KENDALI GERAK PADA ROBOT *FOUR OMNI-DIRECTIONAL* MENGGUNAKAN KONTROLER PID

LAPORAN TUGAS AKHIR

Laporan ini disusun untuk memenuhi salah satu syarat
memperoleh gelar S1 pada Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Sultan Agung Semarang

Disusun Oleh :

MUHAMAD RIYAN FATHUROHMAN

NIM. 30601900030

PROGRAM STUDI TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAS SULTAN AGUNG SEMARANG

2023

MOTION CONTROL OF FOUR OMNI-DIRECTIONAL ROBOT USING PID CONTROLLER

FINAL PROJECT REPORT

Proposed to complete the requirement to obtain a bachelor's degree (S1) at

Department of Electrical Engineering

Faculty of Industrial Technology

Universitas Islam Sultan Agung Semarang

ARANGGED BY :

MUHAMAD RIYAN FATHUROHMAN

NIM. 30601900030

DEPARTMENT OF ELECTRICAL ENGINEERING

FACULTY OF INDUSTRIAL TECHNOLOGY

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2023

LEMBAR PENGESAHAN PEMBIMBING

Laporan tugas akhir dengan judul "**KENDALI GERAK ROBOT *FOUR OMNI-DIRECTIONAL* MENGGUNAKAN KONTROLER PID**" ini disusun oleh

Nama : MUHAMAD RIYAN FATHUROHMAN

NIM : 30601900030

Telah disahkan dan disetujui oleh dosen pembimbing pada :

Hari : Senin

Tanggal : 7 Agustus 2023

Pembimbing I	Pembimbing II
Dr.Bustanul Arifin, ST.,MT.	Agus Suprajutno, ST.,MT.
NIDN.0614117701	NIDN.0602047301

Mengetahui
Ketua Program Studi Teknik elektro

Jenny putri hapsari, S.T., M.T.
NIDN.0607018501

LEMBAR PENGESAHAN PENGUJI

Laporan tugas akhir dengan judul "**KENDALI GERAK ROBOT *FOUR OMNI-DIRECTIONAL* MENGGUNAKAN KONTROLER PID**" ini disusun oleh

Nama : MUHAMAD RIYAN FATHUROHMAN

NIM : 30601900030

Telah disahkan dan disetujui oleh dosen penguji pada :

Hari : Senin

Tanggal : 7 Agustus 2023

Tim Penguji	Tanda
Munaf Ismail,ST.,MT. **NIDN. 0613127302** **Ketua Penguji**	.....
Dr. Muhammad Khosyi'in, ST.,MT. **NIDN. 0625077901** **Penguji I**	.....
Dr. Bustanul Arifin, ST.,MT. **NIDN. 0614117701** **Penguji II**	.....

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama	: Muhamad Riyan Fathurohman
NIM	: 30601900030
Fakultas	: Fakultas Teknologi Industri
Program Studi	: Teknik Elektro

Dengan ini menyatakan bahwa Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) **Teknik Elektro di Fakultas Teknologi UNISSULA Semaran**g dengan judul **"KENDALI GERAK ROBOT FOUR OMNI-DIRECTIONAL MENGGUNAKAN KONTROLER PID",** adalah asli (orisinal) dan bukan menjiplak (plagiat) dan belum pernah diterbitkan/dipublikasikan dimanapun dalam bentuk apapun baik sebagian atau keseluruhan, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab. Apabila di kemudian hari ternyata terbukti bahwa Karya Tugas Akhir tersebut adalah hasil karya orang lain atau pihak lain, maka saya bersedia dikenakan sanksi akademis.

Semarang, 28 Juli 2023

Yang Menyatakan

Muhamad Riyan Fathurohman

NIM. 30601900030

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Muhamad Riyan Fathurohman

NIM : 30601900030

Program Studi : Teknik Elektro

Fakultas : Teknologi industri

Alamat Asal : Desa Simpar Rt 07/ Rw 01 Kec.Bandar, Kab.Batang, Jawa Tengah .

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : **KENDALI GERAK ROBOT FOUR OMNI-DIRECTIONAL MENGGUNAKAN KONTROLER PID** Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiatisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 28 Juli 2023

Yang Menyatakan

Muhamad Riyan Fathurohman

NIM. 30601900030

HALAMAN PERSEMBAHAN

Persembahan :

Pertama,

Allah SWT yang telah memberikan rahmat taufik dan hidayah serta kasih sayangnya, sehingga saya dapat menyelesaikan tugas saya dalam melewati setiap ujian dan cobaan-nya

Kedua,

Ibu Sri Rahayu, Bapak Sugianto dan keluarga saya

Tugas Akhir ini saya persembahkan kepada kedua Orang Tua, keluarga saya yang tak ada henti - hentinya dalam mendoakan dan memberi semangat serta selalu menjadi motivasi saya dalam menyelesaikan studi saya.

Ketiga,

Untuk seluruh Dosen Fakultas Teknologi Industri Prodi Teknik Elektro Universitas Islam Sultan Agung yang selalu memberikan ilmu yang bermanfaat dan motivasi dalam menyelesaikan studi.

HALAMAN MOTTO

“Barang siapa yang mengerjakan amal saleh, maka (pahalanya) untuk dirinya sendiri, dan barang siapa yang berbuat jahat, maka (dosanya) atas dirinya sendiri”

(Surah Al Isra Ayat 7)

“tetaplah menjadi orang baik yang keberadaannya dicari, kehadirannya dinanti, kepergiannya dirindui, kebaikkannya diteladani kelak kematiannya ditangisi dan di do’a i”

(Prof. Fathul Wahid, S.T., M.Sc., Ph.D.)

مَنْ جَدَّ وَجَدَ

“Siapa bersungguh-sungguh kelak ia akan berhasil.”

KATA PENGANTAR

Bismillahirrahmanirrahim

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdullilah puji syukur kehadirat Allah SWT yang telah memberikan Rahmat dan Nikmatnya sehingga masih berkesempatan untuk menuntut ilmu dalam keadaan sehat wal'afiat. Shalawat serta salam semoga selalu tersurahkan kepada baginda Rasullah Muhammad SAW, semoga kelak kita mendapatkan syafaatnya. Amiin Ya Robbalalamin.

Penyusunan Tugas Akhir ini adalah salah satu syarat untuk memperoleh gelar sarjana pada Fakultas Teknologi Industri di Universitas Islam Sultan Agung Semarang. Dalam penulisan Tugas Akhir ini tentunya banyak pihak yang memberikan bantuan secara moril maupun materil. Oleh karena itu penulis menyampaikan ucapan terima kasih yang tiada hingganya kepada :

1. Allah SWT yang telah memberikan rahmat dan ridhonya serta memberikan ketabahan, kesabaran dan kelapangan hati serta pikiran dalam menimba ilmu.

2. Kedua orang tua, yakni bapak Sugianto dan ibu Sri Rahayu yang telah memberikan dukungan baik materil maupun non materil dan tidak pernah berhenti mendo'akan disetiap sholatnya.

3. Bapak Prof. Dr. Gunarto SH., MHum. selaku Rektor Universitas Islam Sultan Agung Semarang.

4. Ibu Dr. Novi Marlyana, ST., MT. sebagai Dekan Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.

5. Ibu Jenny Putri Hapsari, ST., MT. selaku ketua jurusan Prodi Teknik Elektro Fakultas Teknologi Industri Univesitas Islam Sultan Agung Semarang.

6. Bapak Dr. Muhammad Khosyiin, ST., MT. selaku koordinator Tugas Akhir jurusan Teknik Elektro Fakultas Teknologi Industri Univesitas Islam Sultan Agung Semarang.

7. Bapak Dr. Bustanul Arifin ST., MT. dan Bapak Agus Suprajitno ST., MT. selaku dosen pembimbing I dan dosen pembimbing II yang memberikan ilmu yang bermanfaat, memberikan banyak arahan, dan dengan sabar membimbing, sehingga dapat menyelesaikan laporan tugas akhir ini.

8. Seluruh dosen dan karyawan Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Sultan Agung Semarang atas ilmu, bimbingan, dan dukungan dalam penyusunan tugas akhir ini.

9. Kepada teman teman robotic serta elektro 19. yang telah menjadi penyemangat serta pendamping dalam membantu terselesainya tugas akhir ini.

10. Semua pihak yang tidak dapat penulis sebutkan satu per satu atas segala dukungan, semangat, ilmu dalam menyelesaikan tugas akhir ini.

Penulis menyadari bahwa didalam penyusunan laporan ini masih jauh dari kesempurnaan, oleh karena itu kritik dan saran sangat diharapkan untuk mencapai hasil yang lebih baik. Semoga laporan ini dapat memberikan manfaat bagi semua pihak pada terutama Mahasiswa Teknik Elektro Universitas Islam Sultan Agung Semarang dan dapat menambah wawasan.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Semarang, 28 Juli 2023

Penulis

DAFTAR ISI

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR ii
KATA PENGANTAR vii
DAFTAR ISI ix
DAFTAR GAMBAR xi
DAFTAR TABEL xiii
ABSTRAK xiv
ABSTRACT xv
BAB I PENDAHULUAN 1
1.1. Latar Belakang 1
1.2. Perumusan Masalah 2
1.3. Pembatasan Masalah 2
1.4. Tujuan 3
1.5. Manfaat 3
1.6. Sistematika Penulisan 3
BAB II LANDASAN TEORI 5
2.1. Tinjauan Pustaka 5
2.2.1. Penelitian terdahulu 5
2.2.2. Perbandingan penelitian terdahulu dengan penelitian saat ini 5
2.2. Dasar Teori 6
2.2.1. kendali PID (Propotional Integral Derivatif) 6
2.2.2. Metode zigler-nichols 10
2.2.3. Rotary Encoder 12
2.2.4. Roda *Omni-directional* 15
2.2.5. Motor DC 16
2.2.6. Arduino mega 2560 19
2.2.7. Driver motor BTS7960 20
2.2.8. LCD (Liquid Crystal Display) 21
2.2.9. *Universal Battery Elimination Circuit* (UBEC) 23

2.2.10. *Pulse Width Modulation* (PWM) 24
2.2.11. Arduino IDE 25
BAB III METODE PENELITIAN 28
3.1 Perancangan *Hardware* 30
3.1.1 Perancangan mekanik prototipe robot *omni-directional* 30
3.1.2. Perancangan Rangkaian Kontrol menggunakaan Arduino Mega 2560 31
3.1.3 Perancangan input/output pin pada Arduino Mega 2560 33
3.2 Perancangan *Software* 35
3.2.1. perancangan program perhitungan Rpm 36
3.2.2. Perancangan kontrol PID 37
3.2.3 Perancangan pembacaan nilai *rotary encoder* 39
3.2.4 Perancangan program gerak robot 40
3.2.5 Perancangan Komunikasi esp32 sebagai kontrol 41
3.3. Pengujian 42
BAB IV PENGUJIAN DAN ANALISA 44
4.1. Pengujian Putaran Motor untuk Gerak Robot 44
4.2. Kalibrasi Nilai RPM 44
4.3. Pengujian gerak robot dengan kontrol terbuka (*Open loop*) 46
4.4. Tuning PID 51
4.4.1. Menentukan nilai T dan L berdasarkan respon sistem 52
4.4.2 Memasukkan nilai T dan L ke dalam rumus 53
4.4.3. *Fine tuning* 56
4.5. Pengujian kestabilan motor DC dan gerak robot dengan control PID 58
4.5.1. Pengujian gerak robot dengan kontrol PID 59
BAB V PENUTUP 63
5.1. Kesimpulan 63
5.2. Saran 63
DAFTAR PUSTAKA 65

DAFTAR GAMBAR

Gambar 2. 1 Blok Diagram PID 6
Gambar 2. 2 pemberian sinyal step pada sistem 11
Gambar 2. 3 grafik menetukan nilai T dan L 11
Gambar 2. 4 Incremental Optical Encoder dengan LED dan Dua Fototransisitor 13
Gambar 2. 5 Roda Omni-directional 16
Gambar 2. 6 Motor DC PG45 17
Gambar 2. 7 Rangkaian sederhana motor DC magnet permanen 17
Gambar 2. 8 Diagaram blok fungsi alih motor DC 18
Gambar 2. 9 Board Arduino Mega 2560 20
Gambar 2. 10 Driver Motor BTS7960 21
Gambar 2. 11 LCD (Liquid Crystal Display) 22
Gambar 2. 12 Universal Battery Elimination Circuit (UBEC) 23
Gambar 2. 13 Pulse Width Modulation (PWM) 24
Gambar 2. 14 Logo Arduino IDE 26
Gambar 2. 15 Interface Arduino IDE 27
Gambar 3. 1 Diagram alir metodologi penelitian 29
Gambar 3. 2 Desain frame base robot four omni-directional 30
Gambar 3. 3 motor PG45 31
Gambar 3. 4 Blok diagram sistem control 31
Gambar 3. 5 Diagram rangkaian sistem control 33
Gambar 3. 6 Diagram alir perancangan software 35
Gambar 3. 7 Diagram alir perancangan kontrol PID 38
Gambar 4. 1 Kalibrasi RPM menggunakan tachometer 45
Gambar 4. 2 Kalibrasi RPM menggunakan perhitungan Arduino 46
Gambar 4. 3 Titik awal pengujian gerak robot open loop (titik A) 47
Gambar 4. 4 Titik B pengujian gerak robot dan besar nilai erornya 47
Gambar 4. 5 Pengujian gerak robot dari titik B ke titik A 49
Gambar 4. 6 Eror pergerakan robot dengan kontrol open loop 50
Gambar 4. 7 sinyal step kontrol open loop 52
Gambar 4. 8 Sekema menentukan nila T dan L dari grafik step 53
Gambar 4. 9 sinyal respon sistem menggunakan parameter hasil tuning pada motor MA 55
Gambar 4. 10 sinyal respon sistem menggunakan parameter hasil tuning pada motor MB 55
Gambar 4. 11 sinyal respon sistem menggunakan parameter hasil tuning pada motor MC 55
Gambar 4. 12 sinyal respon sistem menggunakan parameter hasil tuning pada motor MD 56

Gambar 4. 13 sinyal respon sistem setelah penyesuaian parameter hasil tuning pada motor MA 57
Gambar 4. 14 sinyal respon sistem setelah penyesuaian parameter hasil tuning pada motor MB 57
Gambar 4. 15 sinyal respon sistem setelah penyesuaian parameter hasil tuning pada motor MC 58
Gambar 4. 16 sinyal respon sistem setelah penyesuaian parameter hasil tuning pada motor MD 58
Gambar 4. 17 Titik awal pengujian gerak robot 59
Gambar 4. 18 Pengujian pergerakan robot dari titik A ke titik B menggunakan kontrol PID 59
Gambar 4. 19 Pengujian pergerakan robot dari titik B ke titik A menggunakan kontrol PID 60
Gambar 4. 20 Eror pergerakan robot dengan kontrol PID 61

UNIVERSITAS ISLAM SULTAN AGUNG
UNISSULA
جامعة سلطان أجونج الإسلامية

DAFTAR TABEL

Tabel 2. 1 rumus Zigler Nichols method 1 11
Tabel 2. 2 konfigurasi pin BTS7960 21
Tabel 2. 3 Konfigurasi pin LCD 16x2 22
Tabel 3. 1 input/output pin pada Arduino meda 34
Tabel 4. 1 Arah putaran motor untuk gerak robot 44
Tabel 4. 2 Data kalibrasi nilai rpm pada Arduino dan tachometer 45
Tabel 4. 3 Data rpm gerak robot dari titik A ke titik B 48
Tabel 4. 4 Data rpm gerak robot dari titik B ke titik A 49
Tabel 4. 5 Hasil pengujian gerak robot tanpa PID 51
Tabel 4. 6 parameter PI dari hasil Fine tuning 56
Tabel 4. 7 Nilai rpm gerak robot dari titik A ke titik B menggunakan kontrol PID 60
Tabel 4. 8 Nilai rpm gerak robot dari titik B ke titik A menggunakan kontrol PID 61
Tabel 4. 9 Hasil pengujian pergerakan robot dengan kontrol PID 62

UNIVERSITAS ISLAM SULTAN AGUNG
UNISSULA
جامعة سلطان أجونج الإسلامية

ABSTRAK

Pada penelitian ini telah dikembangkan sistem kendali gerak robot four omni-directional sebagai dasar Kontes Robot ABU Indonesia (KRAI). Jenis roda yang digunakan adalah roda omni, Pergerakan robot dengan jenis roda ini umumnya menghasilkan arah gerak yang tidak lurus dan arah hadap yang tidak konsisten melalui pegendalian untai terbuka. Pada penelitian ini, dilakukan sebuah perancangan sistem kendali gerak robot omni-directional dengan empat penggerak menggunakan kontrol PID, dengan mengatur kecepatan di masing-masing penggerak diharapkan robot dapat bergerak stabil sesuai dengan kendali yang diberikan Dari hasil pengujian didapatkan nilai putaran masing-masing roda setara dengan nilai set point yang diberikan pada kendali PID. Dengan parameter PID yang ditentukan melalui tuning dengan metode Zigler-Nichols 1 didapatkan nilai perbandingan eror sebelum dan sesudah PID sebesar 1,5 % untuk arah hadap maju dan 1,7% untuk arah hadap mundur.

