10	Rotasi Ke Kanan	-45	-330	-330	0	0

Berdasarkan hasil dari pengujian arah kendali roda terdapat nilai kesalahan paling besar yaitu 13 pulsa *encoder* atau jika dikonversikan dalam satuan derajat sudut roda yaitu $1,8^\circ$ dan nilai rata-rata kesalahannya adalah $0,47^\circ$.

Tabel 4.7 Nilai *Encoder* Roda 2 *Steering* Saat Diberikan Perintah Stik

No	Arah Stik	Sudut Hadap Roda	Nilai Encoder Roda (pulsa)	Nilai Target Encoder (pulsa)	Nilai <i>Error Encoder</i> (pulsa)	Nilai <i>Error</i> Dalam Derajat
1	Maju	90°	665	660	5	0,7
2	Serong Kanan Atas	45°	325	330	5	0,7
3	Kanan	0°	1	0	1	0,1
4	Serong Kanan Bawah	-45°	-332	-330	2	0,3
5	Mundur	-90°	-665	-660	5	0,7
6	Serong Kiri Bawah	45°	320	330	10	1,4
7	Kiri	0°	0	0	0	0,0
8	Serong Kiri Atas	45°	325	330	5	0,7
9	Rotasi Ke Kiri	45°	331	330	1	0,1
10	Rotasi Ke Kanan	45°	330	330	0	0

Berdasarkan hasil dari pengujian arah kendali roda terdapat nilai kesalahan paling besar yaitu 10 pulsa *encoder* atau jika dikonversikan dalam satuan derajat sudut roda yaitu $1,4^\circ$ dan nilai rata-rata kesalahannya adalah $0,47^\circ$.

Tabel 4.8 Nilai *Encoder* Roda 3 *Steering* Saat Diberikan Perintah Stik

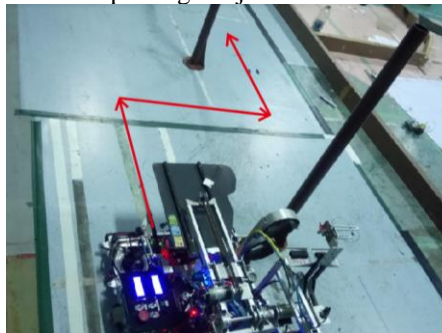
No	Arah Stik	Sudut Hadap Roda	Nilai Encoder Roda (pulsa)	Nilai Target Encoder (pulsa)	Nilai <i>Error Encoder</i> (pulsa)	Nilai <i>Error</i> Dalam Derajat
1	Maju	90°	663	660	3	0,4
2	Serong Kanan Atas	45°	323	330	7	1,0

No	Arah Stik	Sudut Hadap Roda	Nilai Encoder Roda (pulsa)	Nilai Target Encoder (pulsa)	Nilai <i>Error Encoder</i> (pulsa)	Nilai <i>Error</i> Dalam Derajat
3	Kanan	0°	5	0	5	0,7
4	Serong Kanan Bawah	-45°	-330	-330	0	0,0
5	Mundur	-90°	-663	-660	3	0,4
6	Serong Kiri Bawah	45°	321	330	9	1,2
7	Kiri	0°	1	0	1	0,1
8	Serong Kiri Atas	45°	318	330	12	1,6
9	Rotasi Kiri Ke	-45°	-330	-330	0	0,0
10	Rotasi Kanan Ke	45°	331	330	1	0,1

Berdasarkan hasil dari pengujian arah kendali roda terdapat nilai kesalahan paling besar yaitu 12 pulsa *encoder* atau jika dikonversikan dalam satuan derajat sudut roda yaitu 1,6° dan nilai rata-rata kesalahannya adalah 0,55°, sedangkan nilai kesalahan terbesar dari keseluruhan roda adalah 1,8° dengan nilai rata-rata 0,47°.

4.5 Pengujian Pergerakan Robot

Pengujian ini dilakukan dengan cara menggerakkan robot dengan stik ke berbagai arah seperti pada Gambar 4.13 dan mencatat waktu tempuhnya untuk mencapai target tujuan.



Gambar 4.13 Pergerakan Robot

Dari pengujian tersebut didapatkan hasil data yang dicatat pada Tabel 4.9 dan Tabel 4.10 berikut.