Kata Kunci : *four omni-directional*, PID, *Zigler-Nichols 1,*

ABSTRACT

In this research a four omni-directional robot motion control system has been developed as the basis for the Indonesian ABU Robot Contest (KRAI). The type of wheel used is an omni wheel. The movement of the robot with this type of wheel generally results in an inconsistent motion direction and an inconsistent facing direction through open strand control. In this study, a design of an omni-directional robot motion control system was carried out with four drives using PID control, by adjusting the speed at each drive it is hoped that the robot can move stably according to the control given. From the test results, the rotation value of each wheel is obtained. equivalent to the set point value assigned to the PID controller. With the PID parameters determined by tuning using the Ziggler-Nichols 1 method, the error ratio before and after the PID is 1.5% for forward facing and 1,7% for backwards.

Keywords **:** *four omni-directional*, PID, *Zigler-Nichols 1,*

UNIVERSITAS ISLAM SULTAN AGUNG
UNISSULA
جامعة سلطان أجونج الإسلامية

BAB I
PENDAHULUAN

1.1.Latar Belakang

Dengan perkembangan teknologi, perkembangan teknologi robotika juga mengalami kemajuan pesat. Banyak aktivitas dan pekerjaan yang dibantu oleh robot sehingga menjadi lebih mudah, cepat, dan efisien. Terdapat berbagai bentuk dan fungsi robot yang telah diciptakan, baik dengan menggunakan lengan atau roda sebagai penggeraknya. Robot beroda dengan roda standar, gerakan yang dihasilkan robot terbatas dan menjadi kendala pada fungsi yang membutuhkan gerakan dengan arah yang berbeda. Namun, pergerakan robot bisa lebih bervariasi dengan menggunakan *omni-directional wheel*.

Dilengkapi dengan empat roda omni maka robot dapat bergerak lebih dari delapan arah tanpa mengubah hadapnya. Penggunaan empat roda omni membuat kemudi menjadi lebih sulit dibandingkan dengan kemudi roda dua konvensional yang dipasang pada poros tunggal. Selain itu karakteristik yang berbeda pada setiap motor dapat mempengaruhi kecepatan setiap roda dan pada akhirnya mempengaruhi pergerakan robot secara keseluruhan.

Sistem kontrol terbuka adalah sistem di mana output tidak mempengaruhi *input* atau tindakan kontrol. Dalam sistem kontrol terbuka, perintah atau *input* dikirim langsung ke aktuator atau proses tanpa adanya pengukuran atau umpan balik untuk memodifikasi input tersebut. Dengan kata lain, sistem ini tidak memantau atau mengevaluasi outputnya. Dengan menggunakan kontrol terbuka pada masing-masing roda, pergerakan robot menjadi tidak lurus. Oleh karena itu, agar robot dapat bergerak lurus, robot dengan empat roda omni-directional membutuhkan kontrol yang dapat mengatur kecepatan dan arah setiap roda.

Pemilihan sistem yang tidak sesuai terhadap kondisi robot dapat mengakibatkan ketidakakuratan pada pemodelan sistem tersebut. Oleh karena itu, diperlukan metode kendali yang tepat untuk memastikan kinerja yang baik, seperti dengan menggunakan metode PID (Proportional-Integral-Derivative). PID merupakan salah satu contoh metode untuk mengatur pergerakan robot. Metode PID menggabungkan aksi kontrol proporsional, aksi kontrol turunan,

dan aksi kontrol integral. Gabungan dari ketiga aksi ini memiliki keunggulan dibandingkan dengan masing-masing aksi kontrol tersebut. Dengan penggunaan kontrol PID, diharapkan robot dapat bergerak lurus sesuai dengan set point yang diberikan.

Dalam penelitian ini, dilakukan perancangan sistem kendali gerak robot *omni-directional* dengan empat penggerak. Dengan mengatur kecepatan pada masing-masing penggerak, diharapkan robot dapat bergerak secara stabil sesuai dengan kendali yang diberikan.

1.2.Perumusan Masalah

Permasalahan yang akan dibahas dalam Tugas Akhir ini adalah sebagai berikut :

1. Apa penyebab *robot omni-directional* bergerak tidak lurus dan kurang stabil. .
2. Bagaimana cara mengendalikan robot *omni-directional* agar dapat bergerak lurus dan stabil saat dikendalikan.
3. Bagaimana Mendapatkan nilai PID yang optimal dengan menggunakan metode tunning *zigler niclous 1.*

1.3.Pembatasan Masalah

Dalam perancangan penelitian ini telah ditentukan batasan-batasan masalah sebagai berikut:

1. Plant yang akan digunakan adalah robot KRAI milik tim robotik UNISSULA
2. Menggunakan sensor internal *rotary encoder* pada motor PG45 sebagai umpan balik PID.
3. Parameter yang akan digunakan sebagai kontrol PID adalah nilai dari Rpm masing-masing motor penggerak.
4. Robot dikendalikan menggunakan cara manual dengan *joystick* PS3.
5. Gerak robot terbatas maju, mundur, kanan, kiri, serong kanan, dan serong kiri.
6. Metode tuning yang digunakan yaitu metode *Ziegler-Nichols 1*

1.4.Tujuan

Tujuan dari penelitian ini adalah sebagai berikut :

4. Guna merancang hardware untuk kendali gerak pada prototipe robot *omni-directional* .
5. Mampu merancang software yang dapat mensimulasikan control kecepatan dengan metode kontrol PID guna stabilitas gerak robot serta optimalisasi kendali pada robot.
6. Mendapatkan nilai PID yang optimal dengan menggunakan metode tunning *zigler niclous 1.*

1.5.Manfaat

Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Meningkatkan stabilitas gerak robot *omni-directional*.
2. Optimalisasi kendali dari robot

1.6.Sistematika Penulisan

Ada beberapa sistematika penulisan dalam penulisan laporan tugas akhir ini adalah sebagai berikut:

BAB I : PENDAHULUAN

Bab ini berisikan latar belakang, perumusan masalah, pembatasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bab ini membahas rujukan-rujukan dari penelitian sebelumnya yang berkaitan dengan tugas akhir yang dibuat. Selain itu bab ini juga berisikan berbagai teori dan konsep yang mendukung dalam pembutan tugas akhir ini.

BAB III : METODE PENELITIAN

Bab ini menguraikan dan memaparkan mengenai metode penelitian yang digunakan, tempat dan teknik pengumpulan data, rancangan penelitian dan tahapan

untuk dapat merancang kendali gerak robot four omni-directional menggunakan kontroler PID.

BAB IV : PENGUJIAN DAN ANALISA

Bab ini memaparkan mengenai pengujian dan analisa hasil pada Perancangan kendali gerak robot four omni-directional menggunakan kontroler PID.

BAB V : PENUTUP

Bab ini berisikan kesimpulan dan saran. Dari pembahasan dan hasil analisa, maka ditarik beberapa kesimpulan. Selain itu, diberikan juga beberapa saran untuk pengembangan penelitian selanjutnya.

UNIVERSITAS ISLAM SULTAN AGUNG
UNISSULA
جامعة سلطان أجونج الإسلامية

BAB II
LANDASAN TEORI

2.1. Tinjauan Pustaka

2.2.1. Penelitian terdahulu

Pada tugas akhir sebelumnya oleh Aryono Priyambudi[2] telah dirancang kendali kecepatan motor pada robot dengan empat roda *omni-directional* dengan menggunakan metode PID, tujuan dari pengendalian sendiri agar robot dapat bergerak lurus dengan arah hadap yang konsisten.

Pada penelitian oleh Dio Arta Karnadi[3] dengan menggunakan metode PID yang di terapkan pada mobile robot dengan 3 penggerak motor yang disusun membentuk huruf y sebagai sistem kontrol cerdas, dari hasil penelitian yang dilakukan telah didapatkan hasil respon robot yang baik.

Pada penelitian oleh Salsabila Karina Kholiqin [4] dengan menggunakan metode tuning *Ziegler-Nichols* 1 untuk mendapatakn nilai Kp,Ki dan Kd yang optimal ,metode tersebut dipilih karena tidak perlu Menyusun model matematik plant yang rumit untuk mengetahui perilaku plant terebut.

Pada penelitian oleh Raditiya Artha Rochmanto[5] dengan mengimplementasikan kontroler PID pada robot *three omni-directional-directional* dimana kelebihan robot ini mampu bermanuver ke segala arah tanpa perlu merubah arah hadap robot, PID digunakan untuk mengatur kecepatan roda sehingga mendapatkan respon yang cepat ke setpoin ,metode tunning parameter PID sendiri menggunakan metode *hand tunning.*

2.2.2. Perbandingan penelitian terdahulu dengan penelitian saat ini

Dari tugas akhir oleh Aryono Priyambudi [2] tentang kendali kecepatan motor pada robot dengan empat roda *omni-directionaldirectional* dengan menggunakan metode PID, dengan menggunakan manual tuning telah di dapatkan nilai Kp, Ki dan Kd yang

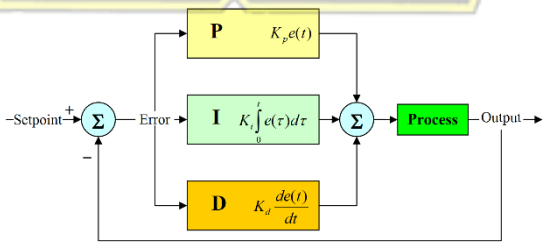
optimal. Dibandingkan dari penelitian tersebut, penelitian saat ini menggunakan metode tuning dengan *Ziegler-Nichols* agar lebih mudah mendapatkan nilai respon yang optimal.

Dari penelitian oleh Raditiya Artha Rochmanto [5] yang mengimplementasikan kontroler PID pada robot *three omni-directional-directional* dimana kelebihan robot ini dapat bergerak ke segala arah tanpa harus mengubah arah hadapnya, penelitian tersebut diimplementasikan pada robot *three omni-directional-directional* sedangkan penelitian saat ini akan di implementasikan pada robot *four omni-directional-directional* dengan perbedaan peletakan penggerak serta bentuk maka berbeda pula nilai respon tiap penggeraknya.

2.2. Dasar Teori

2.2.1. kendali PID (Propotional Integral Derivatif)

Sistem kontrol PID adalah sebuah kontroler yang digunakan untuk mencapai ketepatan dalam sistem instrumentasi dengan menggunakan umpan balik. Kontrol PID terdiri dari tiga metode pengaturan yaitu kontrol Proporsional (P), Derivatif (D), dan Integral (I), yang memiliki keunggulan dan kelemahan masing-masing. Dalam implementasinya, setiap metode dapat bekerja secara independen atau digabungkan bersama-sama[6]. Diagram blok sistem kendali PID ditunjukan pada Gambar 2.1.



Gambar 2. 1 Blok Diagram PID

Dalam merancang sistem kendali PID hal yang sangat penting adalah mencari nilai parameter Kp, Ki atau Kd agar respon sinyal keluaran sistem sesuai dengan plant yang diinginkan.

A. Proporsional

Komponen proporsional menghasilkan sinyal kontrol yang berbanding lurus dengan selisih antara variabel kontrol dan setpoint. Semakin besar selisihnya, semakin besar pula sinyal kontrol yang dihasilkan. Komponen ini bertanggung jawab untuk memberikan respons yang cepat terhadap perubahan setpoint atau gangguan pada sistem.. Dalam kata sederhana, keluaran pengontrol P didapatkan dengan mengalikan konstanta proporsional dengan inputnya. Perubahan pada sinyal masukan akan menghasilkan perubahan yang proporsional sesuai dengan konstanta pengali. Rumus dasar dari pengontrol proporsional adalah:

$$P = K_P e(t) \tag{2.1}$$

Dengan keterangan sebagai berikut :

P = proporsional

Kp = konstanta proporsional

e(t) = eror yang selalu akan berubah

Ada beberapa karakteristik yang perlu diperhatikan saat menerapkan pengontrol proporsional pada suatu sistem. Berikut adalah beberapa ketentuan yang dapat diamati dalam eksperimen:

a) Jika nilai Kp (konstanta proporsional) kecil, pengontrol proporsional hanya mampu melakukan koreksi yang kecil terhadap kesalahan, sehingga respons sistem akan menjadi lambat.
b) Jika nilai Kp ditingkatkan, respons sistem akan semakin cepat mencapai set point dan mencapai keadaan stabil.

c) Namun, jika nilai Kp terlalu besar, sistem dapat menjadi tidak stabil. Peningkatan nilai Kp yang berlebihan dapat menyebabkan kinerja sistem yang tidak diinginkan.

Pengontrol proporsional berfungsi untuk menghasilkan respons sistem dengan kesalahan keadaan stabil yang nol. Namun, jika sistem yang dikendalikan tidak memiliki elemen integral, pengontrol proporsional tidak akan mampu menjaga kesalahan keadaan stabil tetap nol. Dalam hal ini, pengontrol integral diperlukan untuk memperbaiki respons sistem dan mencapai kesalahan keadaan stabil yang nol. Rumus dasar dari pengontrol integral adalah:

$$I = Ki \int_0^t e(t)dt \qquad (2.2)$$

Dengan keterangan sebagai berikut :

I = Integral

Ki = konstanta integral

e(t) = nilai eror

dt = perubahan waktu(second)

B.Integral

Pengontrol integral tidak dapat digunakan secara independen, oleh karena itu perlu digabungkan dengan pengontrol proporsional. Komponen integral mengakumulasi selisih antara variabel kontrol dan setpoint selama waktu. Hal ini membantu mengatasi kesalahan steady-state, yaitu selisih yang tetap ada antara variabel kontrol dan setpoint. Komponen integral memberikan sinyal kontrol yang berbanding lurus dengan integral selisih tersebut, sehingga semakin lama kesalahan steady-state berlangsung, semakin besar pula sinyal kontrol yang dihasilkan.Dengan demikian, pengontrol integral dapat digunakan untuk memperbaiki respons sistem dengan menghilangkan offset, namun perlu memperhatikan nilai Ki agar tidak menyebabkan osilasi yang tidak diinginkan.