Tabel 4.9 Pengujian Gerakan Robot Jarak 120 cm

No	Jenis Gerakan	Waktu Tempuh
1	Belok Kiri 60 cm Maju 60 cm	3,1 detik
2	Belok Kanan 60 cm Maju 60 cm	2.9 detik
3	Mundur 60 cm Belok Kiri 60 cm	2,8 detik
4	Mundur 60 cm Belok Kanan 60 cm	3 detik

Dari pengujian dengan masing-masing jarak total 120 cm tersebut didapatkan waktu paling cepat 2,8 detik dan paling lama 3,1 detik perbedaan tersebut terjadi dikarenakan nilai kesalahan jarak tempuh dan pencatatan waktu pada masing-masing jarak.

Tabel 4.10 Pengujian Gerakan Robot Jarak 3 m

No	Jenis Gerakan	Waktu Tempuh
5	Maju 1 m Belok Kanan 1 m Maju 1 m	7,3 detik
6	Maju 1 m Belok Kiri 1 m Maju 1 m	7,3 detik
7	Mundur 1 m Belok Kiri 1 m Mundur 1m	7,2 detik
8	Mundur 1 m Belok Kanan 1m Mundur 1 m	7,1 detik

Dari pengujian yang dilakukan waktu tempuh terlama untuk mencapai jarak terjauh yakni 3 meter kearah depan dan belok kanan atau kiri dan ke depan lagi atau sebaliknya adalah 7,2 detik sedangkan waktu tercepat untuk jarak tersebut adalah 7, 1 detik, perbedaan tersebut terjadi dikarenakan nilai kesalahan jarak tempuh dan pencatatan waktu pada masing-masing jarak.

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Bab ini berisi tentang kesimpulan-kesimpulan yang didapatkan selama proses pembuatan Proyek Akhir ini beserta saran-saran untuk perbaikan dan pengembangannya.

5.1 Kesimpulan

Berdasarkan pengujian dan analisa perancangan dan implementasi Sistem Kendali Untuk Robot Mobil 4 Roda Menggunakan Mekanisme *Swerve Drive* diperoleh beberapa kesimpulan diantaranya :

1. Mekanisme *swerve drive* dapat dijadikan sebagai penggerak robot mobil secara *omnidirectional* atau bergerak ke segala arah tanpa harus memutar arah hadap robot dengan memiliki nilai kesalahan terbesar 1,8 derajat dan rata-rata kesalahan 0,47 derajat pada sudut dari *steering* roda terhadap semua arah tujuan
2. Penggunaan mekanisme *swerve drive* memerlukan dua mekanisme pada roda yaitu *steering* dan *drive*
3. Pada mekanisme *steering* roda dapat digunakan kontrol posisi untuk menggerakkan arah dari roda, sedangkan pada *drive* roda digunakan kontrol kecepatan untuk memutar roda kearah maju atau mundur

5.2 Saran

Saran –saran yang dapat diberikan untuk implementasi dan pengembangan lebih lanjut dari sistem ini :

1. Dalam pemilihan *rotary encoder* sebagai pembacaan sudut roda dan kecepatan roda gunakan encoder yang memiliki maksimal rpm lebih tinggi dibanding dengan rpm pada roda
2. Penggunaan Mekanisme *Bevel Gear* akan lebih mempermudah dalam proses pengkabelan dc motor, dikarenakan motor dc tidak ikut berputar bersama dengan roda
3. Untuk program kinematika *swerve drive* dapat dibuat lebih umum lagi sehingga lebih mudah apa bila terdapat penggantian mekanik

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

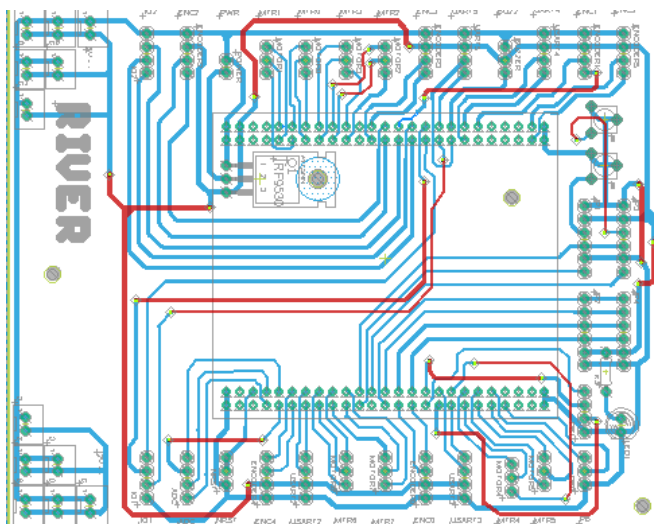
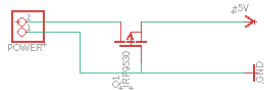
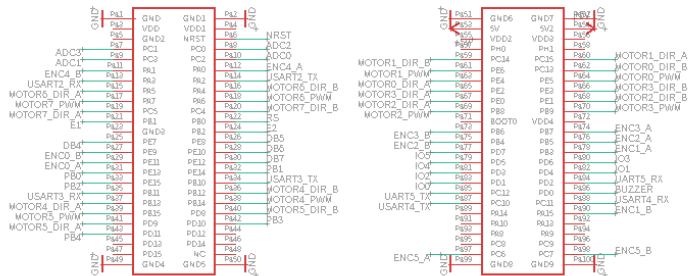
- [1] Siegwart Roland dan Nourbakhsh Illah R, **Introduction Autonomus Mobile Robots**, Cambridge 2004
- [2] Kevin M. Lynch dan Frank C. Park, **Modern Robotic Mechatronic, Planing, and Control**, 2017
- [3], Omron, **Technical Explanation for Rotary Encoders**
- [4] Nyoman Bagia dan I Made Parsa , **Motor-Motor Listrik**, Bandung, 2014
- [5], STMicroelectronics, **UM1472 User manual Discovery kit for STM32F407/417 lines**, 2014.
- [6] Yohan Prakoso, Desain Dan Implementasi Pengukuran Posisi Bola Menggunakan Kamera 360 Derajat Pada Robot Speak Bola, **Tugas Akhir**, Program Studi Teknik Elektro FTI-ITS, 2017
- [7] Nuva Choironi Ersha, Monitoring Program Pengelompokan Dan Penghitug Jumlah Barang Pada Konveyor Menggunakan Mikrokontroler, Tugas Akhir, Departemen Teknik Elektro Otomasi Fakultas Vokasi ITS, Juli 2018.
- [8], STMicroelectronics International N.V., a company, **Atollic Tru studio IDE Overview**

-----Halaman ini sengaja dikosongkan-----

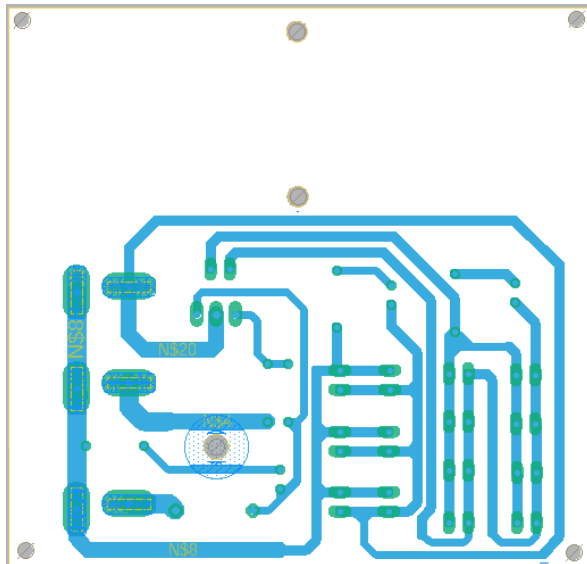
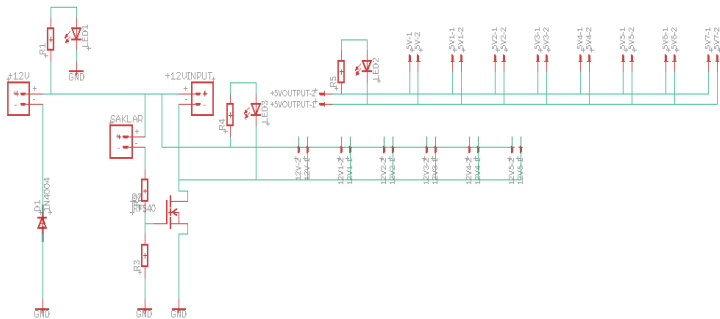
LAMPIRAN A