C. Derivative

Komponen derivative mengukur laju perubahan variabel kontrol terhadap waktu. Komponen ini memberikan sinyal kontrol yang berbanding terbalik dengan laju perubahan tersebut. Penggunaan komponen derivative membantu dalam memprediksi tren perubahan variabel kontrol, sehingga dapat memberikan respons yang lebih cepat dan mencegah overshoot atau osilasi.Dengan demikian, pengontrol derivatif digunakan untuk merespons perubahan cepat pada sistem. Rumus dasar dari pengontrol derivatif adalah:

$$D = K_d \frac{de(t)}{dt} \quad (2.3)$$

Dengan keterangan sebagai berikut :

D = derifatif

Kd = nilai konstanta derivatif

de(t) = nilai perubahan eror

dt = nilai perubahan waktu(second)

Terdapat berbagai representasi transfer fungsi PID kontroler :

a. Fungsi transfer kontroller PID pada domain s dapat dinyatakan sebagai berikut :

$$U(s) = Kp + \frac{Ki}{s} + Kd\,S \quad (2.4)$$

Dengan Kp,Ki,Kd merupakan parameter proporsional, integral,dan derivative. Persamaan 2.4 juga dapat dituliskan dalam bentuk lain sebagai berikut :

$$U(s) = Kp[1 + \frac{1}{Ti\,s} + Td\,s] \quad (2.5)$$

Dalam bentuk ini dikenakan Ti = Kp/Ki, sebagai konstanta waktu integral, dan Td = Kd/Kp sebagai konstanta derivative.

b. Jika dinyatakan dalam bentuk domain waktu *t,* PID kontroler Sering dituliskan dalam bentuk seperti berikut :

$$U(s) = Kp\,[e + \frac{1}{Ti}\int edt + Td\frac{de}{dt}] \quad (2.6)$$

Dimana :

U(t) = sinyal kontrol

e(t) = sinyal eror yang merupakan perbedaan antara setpoint dan output proses

Kp = gain, Kp = 100% / PB (Proposional Band)

Ti = waktu integral

Td = waktu derivative

Apabila kontrol PID direpresentasikan dalam bentuk persamaan waktu dapat dinyatakan dalam dua bentuk :

a) Bentuk Kecepatan

$$U(t) = Kp\,e(t) + Ki\int_0^t e(t)dt + Kd\frac{de(t)}{dt} \quad (2.7)$$

Jika dinyatakan dalam bentuk persamaan yang berbeda :

$$U(k) = Kp\,e(k) + Ki\big(e(k) + e(k-1)\big) + Kd(e(k) - e(k-1)) \quad (2.8)$$

b) Bentuk Posisi

Dengan menurunkan satu kali persamaan sebelumnya :

$$\frac{dU(t)}{dt} = Kp\frac{de(t)}{dt} + Kie(i) + Kd\frac{d^2e(t)}{dt^2}$$

$$\frac{\Delta U}{Ts} = \frac{Kp}{Ts}e + Ki\,e + \frac{Kd\Delta(\frac{\Delta e}{Ts})}{Ts}$$

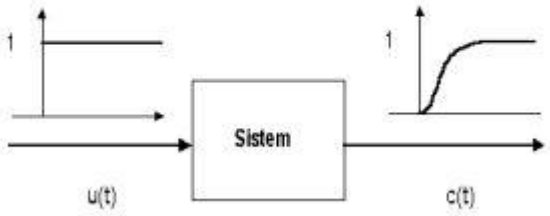
$$\Delta U = Kp\Delta e + Ki.Ts\,e + \frac{Kd\Delta(\Delta e)}{Ts} \quad (2.9)$$

2.2.2. Metode zigler-nichols

Salah satu cara untuk mendapatkan nilai parameter PID yaitu menggunakan metode tuning *Zigler -nichols*. Metode ini terbagi lagi menjadi dua tipe metode yaitu method 1 *open loop* (*step respon*) dan method 2 *close loop* (*osilasi*)

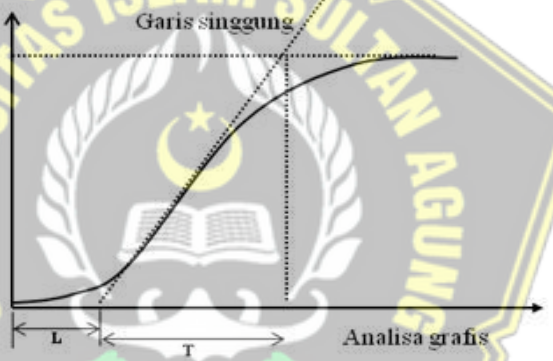
a. Method 1 *open loop (step respon*)

Pada metode ini digunakan sistem *open loop*. Sistem akan diberi input sinyal *step* sehingga dapat di lihat nilai respon sistem saat kontrol *open loop*



Gambar 2. 2 pemberian sinyal step pada sistem

Dari respon *open loop* yang dihasilkan, maka akan didapat nilai parameter parameter ZN1 yaitu L dan T. proses desain menentukan nilai parameter L dan T di tunjukan pada gambar[7].



Gambar 2. 3 grafik menetukan nilai T dan L

Jika nilai L dan T telah ditemukan, maka selanjutnya nilai parameter Kp,Ki,Kd dapat dicari menggunakan rumus-rumus parameter PID untuk metode ZN1[8]. Rumus untuk menentukan parameter PID ZN1 dapat dilihat pada tabel 2.1

Tabel 2. 1 rumus Zigler Nichols method 1

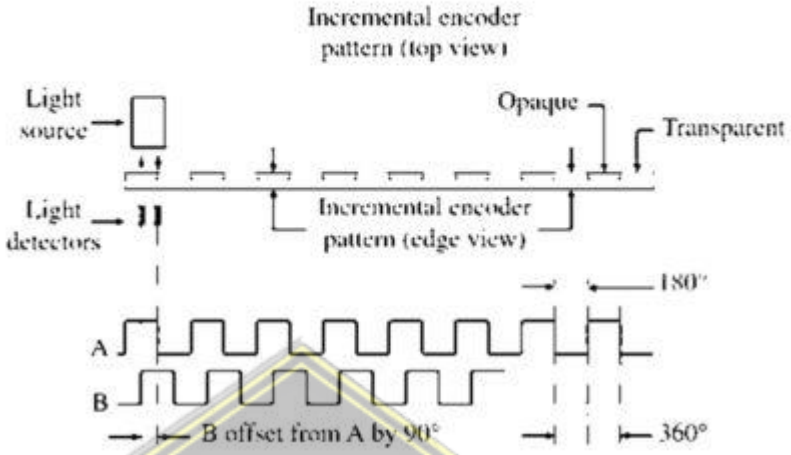
	Kp	Ti	Td
P	T/L	0	0
PI	0,9T/L	L/0,3	0
PID	1,2T/L	2L	0,5L

2.2.3. Rotary Encoder

Encoder merupakan alat yang digunakan untuk menghitung dan mengkomunikasikan perpindahan linier atau perpindahan secara lurus. Alat ini disebut sebagai encoder karena menghasilkan sinyal dalam bentuk data digital, sehingga tidak memerlukan konversi sinyal analog ke digital. Terdapat dua jenis encoder, yaitu encoder inkremental dan absolut. Encoder absolut mencakup berbagai teknik pendeteksian, salah satunya adalah pendeteksian optik.

Rotary encoder, juga dikenal sebagai shaft encoder, adalah prototipe elektromekanik yang digunakan untuk mengubah posisi sudut atau gerakan dari poros menjadi sinyal digital. Rotary encoder juga dapat berfungsi sebagai sensor untuk mendeteksi posisi dan kecepatan dengan mengkonversi perpindahan mekanik menjadi sinyal listrik dan mengubahnya menjadi data.

Encoder optik pada *rotary encoder* menggunakan komponen optoelektronik seperti pengirim/cahaya dan penerima. Optoelektronik adalah cabang ilmu yang berkaitan dengan elektronika yang menggunakan cahaya dan dapat dianggap sebagai bagian dari fotonika. Komponen pengirim umumnya berupa sumber cahaya seperti LED (light emitting diode) yang bisa berjumlah satu atau lebih. Komponen penerima biasanya terdiri dari fototransistor yang bisa berjumlah satu atau lebih. Gambar di bawah ini menunjukkan bagian-bagian *encoder* optik dengan LED dan dua fototransistor.[9]



Gambar 2. 4 Incremental Optical Encoder dengan LED dan Dua Fototransisitor

Dalam penggunaan *rotary encoder* terdapat dua metode untuk menentukan kecepatan putar

a) Pulse-Counting

Metode ini prinsipnya adalah menghitung jumlah pulsa yang terdeteksi oleh sistem pada selang waktu tertentu (tetap), metode ini baik digunakan pada frekuensi tinggi, dimana untuk mengetahui kecepatam sudut atau rad/s dapat menggunakaan persamaan 2.10.

$$\omega = 2\pi f = \frac{2\pi n}{NT_S} \quad (2.10)$$

Dimana :

ω = Angular Velocity (rad/s)

N = Pulse Per Revolution (PPR)

n = Pulses count

Ts = Time Sampling

Sedangkan untuk mengetahui kecepatan dalam bentuk Rpm (*Revolution per minute*) dapat menggunakan persamaan 2.12.

$$f = rps = \frac{jumlah\ putaran}{waktu(detik)} = \frac{n/N}{T_S} = \frac{n}{NT_S} \quad (2.11)$$

$$rpm = 60.rps = \frac{60f}{N} \quad (2.12)$$

Dimana :

rps = revolutiom per secound

N = Pulse Per Revolution (PPR)

n = Pulses count

Ts = Time Sampling

rpm = revolution per minute

b) Pulse-Timing

Metode ini pada prinsipinya adalah dengan mengukur periode pulsa yang terdeteksi oleh sitem atau encoder, metode ini baik digunakan pada frekuensi rendah, berikut merupakan persamaan yang digunakan untuk mencari nilai kecepatan menggunakan mrtode pulse timing, dimana persamaan 2.13 merupakan kecepatan dalam bentuk sudut dan persamaan 2.14 untuk kecepatan dalam bentuk rpm[10].

$$\omega = \frac{2\pi n}{NT_P} = \frac{2\pi}{NT_P} \quad (2.13)$$

$$rpm = \frac{60n}{NT_P} = \frac{60}{NT_P} \quad (2.14)$$

Dimana

ω = Angular Velocity (rad/s)

N = Pulse Per Revolution (PPR)

n = Pulses count

Tp = Time Period

rpm = revolution per minute

2.2.4. Roda *Omni-directional*

Roda Omni adalah jenis roda khusus yang dirancang untuk memungkinkan gerakan multidireksional. Roda ini memiliki rancangan khusus yang memungkinkannya bergerak maju, mundur, dan berbelok dalam sudut yang tajam tanpa perlu memutar atau mengubah arah kendaraan secara keseluruhan.

Roda Omni terdiri dari sejumlah roda kecil yang dipasang disekeliling roda utama. Setiap roda kecil tersebut memiliki sumbu bebas yang memungkinkannya berputar secara independen. Dengan adanya roda kecil ini, roda Omni dapat bergerak dalam berbagai arah dengan mudah.

Keuntungan utama menggunakan roda Omni adalah kemampuan untuk melakukan gerakan lateral yang lancar dan presisi. Hal ini sangat berguna dalam aplikasi seperti robotika, kendaraan otomatis, dan sistem konveyor, dimana manuver yang cepat dan presisi diperlukan.

Roda Omni juga dapat digunakan dalam permainan, seperti robotika kompetitif atau robotika hiburan, dimana kemampuan gerak yang serbaguna sangat penting. Dalam industri, roda Omni sering digunakan dalam sistem pengangkutan material dan perakitan otomatis untuk meningkatkan efisiensi dan fleksibilitas.

Secara keseluruhan, roda Omni adalah jenis roda yang memungkinkan pergerakan multidireksional dengan kemampuan bergerak maju, mundur, dan berbelok dalam sudut yang tajam tanpa perlu mengubah arah kendaraan secara keseluruhan.

Pada roda *omni-directional*, diperlukan kombinasi putaran antara roda untuk menghasilkan gerakan arah yang diinginkan. Sebagai contoh, dapat dilihat pada Gambar 2.5.

Gambar 2. 5 Roda Omni-directional

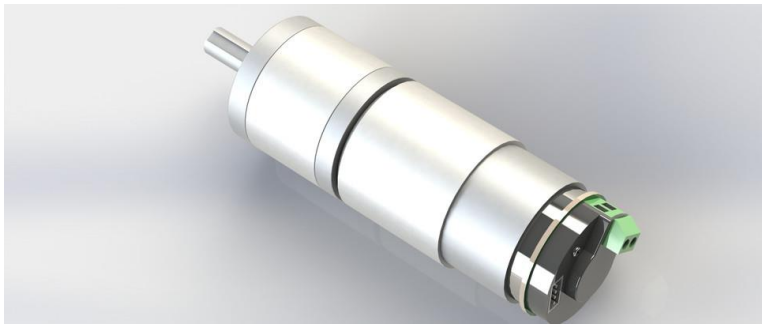
Dari Gambar 2.5 bahwasanya terdapat delapan gerak berbeda yang dapat dihasilkan dari robot dengan empat roda omni. Gerak tersebut dihasilkan dari kombonasi putaran empat buah roda *omni-directional* tersebut yang menghasilkan vektor atau arah.

2.2.5. Motor DC

Motor DC (*Direct Current*) adalah jenis motor listrik yang menggunakan arus searah (DC) sebagai sumber tenaganya. Motor ini bekerja berdasarkan prinsip medan magnet yang berputar untuk menghasilkan gerakan mekanis.

Motor DC terdiri dari dua bagian utama: *rotor* (bagian berputar) dan *stator* (bagian diam). *Rotor* terdiri dari kumparan kawat yang terhubung dengan komutator dan sikat (*brush*). *Stator* terdiri dari medan magnet tetap yang dihasilkan oleh kutub tetap dan kumparan kawat yang disusun dalam pola tertentu.

Saat arus listrik diberikan ke *rotor* melalui sikat dan komutator, medan magnet pada *rotor* berinteraksi dengan medan magnet *stator*, sehingga menyebabkan *rotor* berputar. Arus searah yang mengalir melalui sikat dan komutator berubah arah secara periodik untuk mempertahankan putaran *rotor* yang terus menerus[3].



Gambar 2. 6 Motor DC PG45

Internal rotary yang terdapat pada motor tegolong *rotary* dengan tipe magnetik sehingga memiliki nilai pulsa yang kecil, namun karena *rotary* tersebut terhubung langsung dengan inti motor dan dengan adanya *gearbox* untuk putaran sehingga *rotary* dapat lebih detail dalam melakukan pembacaan putaran untuk mengetahui kecepatan putaran motor tersebut.

Gambar 2. 7 Rangkaian sederhana motor DC magnet permanen

Adapun berikut merupakan keterangan dari Gambar 2.7 :

r_a = Tahanan

e_a = Tegangan

L_a = Induktansi

i_a = Arus

e_b = Gaya Gerak Listrik (emf)

Persamaan kesetimbangan torsi yang diambil oleh motor diberikan oleh persamaan :

$$J\frac{d\omega}{dt}f\omega = K_t i_a \tag{2.15}$$

Dengan *Kt* merupakan konstanta torsi motor.

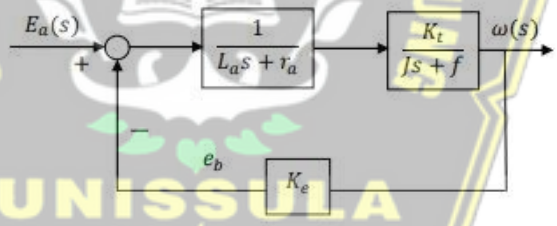
Jika kondisi awal dari persamaan diatas dianggap nol,maka jika dalam bentuk dari kecepatan keluaran motor terhadap tegangan masukan dapat dituliskan :

$$E_a(s) = \frac{r_a}{K_t}[sJ\omega(s) + f\omega(s)] + sL_a\left[\frac{sJ\omega(s) + f\omega(s)}{K_t}\right] + K_t\omega(s)$$

$$E_a(s) = \frac{\omega(s)}{K_t}[r_a sJ + r_a f + s^2 L_a J + sL_a f + K_e K_t]$$

$$\frac{\omega(s)}{E_a(s)} = \frac{K_t}{L_a J s^2 + (L_a f + r_a J)s + r_a f + K_t K_e} \tag{2.16}$$

Dari persaaman diatas maka dapat digambarkan dalam bentuk blok seperti gambar dibawah :



Gambar 2. 8 Diagaram blok fungsi alih motor DC

Nilai induktansi yang berasal dari kumparan jangkar motor $L_a$pada umumnya sangat kecil sekali harganya, maka dari itu dapat diabaikan, sehingga fungsi alih kecepatan kecepatan keluaran motor ω dengan tegangan masukan E_a dapat dinyatakan sebagai :

$$\frac{\omega(s)}{E_a} = \frac{K_m}{s(t_M s + 1)} \tag{2.17}$$

Dengan $K_m$menyatakan konstanta pengaturan motor DC dan t_M adalah konstanta waktu motor DC[12].

2.2.6. Arduino mega 2560

Arduino Mega 2560 adalah salah satu papan pengembangan mikrokontroler yang populer dari keluarga Arduino. Papan ini didasarkan pada mikrokontroler ATMega2560 yang kuat dan serbaguna. Arduino Mega 2560 memiliki fitur yang lebih banyak dibandingkan dengan Arduino Uno, sehingga lebih cocok untuk proyek-proyek yang membutuhkan lebih banyak pin input/output, memori, dan sumber daya pemrosesan.