A. Skematik PCB *Breakout* Mikrokontroler STM32F4 Discovery

STM32F4-DISCOVERY BREAKOUT



B. Desain PCB Power Supply 5V



LAMPIRAN B

A. Listing Program Kontrol Posisi Roda

```
void Motor_Steer_SetPos(short int target_sudut[4], short int  
sudut_roda[4], short int outputmotor[4])  
{  
  short int _target_sudut[4];  
  float proportional_steer[4], integral_steer[4], derivative_steer[4];  
  static float sumError_steer[4], previusError_steer[4];  
  short int error_sudut_steer[4];  
  float kp_steer[4] = { 1.2, 1, 1.2, 1.2};  
  float kd_steer = 0.3;  
  float ki_steer = 0;  
  for (int i = 0; i < 4; i++)  
  {  
    _target_sudut[i] = target_sudut[i];  
    _target_sudut[i] = _target_sudut[i] * 7.41389;  
  }  
  
  for (int i = 0; i < 4; i++)  
  {  
    error_sudut_steer[i] = _target_sudut[i] - sudut_roda[i];  
    sumError_steer[i] += error_sudut_steer[i];  
    if (error_sudut_steer[i] > 1334)  
    error_sudut_steer[i] -= 2669;  
  
    if (error_sudut_steer[i] < -1334)  
    error_sudut_steer[i] += 2669;  
  
    proportional_steer[i] = kp_steer[i] * error_sudut_steer[i];  
    integral_steer[i] = ki_steer * sumError_steer[i];  
    derivative_steer[i] = kd_steer * (error_sudut_steer[i] -  
    previusError_steer[i]);  
  
    previusError_steer[i] = error_sudut_steer[i];  
  
    if (integral_steer[i] > 100)
```

```

    integral_steering[i] = 100;

    if (integral_steering[i] < -100)
        integral_steering[i] = -100;

    outputmotor[i] = (short int) proportional_steering[i] + (short int) integral_steering[i] + (short int) derivative_steering[i];

    if (outputmotor[i] > 349)
        outputmotor[i] = 349;
    if (outputmotor[i] < -349)
        outputmotor[i] = -349;

    }
}

```

B. Listing Program Kontrol Kecepatan Roda

```

void Motor_Drive_SpeedControl(short int output_motor[4],
short int motorSpeedSetPoint[4], short int motorSpeed[4], char
stik_analog)
{
    for (int i = 0; i < 4; i++)
    {
        errorSpeed_motor[i] = motorSpeedSetPoint[i] - motorSpeed[i];

        if (errorSpeed_motor[i] > 999)
            errorSpeed_motor[i] = 999;

        else if (errorSpeed_motor[i] < -999)
            errorSpeed_motor[i] = -999;

        sum_errorSpeed_motor[i] += errorSpeed_motor[i];
        if (integralSpeed_motor[i] > 100)

        integralSpeed_motor[i] = 100;
        else if (integralSpeed_motor[i] < -100)
            integralSpeed_motor[i] = -100;
    }
}

```

```

if (motorSpeedSetPoint[i] == 0)
{
    integralSpeed_motor[i] = 0;
    sum_errorSpeed_motor[i]=0;
}

proportionalSpeed_motor[i]= kpSpeed_motor[i] * errorSpeed_motor[i];
integralSpeed_motor[i]=kiSpeed_motor[i] * sum_errorSpeed_motor[i];
derivativeSpeed_motor[i]=kdSpeed_motor[i]*(motorSpeed[i]-
previous_Speed_motor[i]);
previous_Speed_motor[i] = motorSpeed[i];

output_motor[i]=(short int) (proportionalSpeed_motor[i] +
integralSpeed_motor[i] - derivativeSpeed_motor[i]);

if (output_motor[i] > 999)
    output_motor[i] = 999;
else if (output_motor[i] < -999)
    output_motor[i] = -999;
if (stik_analog)
{
    sum_errorSpeed_motor[i] = 0;
    previous_Speed_motor[i] = 0;
    output_motor[i] = 0;
}
}

motor_drive_SetPWM(output_motor);
}

```

C. Listing Program Kinematika Robot

```

void Motor_Vector_Kine(int kecepatan_x, int kecepatan_y, int
kecepatan_sudut)
{
    int buffer_x[4], buffer_y[4];
    short int swerve_sudut[4], swerve_kecepatan[4];
    static short int error_swerve_sudut[4];