Berikut adalah beberapa fitur utama dari Arduino Mega 2560:

a. Mikrokontroler ATMega2560: Arduino Mega 2560 menggunakan mikrokontroler ATMega2560 dengan *clock speed* 16 MHz. Mikrokontroler ini memiliki 256 KB memori program (*flash*), 8 KB RAM, dan 4 KB EEPROM untuk penyimpanan data.
b. Pin I/O yang banyak: Arduino Mega 2560 dilengkapi dengan 54 pin digital *input/output*, di antaranya 14 pin dapat digunakan sebagai output PWM (*Pulse Width Modulation*), dan 16 pin analog input. Hal ini memungkinkan koneksi ke banyak sensor, perangkat eksternal, dan aktuator.
c. Komunikasi serial: Arduino Mega 2560 mendukung komunikasi serial melalui beberapa antarmuka, termasuk UART (*Universal Asynchronous Receiver/Transmitter*), SPI (*Serial Peripheral Interface), dan I2C (Inter-Integrated Circuit*). Fitur ini memungkinkan koneksi dengan perangkat lain, seperti sensor, display, atau modul komunikasi.
d. USB Interface: Arduino Mega 2560 memiliki USB interface yang memungkinkan untuk pemrograman dan komunikasi dengan komputer. Anda dapat menggunakan *software* Arduino IDE untuk mengembangkan dan memprogram papan Arduino Mega 2560.
e. Power Supply: Arduino Mega 2560 dapat ditenagai melalui USB atau dengan menggunakan sumber daya eksternal melalui jack power. Papan

ini juga dilengkapi dengan regulator tegangan *onboard* yang dapat mengubah tegangan input menjadi 5V untuk mengoperasikan mikrokontroler dan komponen lainnya.

f. Kompatibilitas: Arduino Mega 2560 kompatibel dengan perangkat lunak dan library Arduino yang luas, sehingga memungkinkan pengembangan proyek dengan mudah dan cepat.

Arduino Mega 2560 digunakan dalam berbagai proyek elektronika dan sistem kontrol. Papan ini cocok untuk proyek-proyek yang membutuhkan pin I/O yang banyak, seperti kendali robot, sistem otomatisasi, sistem monitoring, dan banyak lagi. Dengan fitur-fitur yang lengkap dan kemampuan pemrosesan yang lebih tinggi, Arduino Mega 2560 menjadi pilihan yang populer di kalangan pengembang dan hobi elektronika..[13]

Gambar 2. 9 Board Arduino Mega 2560

2.2.7. Driver motor BTS7960

Driever motor ini mampu menghantarkan arus hingga 43A, dengan memiliki fungsi PWM. Tegangan input DC yang dapat digunakan antara 5.5V-27VDC, sementara itu tegangan input level antara 3.3V-5VDC, driver motor ini menggunakan rangkaian full H-bridge dengan IC BTS7960 dengan perlindungan saat terjadi panas dan arus berlebihan[14]. Gambar driver motor BTS7960 dapat dilihat pada Gambar

Gambar 2. 10 Driver Motor BTS7960

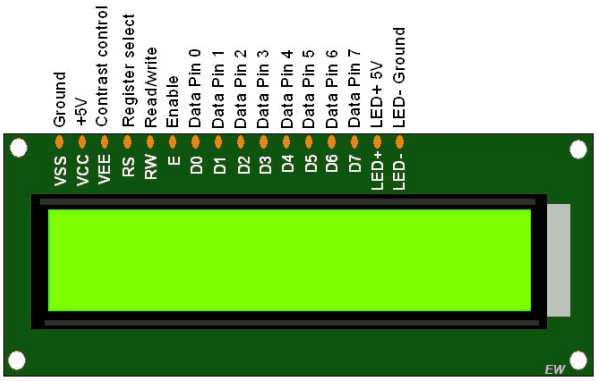
Berikut merupakan tabel fungsi pin kontrol input pada driver motor BTS7960

Tabel 2. 2 konfigurasi pin BTS7960

Pin No	Fungsi	Deskripsi
1	RPWM	Sinyal PWM untuk berputar maju dengan memberikan input high
2	LPWM	Sinyal PWM untuk berputar mundur dengan memberikan input high
3	R_EN	Input high untuk enebel mode putar maju dan input low untuk disable
4	L_EN	Input high untuk enebel mode putar maju dan input low untuk disable
5	R_IS	*Forward Drive, Side current alarm output*
6	L_IS	*Reverse Drive, Side current alarm output*
7	Vcc	*+5V Power Supply microcontroller*
8	Gnd	*Ground Power Supply microcontroller*

2.2.8. LCD (Liquid Crystal Display)

Layar LCD (*Liquid Crystal Display*) adalah jenis layar datar yang menggunakan kristal cair sebagai elemen penyusunnya. Layar LCD terdiri dari sejumlah piksel (titik gambar) yang terdiri dari lapisan-lapisan kristal cair yang dapat dikontrol secara elektronik untuk menghasilkan gambar atau teks.



Gambar 2. 11 LCD (Liquid Crystal Display)

Modul LCD memiliki mikrokontroler yang berfungsi sebagai pengontrol karakter untuk layar LCD. Mikrokontroler LCD dilengkapi dengan memori yang dapat digunakan dan register memori dan pin LCD 16x2.

Tabel 2. 3 Konfigurasi pin LCD 16x2

PIN	Nama	*Fungsi*
1	Vss	*GND*
2	Vcc	*+5V*
3	Vee	*Contrast Voltage*
4	Rs	*Register select* digunakan oleh Arduino untuk memilih lokasi memori saat penulisan data.
5	Rw	*Read/write,* digunakan untuk menentukan mode LCD, mode read atau mode write.
6	E	*Enable*,digunakan untuk mengaktifkan atau menonaktifkan mode penulisan karakter.
7	Db0	*Data bit*
8	Db1	*Data bit*
9	Db2	*Data bit*
10	Db3	*Data bit*
11	Db4	*Data bit*

12	Db5	*Data bit*
13	Db6	*Data bit*
14	Db7	*Data bit*
15	BPL (+)	Panel cahaya *(+)*
16	BPL (-) /GND	*GND*

2.2.9. *Universal Battery Elimination Circuit* (UBEC)

UBEC, juga dikenal sebagai pengatur, berfungsi untuk mengubah tegangan dari tinggi ke rendah atau sebaliknya. Untuk mencapai tingkat efisiensi yang tinggi, diperlukan rangkaian yang cocok. Penurunan tegangan biasanya dilakukan menggunakan IC pengatur seperti 7805 yang umum digunakan. Pengatur ini mampu menangani arus hingga 1A dan membutuhkan tegangan *input* minimal sebesar 7V untuk menghasilkan *output* 5V. Namun, jika tegangan inputnya adalah 9V, maka akan terjadi disipasi daya sebesar ~4Watt, yang merupakan nilai yang cukup besar dan menghasilkan panas. Alternatif lain adalah menggunakan pengatur linier tipe LDO seperti 2940, yang juga mampu menangani arus hingga 1A dan membutuhkan tegangan input minimal sebesar 5.5V untuk menghasilkan output 5V.

Pilihan lainnya adalah menggunakan pengatur *switching*. Jika membutuhkan tegangan 5V - 6V untuk mengoperasikan motor servo atau rangkaian lainnya, UBEC dapat digunakan. UBEC (*Universal Battery Elimination Circuit*) merupakan rangkaian elektronik untuk menurunkan tegangan menjadi 5V atau 6V DC. Tegangan input maksimum UBEC tergantung pada spesifikasinya

Gambar 2. 12 Universal Battery Elimination Circuit (UBEC)

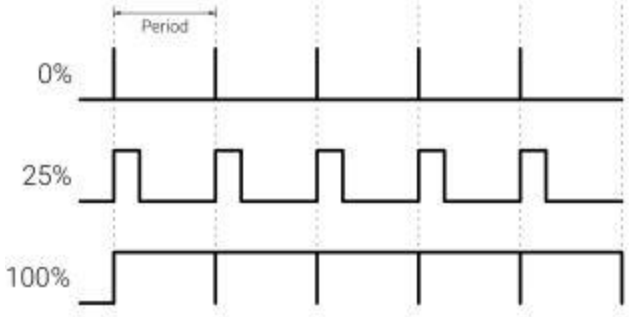
UBEC memiliki kemampuan untuk menjaga tegangan keluaran yang stabil dan bebas dari fluktuasi yang dapat merusak perangkat elektronik. Selain itu, UBEC juga mampu menghasilkan arus keluaran yang cukup tinggi untuk memenuhi kebutuhan daya perangkat yang sedang dioperasikan.

Penggunaan UBEC dapat membantu dalam menyederhanakan sistem daya dan meningkatkan efisiensi energi, karena dapat mengubah tegangan dari berbagai jenis sumber daya menjadi tegangan yang diperlukan oleh perangkat elektronik yang sedang digunakan.

2.2.10. *Pulse Width Modulation* (PWM)

PWM (*Pulse Width Modulation*) adalah sebuah teknik modulasi yang digunakan untuk mengatur lebar pulsa dari sinyal berfrekuensi tetap. Dalam PWM, lebar pulsa sinyal yang dihasilkan dapat diubah secara proporsional terhadap nilai amplitudo sinyal asli yang belum dimodulasi. Teknik ini sering digunakan dalam pengendalian kecepatan motor DC, pengaturan kecerahan lampu LED, dan pengendalian sudut pada motor servo.

Pada PWM, satu siklus pulsa terdiri dari dua kondisi, yaitu tinggi (*high*) dan rendah (*low*). Lebar pulsa (*duty cycle*) PWM dinyatakan sebagai persentase dari waktu siklus total di mana sinyal berada dalam kondisi tinggi. Misalnya, jika *duty cycle* adalah 50%, berarti sinyal tinggi dan rendah memiliki durasi yang sama dalam satu siklus. *Duty cycle* yang lebih besar akan menghasilkan sinyal dengan lebar pulsa yang lebih panjang, sementara *duty cycle* yang lebih kecil akan menghasilkan lebar pulsa yang lebih pendek.



Gambar 2. 13 Pulse Width Modulation (PWM)

Dengan menggunakan PWM, dapat mengatur besaran daya atau kecepatan dengan mengontrol lebar pulsa sinyal yang dihasilkan. Kemampuan ini memungkinkan untuk mengontrol perangkat-perangkat elektronik dengan presisi, seperti mengendalikan kecepatan motor DC secara proporsional atau mengatur kecerahan lampu LED dengan tingkat yang sesuai.

Dalam membandingkan dengan tegangan DC, PWM memiliki tiga metode operasi yang umum digunakan:

1. Mode *Inverted*: Pada mode ini, jika nilai sinyal lebih besar dari titik pembanding (*compare level*), maka output akan diatur tinggi (5V), dan sebaliknya, jika nilai sinyal lebih kecil, maka output akan diatur rendah (0V).
2. Mode *Non-Inverted*: Pada mode ini, *output* akan tinggi (5V) jika titik pembanding lebih besar dari nilai sinyal, dan sebaliknya, *output* akan rendah (0V) jika titik pembanding lebih kecil dari nilai sinyal.
3. Mode *Toggle*: Pada mode ini, *output* akan berubah dari tinggi (5V) *menjadi* rendah (0V) jika titik pembanding sesuai, dan sebaliknya, output akan berubah dari rendah ke tinggi.

Metode operasi PWM yang dipilih tergantung pada kebutuhan aplikasi dan cara pengendalian yang diinginkan.

2.2.11. Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah perangkat lunak yang digunakan untuk memprogram dan mengembangkan perangkat berbasis Arduino. IDE ini menyediakan lingkungan pengembangan yang intuitif dan ramah pengguna untuk menulis, mengedit, mengunggah, dan memantau kode program Arduino.

Arduino IDE menyederhanakan proses pengembangan perangkat keras (*hardware*) dan perangkat lunak (*software*) dengan menyediakan alat yang mudah digunakan untuk pemrograman mikrokontroler Arduino. Dalam Arduino IDE, pengguna dapat menulis kode program dalam bahasa pemrograman Arduino yang berbasis *Wiring*. Bahasa

pemrograman Arduino ini memiliki *sintaks* yang sederhana dan mudah dipahami, cocok untuk pemula yang baru mempelajari pemrograman.

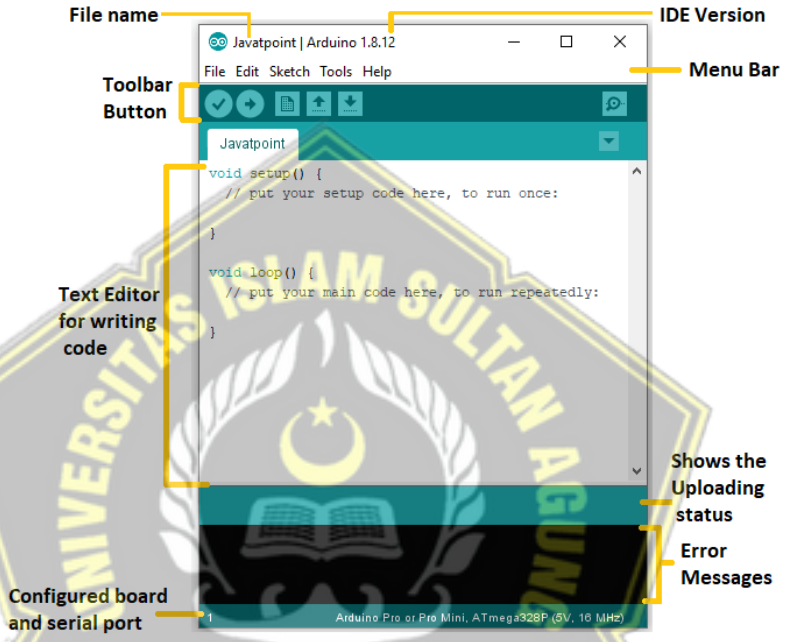
Gambar 2. 14 Logo Arduino IDE

Beberapa fitur utama dari Arduino IDE meliputi:

1. *Text Editor*: Arduino IDE menyediakan editor teks yang digunakan untuk menulis kode program. Editor ini dilengkapi dengan fitur *sintaks highlighting* untuk memudahkan pembacaan dan penulisan kode.
2. *Libraries*: Arduino IDE dilengkapi dengan sejumlah library (perpustakaan) yang telah disediakan oleh komunitas Arduino. Library ini berisi kumpulan fungsi dan perintah yang siap pakai untuk perangkat dan sensor yang umum digunakan. Pengguna dapat dengan mudah mengimpor dan menggunakan library-library ini dalam kode program mereka.
3. *Compiler*: Arduino IDE menyertakan kompiler yang mengonversi kode program yang ditulis dalam bahasa Arduino menjadi bahasa yang dimengerti oleh mikrokontroler Arduino yaitu dalam bentuk Bahasa mesin. Kompiler ini menghasilkan file dengan ekstensi ".hex" yang kemudian dapat diunggah ke papan Arduino.
4. *Serial Monitor*: Arduino IDE memiliki fitur Serial Monitor yang memungkinkan pengguna memantau data yang dikirim melalui koneksi serial antara Arduino dan komputer. Fitur ini sangat berguna

saat mengembangkan kode yang berkomunikasi dengan perangkat eksternal atau saat melakukan debugging.

5. *Tools dan Utilities*: Arduino IDE juga menyediakan berbagai alat dan utilitas lainnya untuk memfasilitasi pengembangan proyek Arduino, seperti pengaturan *board*, pengaturan port serial, pemrograman *bootloader*, dan lain-lain.



Gambar 2. 15 Interface Arduino IDE

Arduino IDE adalah alat yang populer dan banyak digunakan oleh pemula maupun pengguna yang berpengalaman dalam pengembangan proyek dengan papan Arduino. Dengan antarmuka yang sederhana dan dukungan komunitas yang kuat, Arduino IDE mempermudah pengembangan proyek elektronika dan memungkinkan pengguna untuk mengambil keuntungan penuh dari fitur dan kemampuan mikrokontroler Arduino

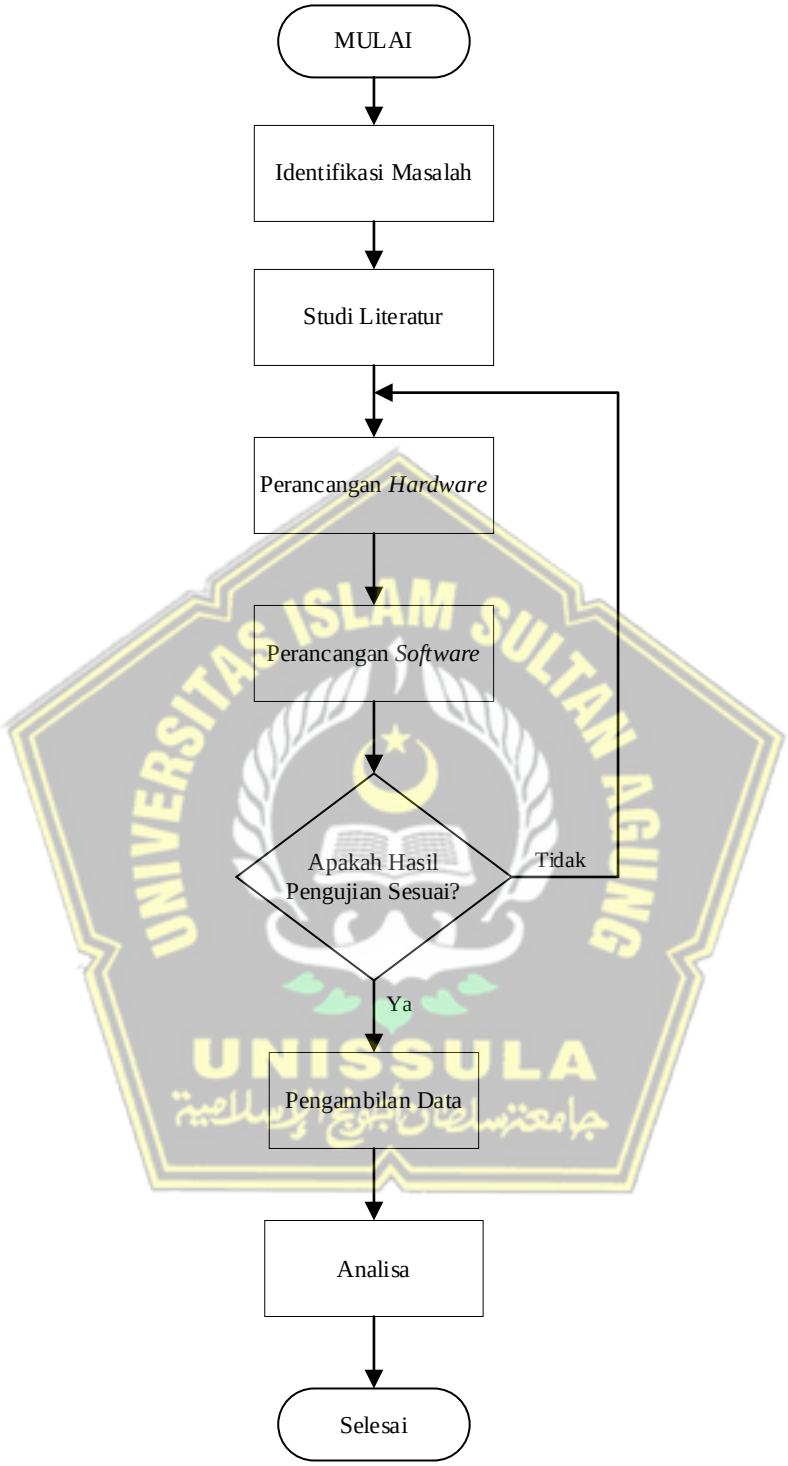
BAB III
METODE PENELITIAN

Pada tugas akhir ini terdapat beberapa tahapan dalam pengerjaannya diantaranya dengan mempelajari buku serta jurnal yang berkaitan dengan Kontrol PID pada robot *omni-directional*. Tahap selanjutnya adalah perancangan *Hardware*, pada tahapan ini terdapat beberapa perancangan yang perlu dilakukan seperti perancangan mekanik prototipe robot *four omni-directiona*l dan juga perancangan rangkaian sistem control dengan menggunakan Arduino mega 2560 sebagai otak dari robot itu sendiri.

Setelah tahap perancangan *Hardware* terpenuhi, tahapan selanjutnya merupakan perancangan *Software* yang meliputi perancangan program pada Arduino IDE seperti program gerak , program konversi dari nilai pulsa *rotary encoder* menjadi Rpm serta program PID dari rumus yang telah di peroleh di ubah dalam bentuk program. Tahap berikutnya merupakan pengujian dari *hardware* maupun *software* bertujuan untuk menegetahui apakah sistem telah sesuai dengan yang diinginkan, jika belum maka akan dilakukan perancangan ulang guna memastikan agar sistem sesuai, jika sudah makan akan dilakukan pengambilan data percobaan meliputi sebagai berikut :

1. Pengujian putaran motor terhadap gerak robot
2. Kalibrasi Rpm bertujuan untuk memastikan bahwasanya perhitungan yang dilakukan oleh mikrokontroler telah sesuai dengan nilai Rpm sebenarnya.
3. Pengujian gerak robot dengan kontrol *open loop* robot digerakkan dari titik A ke titik B untuk melihat seberapa eror geraknya
4. Tuning PID menggunakan metode *Ziegler-Nichols 1*
5. Pengujian kestabilan gerak robot setelah menggunakan kontrol PID

Dari tahapan pengujian maka akan dilakukan Analisa dari data percobaan guna mendapatkan kesimpulan dan hasil dari penilitian .



Gambar 3. 1 Diagram alir metodologi penelitian

3.1 Perancangan *Hardware*

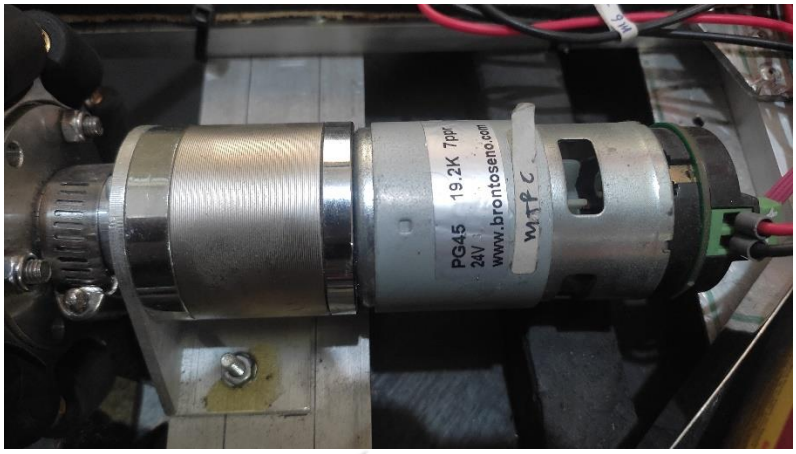
Pada tahap ini akan dibahas tentang pembuatan *Hardware* diantaranya adalah perancangan mekanik prototipe robot *omni-directional* dan perancangan rangkaian control dengan menggunakan Arduino mega 2560.

3.1.1 Perancangan mekanik prototipe robot *omni-directional*

Perancangan mekanik dilakukan degan menggambar pada software Autocad untuk mendapatkan gambaran awal dari bentuk mekanik serta peletakan komponen pendukung seperti peletakan motor DC, kemudian akan dilakukan proses pembuatan robot seperti yang telah Digambar pada software, dalam penelitian ini *frame base* robot dibuat persegi dengan 4 roda omni diletakan menyilang seperti pada Gambar 3.2.

Gambar 3. 2 Desain frame base robot four omni-directional

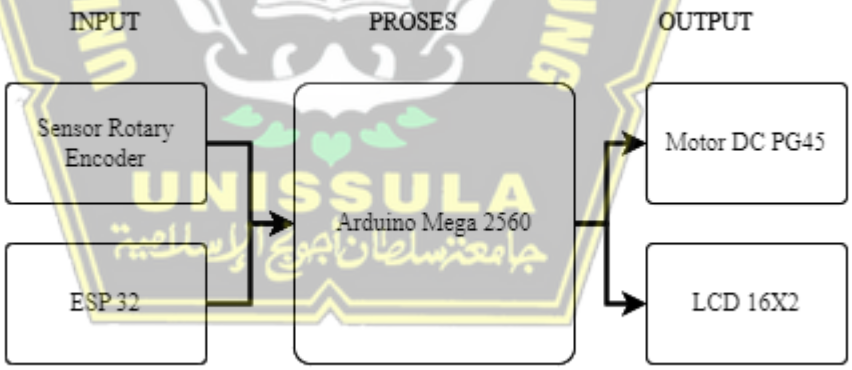
Untuk bahan dari *frame base* menggunakan plat hollow ½ x ½ dengan *alumunium* 3mm sebagai penghubung tiap sambungan dari *frame base*, bahan ini dipilih karena tingkat ketahanan yang kuat dan ringan serta harga yang lebih efisien, untuk penggerak utama dari robot omni sendiri menggunakan motor PG (*Planetary Gear*) 45 mm yang telah terdapat *internal rotary encoder* dengan konfigurasi 7 ppr seperti pada Gambar 3.3.



Gambar 3. 3 motor PG45

3.1.2. Perancangan Rangkaian Kontrol menggunakaan Arduino Mega 2560

Pada penelitian ini dilakukan pembuatan suatu sistem kontrol yang akan mengatur kesetabilan gerak robot *omni-directional*, actuator ataupun output yang digunakan merupakan motor DC PG45 yang di pasang dengan roda omni berdiameter 100mm serta LCD 16X2 untuk menampilkan nilai pembacaan Rpm.



Gambar 3. 4 Blok diagram sistem control

Tiap-tiap bagaian dari blok diagram sistem pada Gambar 3.4. dapat dijelaskan sebagai berikut :

1. Input

a. ESP 32 : digunakan sebagai penerima input kontrol dari *joystick*, dikarenakan ESP32 sudah memiliki modul wifi dan bluethooth guna konektifitas dengan *joystick wireless*

b. Sensor internal rotary : untuk mengambil nilai pulsa tiap putaran encoder yang akan dikonversi mencadi Rpm melalui rumus perhitungan pada Arduino.

2. Output

a. LCD 16x2 : digunakan untuk menampilkan informasi seperti Rpm terbaca dan juga status dari kondisi robot

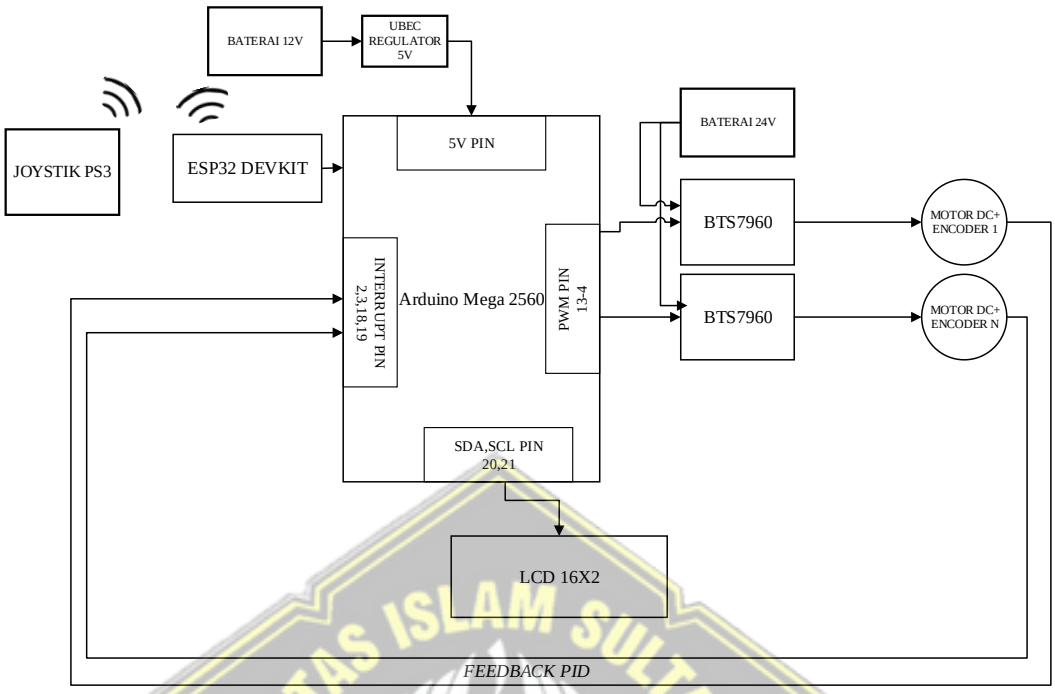
b. Driver dan Motor DC : digunakan sebagai actuator sehingga robot dapat bergerak.

3. Proses

Pada tahap ini pemprosesan input dan output akan dilakukan menggunakan mikrokontroler Arduino mega 2560.

Berdasarkan pada Gambar3.4. dapat dilihat bahwa sistem ini memiliki sumber masukan yaitu berupa empat *internal rotary encoder* yang terdapat pada setiap penggerak motor PG45, masukan ini digunakan sebagai umpan balik guna mengetahui Rpm di setiap penggerak, dimana nilai masukan tersebut diproses pada mikrokontroler Arduino Mega 2560 untuk melakukan perhitungan sesuai nilai Kp,Ki dan Kd yang di tentukan agar nilai respon dapat cepat naik sesuai setpoint yang ditentukan, kemudian informasi akan ditampilkan pada layer LCD agar memudahkan dalam Analisa.

Robot ini dikendalikan menggunakan *Joystik* PS3 yang dihubungkan dengan ESP32 dengan menyamakan MAC (Media Access Control) adresss pada joy dengan ESP32 . Output yang diberikan yaitu berupa kesetabilan gerak pada robot karena keselarasan kecepatan pada tiap penggerak sehingga menciptakan gerak yang lurus sesuai yang dinginkan.



Gambar 3. 5 Diagram rangkaian sistem control

Dari diagram diatas dapat diketahui bahwasannya Robot ini disupply menggunakan tiga baterai lipo 12V dengan konfigurasi satu baterai sebagai supply mikrokontroller, dan dua baterai lainnya di seri untuk menghasilkan tegangan 24V guna supply motorDC. Guna menyesuaikan tegangan mikrokontroler di karenakan Arduino mega 2560 bekerja pada tegangan 5V maka dibutuhkan regulator/ *step down* pada penelitian ini menggunakan UBEC Regulator demi memenuhi keadaan tersebut.

3.1.3 Perancangan input/output pin pada Arduino Mega 2560

Pada ini bertujuan untuk mempermudah dalam insialisasi pin pada saat proses pemrograman, penggunaan sensor *rotary encoder* yang membutuhkan pin *intterup* pada Arduino mega untuk pin *interrup* terdapat pada pin 2,3,18,19,20,21 namun untuk pin 20,21 yang merupakan pin SDA SCL di gunakan untuk perangkat I2C guna menampilkan karakter pada LCD 16x2.

Pada pin yang dihubungkan ke driver motor bts terdapat dua pin digital dengan pin R_EN dan L_EN dihubungkan langsung dengan 5v untuk input High agar Ketika RPWM dan LPWM diberikan input High – high atau low – low maka

motor akan mengunci, untuk mengatasi motor agar tidak mengunci maka perlu memberikan input high atau low ke pin enebel dangan menghubungkan pin enebel ke pin digital Arduino .