```

```

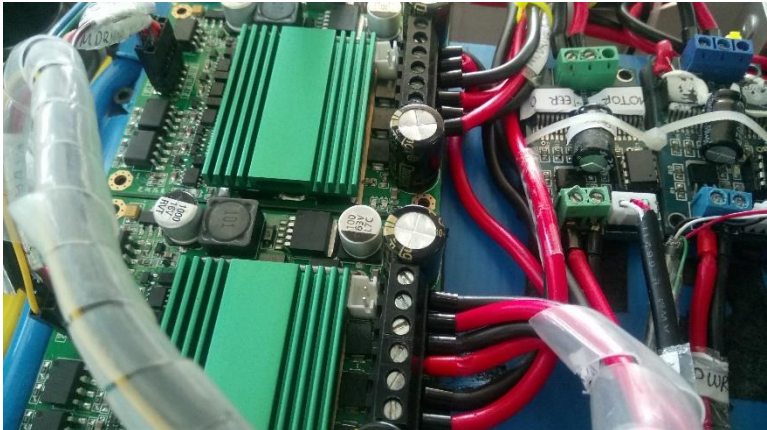
buffer_x[0] = kecepatan_x + kecepatan_sudut * cosf(45 *
0.0174533);
buffer_x[1] = kecepatan_x + kecepatan_sudut * cosf(135 *
0.0174533);
buffer_x[2] = kecepatan_x + kecepatan_sudut * cosf(-135 *
0.0174533);
buffer_x[3] = kecepatan_x + kecepatan_sudut * cosf(-45 *
0.0174533);
buffer_y[0] = kecepatan_y + kecepatan_sudut * sinf(45 *
0.0174533);
buffer_y[1] = kecepatan_y + kecepatan_sudut * sinf(135 *
0.0174533);
buffer_y[2] = kecepatan_y + kecepatan_sudut * sinf(-135 *
0.0174533);
buffer_y[3] = kecepatan_y + kecepatan_sudut * sinf(-45 *
0.0174533);
for (int i = 0; i < 4; i++)
{
    swerve_sudut[i] = atan2f(buffer_y[i], buffer_x[i]) * 57.2958;
    swerve_kecepatan[i] = sqrtof(powf(buffer_x[i], 2) +
powf(buffer_y[i], 2));
    error_swerve_sudut[i] = (short int) (enc_steer_hasil[i] * 0.1348) -
    swerve_sudut[i]; // 0.1348 = konvert enc ke deg

    if (error_swerve_sudut[i] > 180)
    error_swerve_sudut[i] -= 360;
    else if (error_swerve_sudut[i] < -180)
    error_swerve_sudut[i] += 360;
    if (abs(error_swerve_sudut[i]) > 90)
    {
        swerve_sudut[i] -= 180;
        swerve_kecepatan[i] *= -1;
    }
}
for (int i = 0; i < 4; i++)
{
    Motor_Steer_AngleSP[i] = swerve_sudut[i];
    Motor_Drive_SpeedSP[i] = swerve_kecepatan[i];
}}

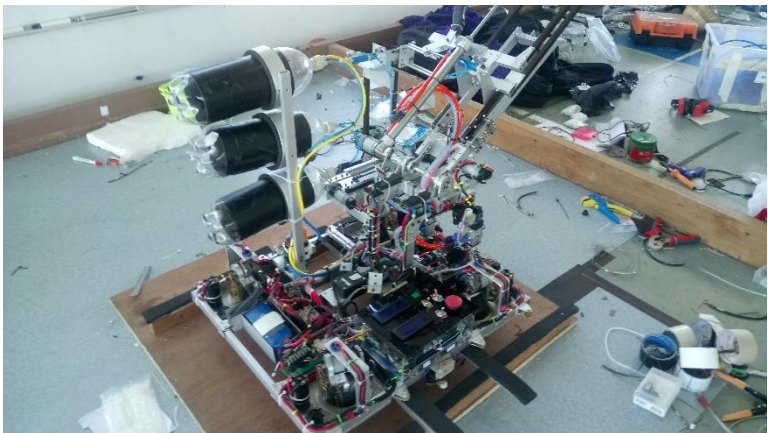
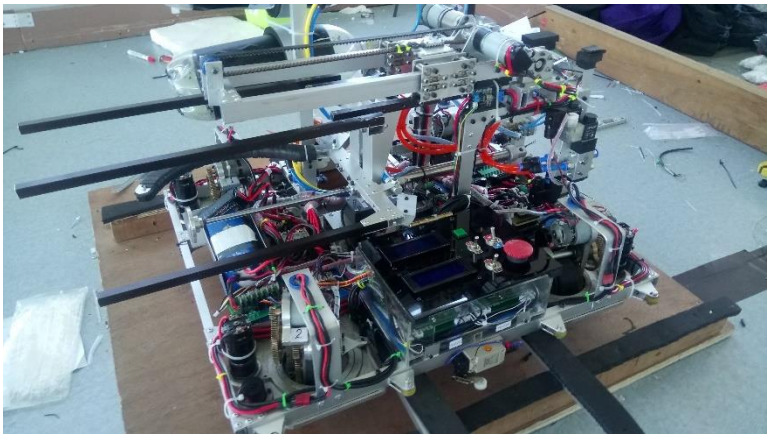
```