Penggunaan pin serial yang digunakan untuk komunikasi antar board dalam hal ini esp32 devkit 1 menggunakan pin tx3 dan rx3 serta sebagai supplay karena esp32 bekerja pada tegangan 3.3 v maka di hubungkan langsung menuju pin 3.3v pada Arduino Mega, berikut merupakan Tabel input/outpun pin pada Arduino mega

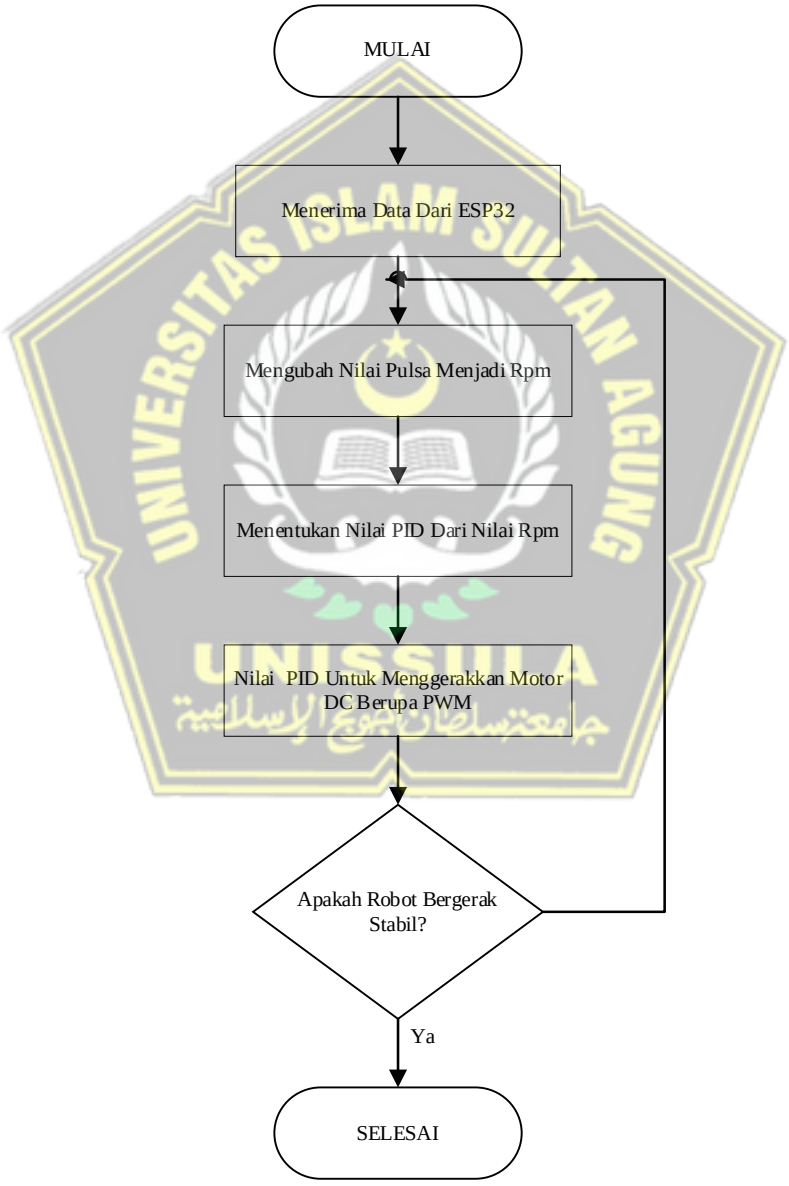
Tabel 3. 1 input/output pin pada Arduino meda

NO	Pin Arduino	Keterangan
1	13,12	Dihubungkan dengan driver motor MA sebagai control PWM
2	11,10	Dihubungkan dengan driver motor MB sebagai control PWM
3	9,8	Dihubungkan dengan driver motor MC sebagai control PWM
4	7,6	Dihubungkan dengan driver motor MD sebagai control PWM
5	18,36	Dihubungkan dengan pin channel A dan channel B rotary encoder motor MB
6	19,34	Dihubungkan dengan pin channel A dan channel B rotary encoder motor MD
7	3,32	Dihubungkan dengan pin channel A dan channel B rotary encoder motor MA
8	2,30	Dihubungkan dengan pin channel A dan channel B rotary encoder motor MC
9	20,21(SDA,SCL)	Dihubungkan ke pin SDA,SCL pada driver I2C LCD
10	14,15(TX,RX)	Dihubungkan ke pin RX,TX ESP32 sebagai komunikasi Serial
11	Vin	Dihubungkan ke pin positif SMKDSP (sumber tegangan) sebagai masukan sumber tegangan Arduino
12	Gnd	Dihubungkan ke pin negatif SMKDSP (sumber tegangan) serta di hubungkan ke setiap soket Molex untuk kutub negative

13	5V	Dihubungkan ke setiap soket molex sebagai input sumber tegangan di setiap komponen
14	3.3V	Dihubungkan ke pin vin esp32 sebagai sumber tegangan karna bekerja pada tegangan tersebut

3.2 Perancangan *Software*

Berikut merupakan diagaram alir perancangan *software* :



Gambar 3. 6 Diagram alir perancangan software

Perancangan *software* dilakukan dengan cara menentukan nilai setpoint rpm yang akan dicapai untuk dibaca dan diproses oleh Arduino, *feedback* dari *rotary encoder* akan memberi informasi kepada mikrokontroler mengenai nilai rpm roda yang kemudian akan diketahui nilai eror rpm terhadap setpoint, setelah nilai eror diketahui maka PID akan mengusahakan sisten agar nilai rpm selalu berada pada nilai setpoint yang ditentukan, maka keluaran nilai kontrol PID berupa nilai PWM yang digunakan untuk mengkontrol motor DC

Pada Gambar 3.8 menjelaskan gambaran mengenai sistem kerja dari program kontrol kecepatan motor, apabila nilai rpm tidak sesuai dengan setpoint yang ditentukan maka sitem akan mengupayakan untuk menuju ke setpoint yang telah ditentukan. Dalam hal ini adalah mencari kondisi dimana kecepatan PWM diatur agar robot bisa menuju ke setpoint yang diberikan, hal diatas bisa menggunakan kontrol PID pada robot *four omni-directional*

3.2.1. perancangan program perhitungan Rpm

Dalam penelitaian ini menggunakan motor DC PG45 yang telah terdapat *internal rotary encoder* dengan nilai *pulse per rotation* (ppr) sebanyak 7 ppr kemudian rasio *gearbox* dari motor memiliki perbandingan 1:19 maka dapat di ketahui nilai poros roda memiliki 133 ppr

Ppr roda = 19 x 7

Ppr roda = 133

Dalam pemrograman akan dilakukan *time sampling* dalam setiap 0.03 detik (30 *milisecound*), kemudian diperlukan nilai frekuensi atau rps (*Revolution per second*) *rotary encoder* unruk mencari nilai frekuensi signal maka digunakan persamaan 2.11 sebagai berikut.

$$f = rps = \frac{n}{T_S}$$

Time Sampling dibagi 1000 karena dalam *timesampling* masih dalam satuan *millisecond*, karena unruk mendapatkan nilai Rpm satuan yang digunakan adalah second. Kemudian untuk mendapatkan satuan revolusi per menit perlu

dikalikan 60 karena dalam satu detik sama dengan 60detik maka digunakanlah persamaan 2.12 untuk mendapatkan nilai Rpm.

$$rpm = \frac{60f}{133}$$

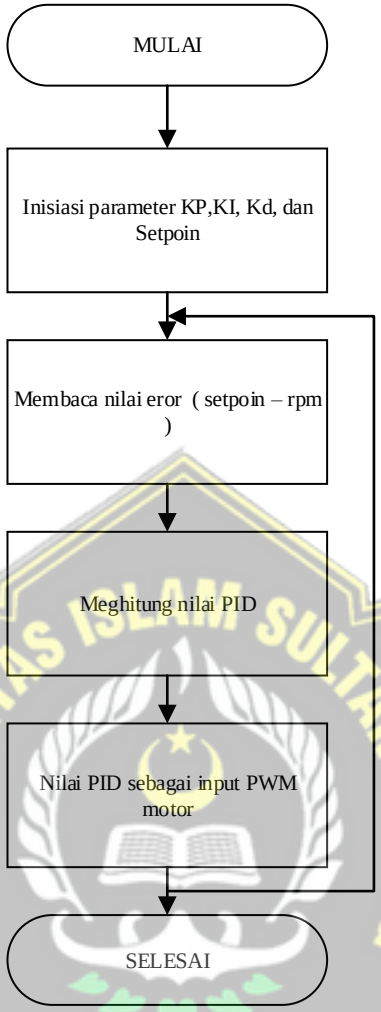
Dari perhitungan tersebut telah didapat rumus perhirungan rpm maka dari rumus akan dituangkan dalam bentuk program, nilai Rpm sendiri digunakan sebagai umpan balik dari kontrol PID, berikut merupakan algoritma pemrograman perhitungan nilai Rpm.

```
void bacarpm (){

  unsigned long timenow = millis();
  float deltaTime = timenow - timelast;
  if(deltaTime>=TimeSampling){
  float freqSignal = (float)pulse/(deltaTime/1.0e3);
  float freqSignal1 = (float)pulse2/(deltaTime/1.0e3);
  float freqSignal2 = (float)pulse3/(deltaTime/1.0e3);
  float freqSignal3 = (float)pulse4/(deltaTime/1.0e3);
  pulse = 0;
  pulse2 = 0;
  pulse3 = 0;
  pulse4 = 0;
  float rpm  = freqSignal * 60.0 /133;
  float rpm2 = freqSignal1 * 60.0 / 133;
  float rpm3 = freqSignal2 * 60.0 / 133;
  float rpm4 = freqSignal3 * 60.0 / 133;
}
```

3.2.2. Perancangan kontrol PID

PID (Proportional-Integral-Derivative) adalah metode kontrol yang digunakan dalam sistem kontrol untuk mengatur dan mempertahankan variabel suatu sistem pada setpoint yang diinginkan. PID merupakan salah satu teknik kontrol yang paling umum digunakan dalam berbagai aplikasi, termasuk kontrol suhu, posisi, kecepatan, dan banyak lagi.



Gambar 3. 7 Diagram alir perancangan kontrol PID

Dari gambar 3.7 bahwasanya tahapan merancang PID menggunakan metode thuning ZN1 diawali inisiasi parameter pada header progam kemudian program akan membaca nilai eror yang akan dijadikan nilai refrensi setelah nilai eror didapat maka sistem akan melakukan perhitungan dari persamaan PID, nilai kluaran PID digunakan sebagai input PWM motor DC.

Persamaan dengan tiga kombinasi ditunjukan pada persamaan 2.7. Dari persamaan 2.7 tersebut perlu di ubah dalam Bahasa program maka kontrol PID pada program ditunjukan pada persamaan

$$out = K_P \times eror + K_i \times (eror + lasteror) \times t_s + \frac{K_d}{t_s} \times (eror - lasteror)$$

Dimana

- K_p adalah konstanta proposional
- K_i adalah konstanta integral
- K_d adalah konstanta derivativ
- Eror adalah nilai nilai kesalahan (setpon – rpm)
- Lasteror adalah nilai kesalahan sebelumnya
- t_s aadalah time sampling

Berikut merupakan algoritma perancangan kontrol PID :

```
void pidrun(){
    eror = setpoin - rpm;
    integral = (eror + preveror )*dt;
    derivativ = (eror-preveror);
    out=((kp*eror)+(ki*integral)+(kd/dt)*derivativ);
    preveror = eror;
}
```

3.2.3 Perancangan pembacaan nilai *rotary encoder*

Pembacaan nilai *rotary encoder* dilakukan dengan cara menghitung nilai masukan pada pin *interrupt* dalam bentuk pulsa. Berikut script program fungsi pembacaan nilai *rotary encoder* :

```
Void setup(){
attachInterrupt(digitalPinToInterrupt(3),encoder,
CHANGE);
attachInterrupt(digitalPinToInterrupt(18),encoder2,
CHANGE);
attachInterrupt(digitalPinToInterrupt(2),encoder3,
CHANGE);
attachInterrupt(digitalPinToInterrupt(19),encoder4,
CHANGE);
}
```

```
}
void encoder(){
  pulse++;
}
void encoder2(){
  pulse2++;
}
void encoder3(){
  pulse3++;
}
void encoder4(){
  pulse4++;
}
```

3.2.4 Perancangan program gerak robot

Pergerakan robot beroda dipengaruhi oleh arah putaran masing-masing motor, pengaturan arah gerak motor dipermudah dengan pemberian nilai positif dan, dimana positif berarti motor diperintah berputar searah jarum jam dengan negatif nilai PWM yang dimasukkan, lalu nilai negative berarti motor bergerak berlawanan jarum jam, berikut untuk scrip program gerak manual robot :

```
void MA (int a){
  if (a > 0){
    analogWrite(motora1,a);
    digitalWrite(motora2,LOW);
  }
  else if (a < 0 ){
    analogWrite (motora1,255+a);
    digitalWrite(motora2,HIGH);
  }
  else {
    digitalWrite(motora1,HIGH);
    digitalWrite(motora2,HIGH);
  }
}
void MB (int b){
  if (b > 0){
    analogWrite(motorb1,b);
    digitalWrite(motorb2,LOW);
  }
  else if (b < 0 ){
    analogWrite (motorb1,255+b);
```

```
    digitalWrite(motorb2,HIGH);
  }
  else {
    digitalWrite(motorb1,HIGH);
    digitalWrite(motorb2,HIGH);
  }
}
void MC (int c){
  if (c > 0){
    analogWrite(motorc1,c);
    digitalWrite(motorc2,LOW);
  }
  else if (c < 0 ){
    analogWrite (motorc1,255+c);
    digitalWrite(motorc2,HIGH);
  }
  else {
    digitalWrite(motorc1,HIGH);
    digitalWrite(motorc2,HIGH);
  }
}
void MD (int d){
  if (d > 0){
    analogWrite(motord1,d);
    digitalWrite(motord2,LOW);
  }
  else if (d < 0 ){
    analogWrite (motord1,255+d);
    digitalWrite(motord2,HIGH);
  }
  else {
    digitalWrite(motord1,HIGH);
    digitalWrite(motord2,HIGH);
  }
}
```

3.2.5 Perancangan Komunikasi esp32 sebagai kontrol

Pada program ini menggunakan input esp32 yang dihubungkan ke pin tx,rx guna perintah kendali robot itu sendiri ,masukan yang diterima sendiri yaitu berupa code ASCII yang sebelumnya telah diuplod pada esp32. Berikut scrip program komunikasi esp32 sebagai kontrol:

```
void bacajoy(){
```

```
  if(Serial3.available()){
    Joy = Serial3.read();
  }
}

void loop() {
  if(Joy=='1'){MAJU();}//ATAS
  else if(Joy=='2'){MUNDUR();}//R1
  else if(Joy=='-'){BERHENTI();}//BAWAH

}
```

3.3. Pengujian

Pada tahap ini sistem yang telah dirancang dan dibuat baik *hardware* maupun *software* selanjutnya perlu dilakukan pengujian guna mengetahui keandalan sistem, apabila sistem dinilai belum sesuai maka akan dilakukan evaluasi dan diskusi Bersama pihak-pihak terkait.

Untuk mendapatkan nilai presentase eror pergeseran sudut robot maka dapat menggunakan persamaan trigonometri berikut :

$$Eror = \frac{\propto}{180} \tag{3.1}$$

$$\propto = tan\frac{jarank\ melemnceng}{jarak\ tempuh} \tag{3.2}$$

Persamaan ini digunakan karena perlu diketahuinya sudut pergeseran robot pada saat pengujian gerak robot, dari nilai sudut pergeseran robot tersebut nantinya akan di jadikan sebagai nilai perbandingan presentase sebelum dan sesudah menggunakan PID, Adapun metode yang akan dilakukan untuk pengujian sistem sebagai berikut:

1. Pengujian putaran motor terhadap gerak robot
2. Kalibrasi Rpm bertujuan untuk memastikan bahwasanya perhitungan yang dilakukan oleh mikrokontroler telah sesuai dengan nilai Rpm sebenarnya.
3. Pengujian gerak robot dengan kontrol open loop robot digerakkan dari titik A ke titik B untuk melihat seberapa eror geraknya.

4. Tuning PID menggunakan metode *Ziegler-Nichols 1*
5. Pengujian kestabilan gerak robot setelah menggunakan kontrol PID

BAB IV
PENGUJIAN DAN ANALISA

Pada Bab ini berisi tentang data hasil pengujian dan Analisa besarnya nilai Rpm yang terukur oleh *rotary encoder*, pengujian dilakukan dengan cara membandingka perpindahan pergerakan robot sebelum dan sesudah menggunakan sistem kontrol PID.

4.1. Pengujian Putaran Motor untuk Gerak Robot

Pengujian ini bertujuan untuk mengetahui gerak perpindahan robot, dengan melakukan pengujian arah putaran tiap roda omni dan gerak perpindahan arah robot yang dihasilkan dari putaran tersebut, pada table 4.1 telah disajikan data hasil pengujian dimana CW (*clockwise*) atau searah jarum jam dan CCW (*Counter clockwise*) atau berlawanan arah jarum jam, arah putaran motor tersebut yang nantinya akan menentukan arah dari gerak robot itu sendiri.

Tabel 4. 1 Arah putaran motor untuk gerak robot

Arah Robot	MA	MB	MC	MD
Maju	CW	CCW	CW	CCW
Mundur	CCW	CW	CCW	CW
Kanan	CCW	CCW	CW	CW
Kiri	CW	CW	CCW	CCW
Maju kiri	-	CW	CCW	-
Maju kanan	CCW	-	-	CW
Mundur kiri	CW	-	-	CCW
Mundur kanan	-	CCW	CW	-

4.2. Kalibrasi Nilai RPM

Kalibrasi RPM dilakukan untuk memastikan perhitungan nilai RPM yang dilakukan oleh mikrokontroler telah sesuai dengan rpm aktual, penyesuaian tersebut dilakukan dengan cara membandingkan rpm yang dihitung oleh

mikrokontroler dengan nilai rpm aktual yang terbaca oleh alat standar. Pada pengujian ini alat menggunakan *tachometer* untuk menentukan RPM sebenarnya.

Tabel 4. 2 Data kalibrasi nilai rpm pada Arduino dan tachometer

No	PWM	Rpm Terbaca pada *Tachometer*	Rpm Terbaca pada Arduino	Persentase eror
1	15	51,5	50,5	2%
2	25	83,4	82,5	1%
3	50	164,8	163,3	1%
4	75	243	244,9	1%
5	100	326,5	326,7	0%
6	125	406	407	0%
7	150	488	488	0%
8	175	572	570	0%
9	200	663,6	665,4	0%
10	255	852,9	857,5	1%
Rata-rata nilai eror				0,6%

Gambar 4. 1 Kalibrasi RPM menggunakan tachometer

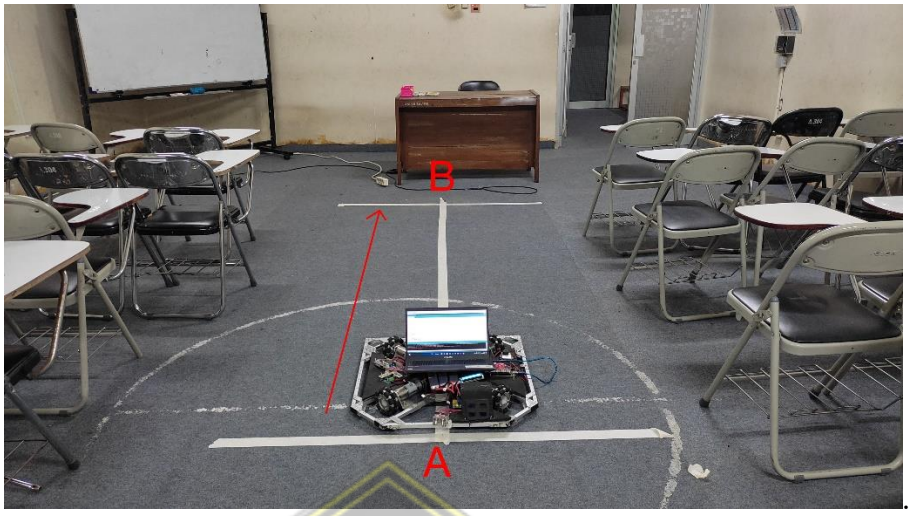
Gambar 4. 2 Kalibrasi RPM menggunakan perhitungan Arduino

Dari kalibrasi pembacaan nilai Rpm pada kontroler terdapat nilai perbedaan yang cukup kecil yaitu dengan rata-rata 0,6 %, maka dari itu dapat dikatakn bahwa pembacaan kecepatan motor pada mikrokontroler telah cukup baik dan dapat digunakan sebagai umpan balik dari sistem kontrol PID.

4.3. Pengujian gerak robot dengan kontrol terbuka (*Open loop*)

Pada pengujian ini dilakukan bertujuan untuk melihat arah gerak robot dengan menggunakan kontrol terbuka, kemudian hasil dari pengujian akan digunakan sebagai perbandingan dengan hasil pengujian *close loop control* menggunakan metode PID.

Pada pengujian ini robot akan digerakkan dari titik A ke titik B untuk mengetahui pergerakan robot dengan kontrol *open loop*, kemudian akam diambil berupa eror sudut dari titik A ke titik B serta eror nilai Rpm pada masing-masing penggerak. Untuk menguji robot akan digerakkan kearah hadap depan (arah hadap robot) dengan jarak 3 meter, kemudian akan dilakukan pengukuran pergeseran robot dari sumbu gerak arah robot dan nilai eror Rpm pada masing masing penggerak yang mempengaruhi pergeseran arah hadap robot.



Gambar 4. 3 Titik awal pengujian gerak robot open loop (titik A)

Gambar 4.3 tersebut menunjukan titik awal pengujian gerak robot (titik A) dengan menggunakan kontrol terbuka , pada tahap awal pengujian robot akan digerakkan menuju titik B, kemudian akan diukur nilai eror-nya yaitu jarak robot dengan sumbu gerak robot serta nilai eror pada Rpm masing-masing penggerak robot.

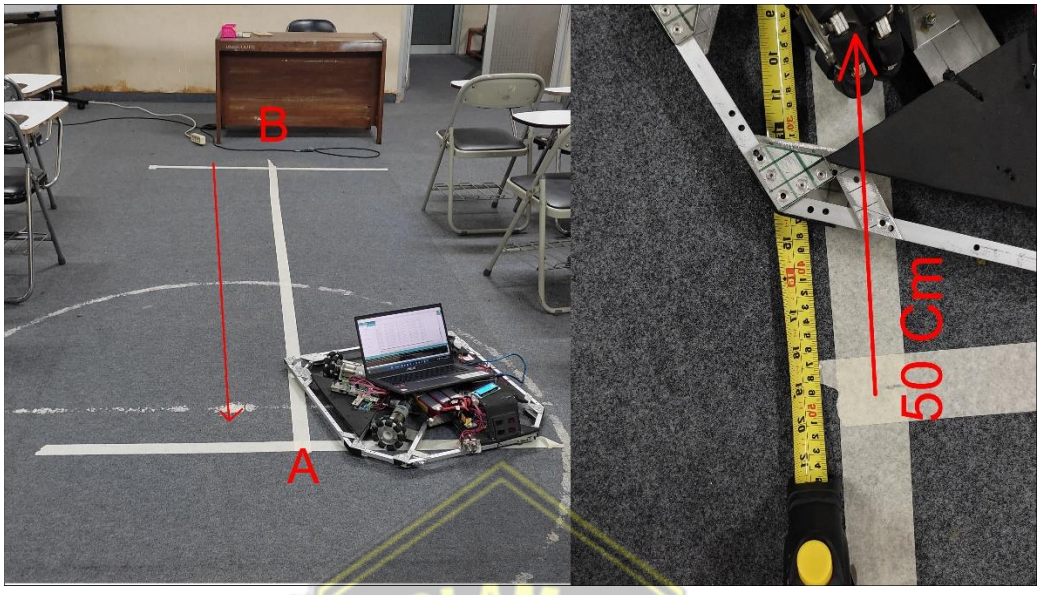
Gambar 4. 4 Titik B pengujian gerak robot dan besar nilai erornya

Tabel 4. 3 Data rpm gerak robot dari titik A ke titik B

No	PWM yang diset	MA	MB	MC	MD
1	25	59,10	66,77	65,86	67,67
2	25	64,69	70,38	68,12	68,12
3	25	65,41	69,92	68,12	68,12
4	25	65,86	69,92	68,12	68,57
5	25	66,32	70,83	68,57	11,73
Rata - rata		64,276	69,564	67,758	56,842

Dimana MA,MB,MC dan MD merupakan sebutan untuk motor DC dengan posisi sesuai dengan Gambar 3.2, Nilai eror yang didapat dari pengujian gerak arah hadap maju dari titik A ke titik B , yaitu sebesar 20cm dengan nilai eror Rpm terjadi pada MA (motor A) dan MB (motor B)dimana nilai Rpm terbaca pada MA lebih rendah dengan rata-rata 64,276 Rpm serta nilai Rpm pada MB tinggi dengan rata-rata 69,564 Rpm, hal ini yang menyebabkan terjadinya perubahan arah gerak condong ke kiri, penyebab dari perubahan gerak sendiri biasanya karena order beban robot yang tidak seimbang atau karena dalam pembangunan robot terjadi ketidaksimetrisan pemasangan penggerak yang menyebabkan salah satu penggerak menerima gaya gesek yang lebih .

Setelah pengujian arah gerak hadap maju selanjutnya dilakukan tahap kedua yaitu pengujian arah gerak hadap mundur dengan menggerakkan robot dari titik B menuju ke titik A .



Gambar 4. 5 Pengujian gerak robot dari titik B ke titik A

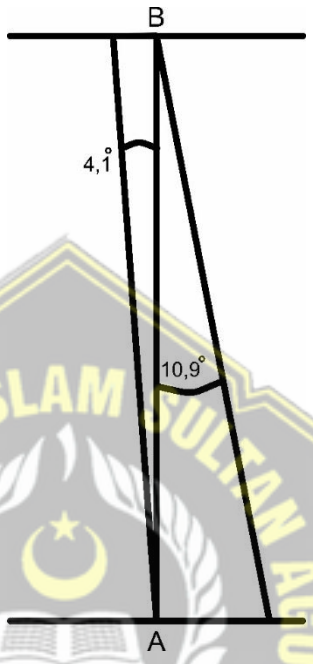
Tabel 4. 4 Data rpm gerak robot dari titik B ke titik A

No	PWM yang diset	MA	MB	MC	MD
1	25	49,62	68,12	71,73	67,22
2	25	72,63	68,57	72,63	67,67
3	25	73,08	68,12	72,18	68,57
4	25	73,53	69,02	73,08	67,67
5	25	72,63	40,60	40,60	35,19
Rata - rata		68,298	62,886	66,044	61,264

Pada pengujian arah gerak hadap mundur didapat nilai eror pergeseran arah robot dari sumbu gerak sebesar 50cm. pergeseran tersebut dapat dilihat pada gambar dimana pergeseran arah robot condong ke sebelah kanan saat di gerakkan mundur dengan data nilai eror Rpm pada tabel 4.4, dimana nilai Rpm pada MA (motor a) dan juga MC (motor c) memiliki nilai yang tinggi dengan rata-rata 68,298 Rpm dibandingkan nilai Rpm pada motor MB dan MD , hal ini yang menyebabkan arah gerak robot dominan ke kanan serta agak berputar.

Pada gambar dapat dilihat eror pergerakan robot saat digerakkan maju dan mundur cukup besar, hal ini disebabkan robot omni masih menggunakan

kontrol terbuka (*open loop*) serta terdapat beberapa penyebab lain seperti , titik beban pada robot yang tidak berada pada titik simetris dari dimensi *base robot* atau karena saat pembuatan robot terjadi eror pemasangan letak penggerak yang mengakibatkan tidak simetrisnya letak penggerak robot omni.



Gambar 4. 6 Eror pergerakan robot dengan kontrol open loop

Pada gambar dapat dilihat bahwasanya eror dari pergerakan pergeseran robot dari sumbu garis Tengah dimana robot bergerak dari titik A menuju ke titik B menghasilkan Pergeseran sebesar 4,1 derajat dari garis Tengah sedangkan saat di gerakkan dari titik B menuju titik A robot menghasilkan pergeseran sebesar 10,9 derajat, angka tersebut didapatkan dengan menggunakan persamaan 3.1, nilai derajat eror tersebut cukup besar karena yang dinginkan robot dapat bergerak stabil dengan lurus, hal ini disebabkan kontrol yang digunakan pada tiap penggerak masih menggunakan kontrol *open loop* serta penyebab lainnya yaitu titik beban robot yang tidak simetris dari dimensi base robot.

Tabel 4. 5 Hasil pengujian gerak robot tanpa PID

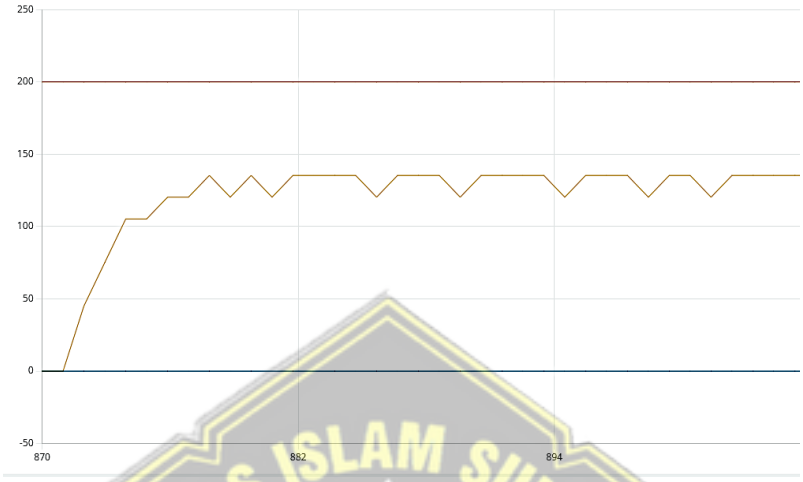
TANPA PID						
No	A-B	B-A	Sudut melenceng θ_1	Sudut melenceng θ_2	Eror θ_1	Eror θ_2
1	22 cm	58 cm	4,2	10,9	2,33 %	6,08 %
2	21 cm	53 cm	4,0	10,0	2,22 %	5,57 %
3	20 cm	49 cm	3,8	9,3	2,12 %	5,15 %
4	13 cm	60 cm	2,5	11,3	1,38 %	6,28 %
5	18 cm	51 cm	3,4	9,6	1,91 %	5,36 %
6	11 cm	64 cm	2,1	12,0	1,17 %	6,69 %
7	11 cm	60 cm	2,1	11,3	1,17 %	6,28 %
8	10 cm	48 cm	1,9	9,1	1,06 %	5,05 %
9	8 cm	53 cm	1,5	10,0	0,85 %	5,57 %
10	7 cm	61 cm	1,3	11,5	0,74 %	6,39 %
Nilai *Eror* Rata-rata					1,49 %	5,84 %

Dari tabel 4.5 hasil pengujian gerak robot tanpa PID dengan dilakukannya pengujian sebanyak sepuluh kali dengan hasil rata-rata nilai Eror θ_1 sebesar 1,49 % terhadap sumbu garis lurus saat digerakkan dari titik A ke titk B kemudian nilai rata-rata nilai Eror θ_2 gerak robot saat digerakkan dari titik B ke titik A yaitu sebesar 5,84 % dari sumbu garis lurus.

4.4. Tuning PID

Pada tahap ini bertujuan untuk mencari parameter Kp,Ki, dan Kd dengan nilai yang terbaik sesuai dengan yang diinginkan sitem, pada penelitian ini tunning dilakukan menggunakan metode *Zigler-Niclous method 1*, Adapun tahapan pertaman dari tunning ZN1 adalah menentukan nilai T dan L berdasarkan respon sistem yang didapat saat dikontrol dengan open loop.

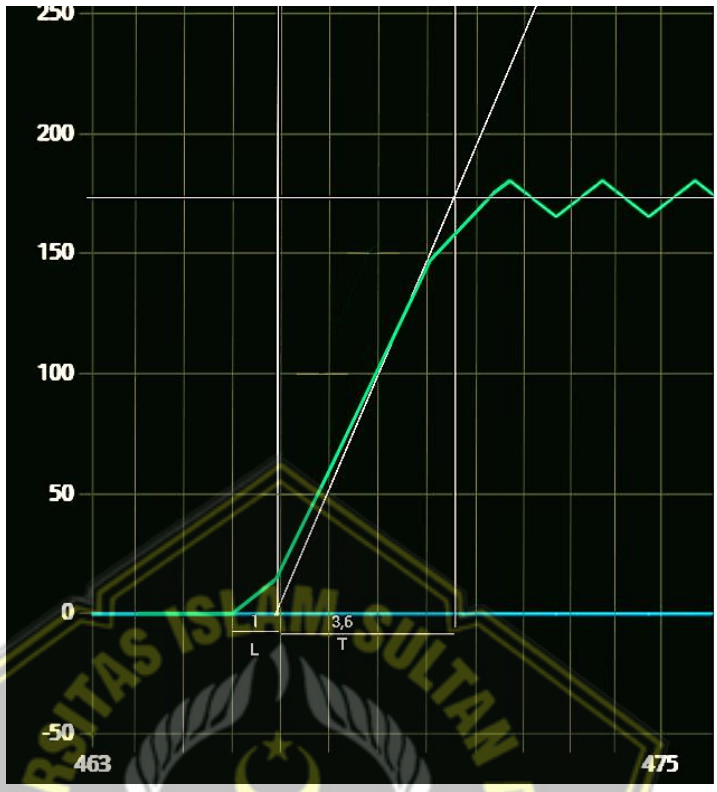
4.4.1. Menentukan nilai T dan L berdasarkan respon sistem



Gambar 4. 7 sinyal step kontrol open loop

Setelah didapatkan nilai respon awal seperti pada gambar, maka akan ditentukan parameter Kp, Ki dan Kd dari hasil T dan L dengan menggunakan ZN1 yaitu dengan memasukkan nilai tersebut ke dalam rumus seperti pada Tabel 2.1.