LAMPIRAN C

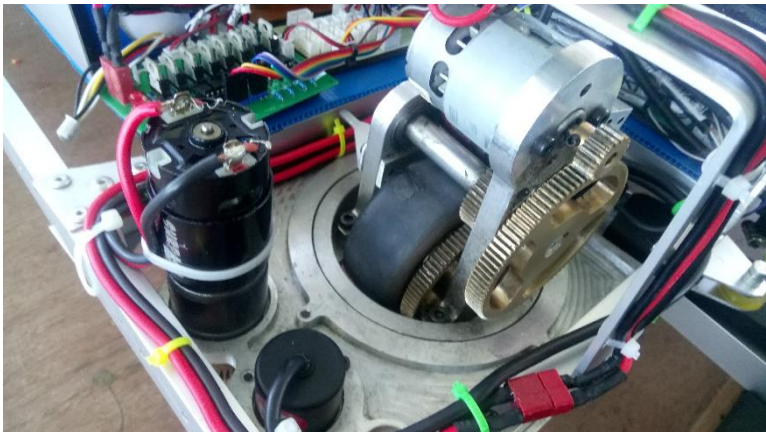
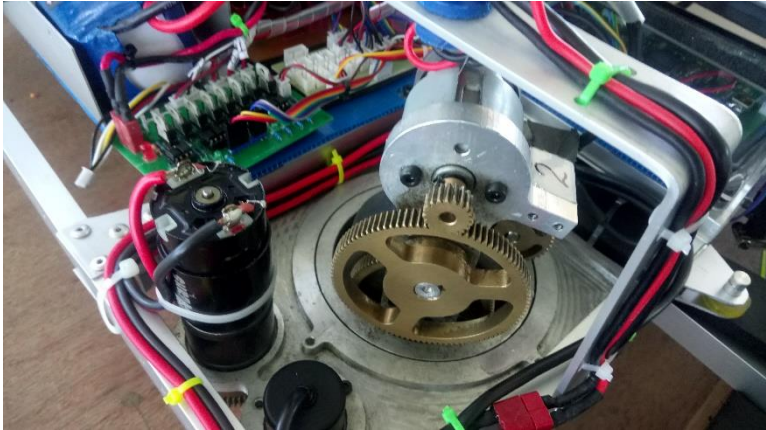
A. Hasil Jadi Sistem Elektronik Robot



B. Bentuk Jadi Robot



C. Bentuk Jadi Roda



-----Halaman ini sengaja dikosongkan-----

DAFTAR RIWAYAT HIDUP



Nama : Moh Ismarintan Zazuli
TTL : Nganjuk, 26 Maret
1998
Jenis Kelamin : Laki – Laki
Agama : Islam
Alamat Rumah : Dsn. Bomo, Ds. Duren
Kec. Sawahan, Kab.
Nganjuk
Telepon/HP : 085608575744
Email : ismarintan98@gmail
.com

RIWAYAT PENDIDIKAN

1. 2001 – 2004 : TK Dharma Wanita Duren
2. 2004 – 2010 : SDN Duren I
3. 2010 – 2013 : SMPN 1 Sawahan
4. 2013 – 2016 : SMAN 2 Nganjuk
5. 2016 – 2019 : Departemen Teknik Elektro
Otomasi Program Studi Elektro
Industri - Fakultas Vokasi (FV)
Institut Teknologi Sepuluh
Nopember (ITS)

PENGALAMAN KERCA

1. Magang Kerja di CV. Panca Gourmet Nusantara

PENGALAMAN ORGANISASI

1. Tim Abu Robocon ITS (RI-GRANT) 2016-2017
2. Tim Abu Robocon ITS (RI-VER) 2017-2018
3. Tim Abu Robocon ITS (RISMA) 2018-2019