Untuk menentukan nilai T dan L, pertama-tama gambar garis horizontal pada sinyal step yang sudah dalam keadaan *steady state*, kemudian garis horizontal saat sinyal mulai step respons, dan kemudian garis diagonal pada sinyal step respons . maka didapatkan nilai T. dan L.



Gambar 4. 8 Sekema menentukan nila T dan L dari grafik step

Dari gambar 4.7 untuk menentukan nilai T dan L dengan cara menghitung Panjang waktu yang dilalui sinyal sesuai dengan data time pada tampilan serial plotter Arduino, maka di dapatkan nilai T = 3,6 dan nilai L = 1 yang selanjutnya akan di masukkan dalam rumus sesuai kontrol yang diinginkan.

4.4.2 Memasukkan nilai T dan L ke dalam rumus

Dari sinyal respon telah didapatkan nilai T dan L yang kemudian akan dilakukan perhitungan menggunakan rumus seperti pada table 2.1. Untuk penentuan tipe kontroler sendiri perlu dilakukan satu persatu, yaitu dengan menggunakan kontrol P terlebih dahulu. Jika respon sistem masih belum sesuai maka menggunakan tipe kontroler PI. Jika dengan kontrol PI respon sistem sudah sesuai yang diinginkan maka dapat diartikan bahwasanya sistem yang dibangun sudah cukup jika hanya menggunakan kontrol PI

Maka dari itu penelitian ini menggunakan kontrol PI rumus yang digunakan adalah pada bagian baris ke tiga dari table untuk menentukan nilai PID:

Diketahui : L = 1

T = 3,6

Kp = 0,9 x T/L

= 0,9 x 3,6/1

= 3,24

Ti = 2L

= 2 x 1

= 2

Untuk menentukan nilai Ki maka perlu memasukan nilai Ti kedalam rumus

Ki = Kp/Ti

maka dari rumus tersebut akan didapatkan nilai Ki

Diketahui : Kp = 3,24

Ti = 2

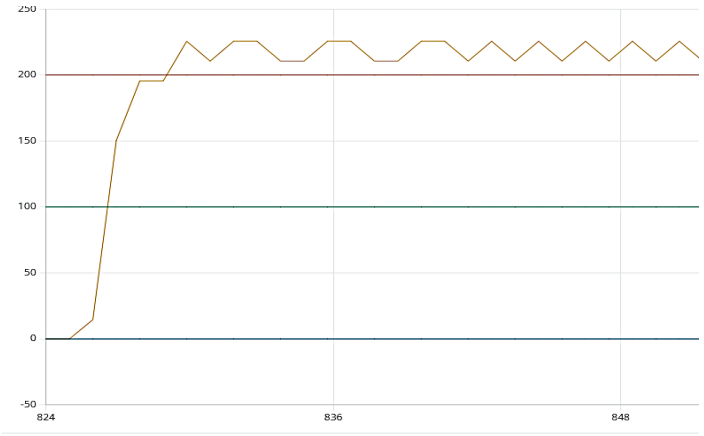
Ki = Kp/Ti

= 3,24 / 2

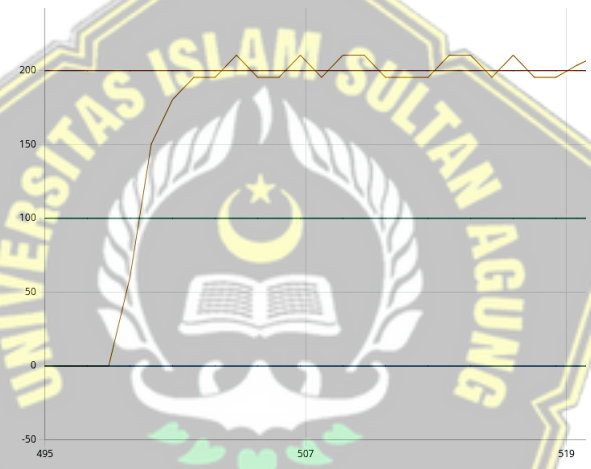
= 1,62

Dari hasil respon awal dan perhitungan yang sudah dilakukan didapatkan nilai parameter Kp, dan Ki seperti yang ditunjukan pada tabel

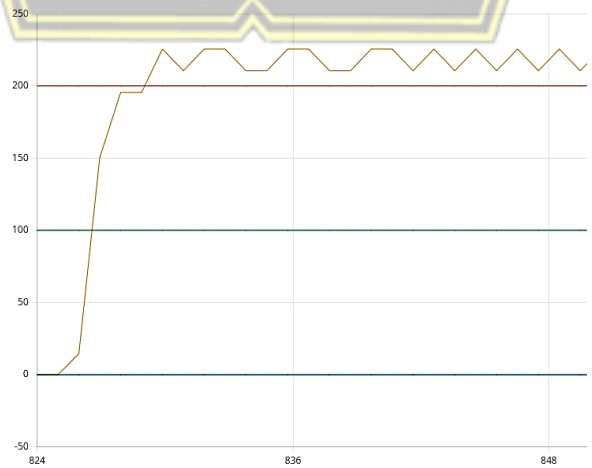
Parameter	Kp	Ki
PI	3,24	1,62



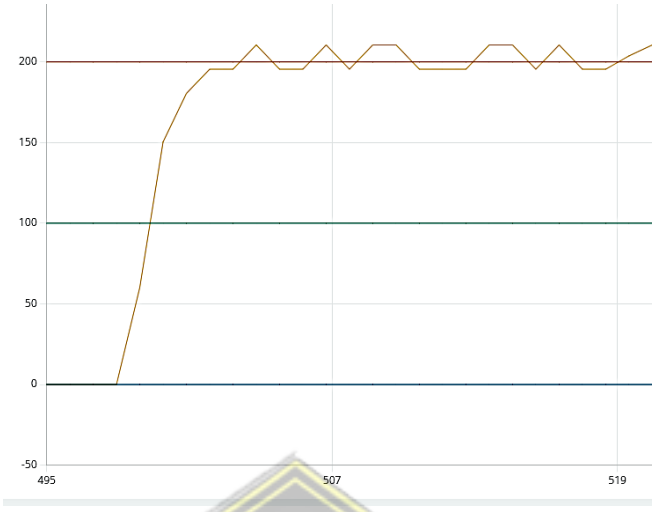
Gambar 4. 9 sinyal respon sistem menggunakan parameter hasil tuning pada motor MA



Gambar 4. 10 sinyal respon sistem menggunakan parameter hasil tuning pada motor MB



Gambar 4. 11 sinyal respon sistem menggunakan parameter hasil tuning pada motor MC



Gambar 4. 12 sinyal respon sistem menggunakan parameter hasil tuning pada motor MD

Dari gambar diatas respon setelah menggunakan parameter Kp,Ki,Kd dari hasil tuning menggunakan metode ZN1 didapatkan bahwa sinyal respon melebihi setpoint namun untuk *steady state* dan *overshoot* sudah cukup baik, sesuai dengan yang dikatakan pada buku *modern control engineering* milik *Katsuhiko ogata* bahawasanya untuk memperbaiki respon sistem maka perlu dilakukan *fine tuning* dengan menggunakan parameter PID yang telah didapatkan sebagai patokan agar respon sistem menjadi seperti yang diharapkan.

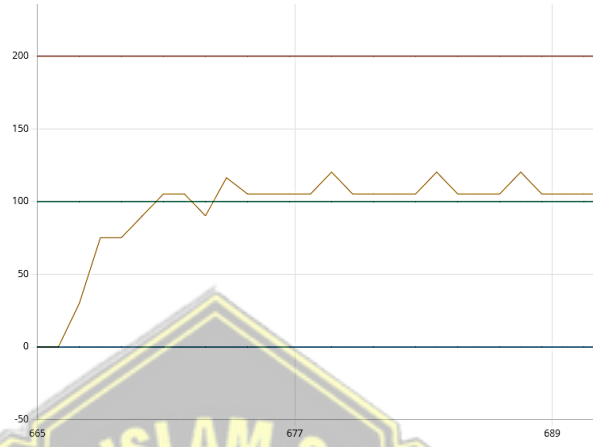
4.4.3. *Fine tuning*

Setelah beberapa kali melakukan *fine tuninig* maka didapat parameter yang dinilai paling optimal pada masing-masing motor, nilai parameter sendiri tidak memiliki nilai pasti yang terpenting respon dari sistem sudah sesuai dengan yang diinginkan, maka didapatkan parameter PI seperti yang ditunjukan pada tabel

Tabel 4. 6 parameter PI dari hasil Fine tuning

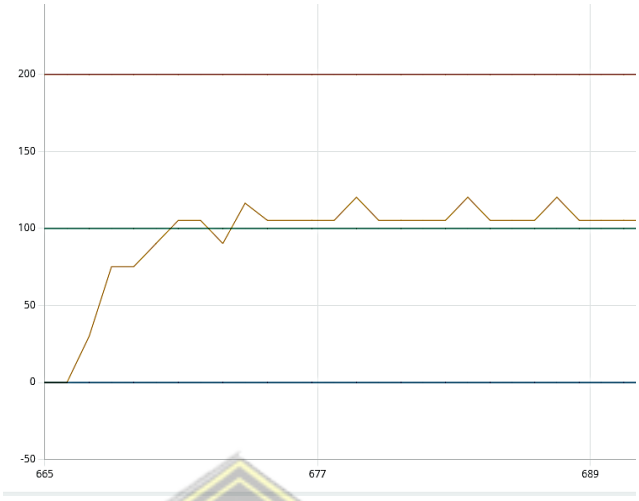
Parameter	Kp	Ki
PI motor MA	3.2	2
PI motor MB	3.2	2
PI motor MC	3.2	2
PI motor MD	3.2	2

Dari tabel 4.6 parameter tersebut dimasukan kedalam program, berikut merupakan grafik dari hasil *fine tunning* dengan hasil sinyal respon dianggap sudah yang terbaik.

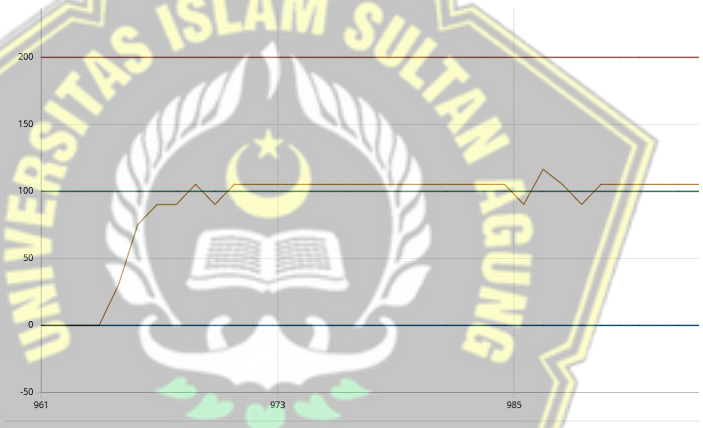


Gambar 4. 13 sinyal respon sistem setelah penyesuaian parameter hasil tuning pada motor MA

Gambar 4. 14 sinyal respon sistem setelah penyesuaian parameter hasil tuning pada motor MB



Gambar 4. 15 sinyal respon sistem setelah penyesuaian parameter hasil tuning pada motor MC



Gambar 4. 16 sinyal respon sistem setelah penyesuaian parameter hasil tuning pada motor MD

Pada gambar diatas respon setelah melakukan tahapan *fine tunning* parameter, respon telah sesuai dengan setpoint yang ditentukan dan mencapai *steady state* yang cukup bagus dengan nilai parameter sesuai dengan pada tabel 4.4.

4.5. Pengujian kestabilan motor DC dan gerak robot dengan control PID

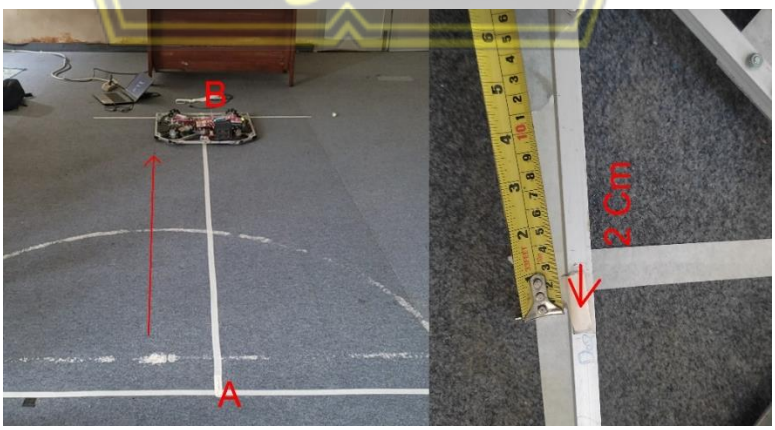
Pada pengujian ini, dilakukan pengujian stabilitas masing-masing kecepatan motor *omni-wheel* dan arah gerak robot setelah menggunakan kontrol PID, maka akan dapat diketahui arah error akibat pergerakan gerak robot dibandingkan dengan gerak kontrol loop terbuka yang dilakukan pada pengujian sebelumnya..

4.5.1. Pengujian gerak robot dengan kontrol PID

Pengujian ini sama tekniknya dengan pengujian pergerakan robot dengan kontrol terbuka, dimana robot akan digerakkan dari titik A ke titik B dengan jarak 3 meter dengan arah hadap robot kedepan, kemudian akan diukur besar pergeseran nilai eror dari sumbu gerak lurus robot

Gambar 4. 17 Titik awal pengujian gerak robot

pada gambar 4.17 eror pergeseran dari sumbu lurus yang terukur pada pergerakan robot setelah menggunakan kontrol PID yaitu sebesar 2 cm dimana arah eror sama dengan pengujian gerak kontrol terbuka yaitu kesebelah kiri namun pergeseran eror menjadi lebih kecil setelah menggunakan kontrol PID.

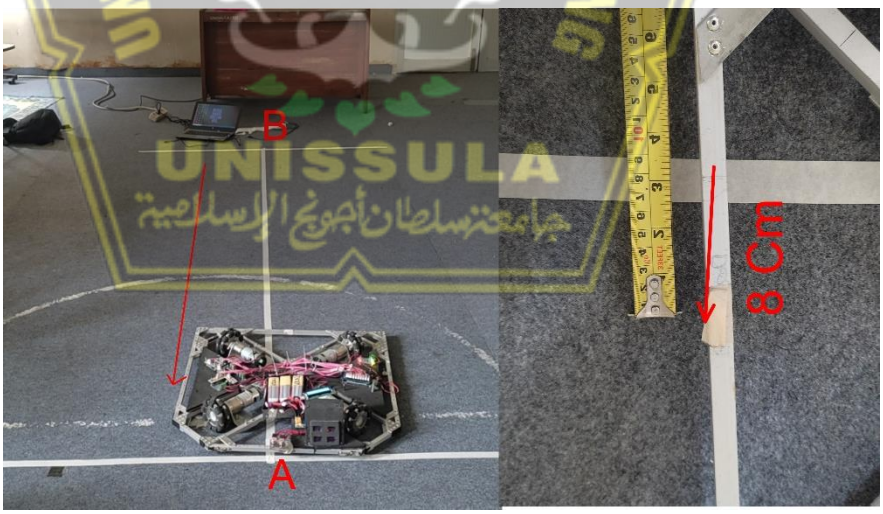


Gambar 4. 18 Pengujian pergerakan robot dari titik A ke titik B menggunakan kontrol PID

Tabel 4. 7 Nilai rpm gerak robot dari titik A ke titik B menggunakan kontrol PID

No	MA	MB	MC	MD
1	79,15	87,06	47,49	87,06
2	96,67	96,67	88,61	96,67
3	96,67	88,61	96,67	88,61
4	96,67	104,73	104,73	96,67
5	101,11	93,34	93,34	93,34
6	94,97	94,97	94,97	102,89
7	99,4	91,75	91,75	91,75
8	96,67	96,67	104,73	96,67
9	102,89	102,89	94,97	94,97
10	93,34	93,34	93,34	101,11
Rata-rata	95,754	95,003	95,06	94,974

Kemudian robot diuji pergerakkannya arah hadap mundur untuk digerakkan menuju titik awal, diaman pada gambar 4.18 menujukan eror gerak mundur robot sebesar 8cm dari titik awal, nilai tersebut lebih kecil erornya daripada nila pergeseran pada pengujian secara kontrol terbuka sebesar 50cm .



Gambar 4. 19 Pengujian pergerakan robot dari titik B ke titik A menggunakan kontrol PID

Tabel 4. 8 Nilai rpm gerak robot dari titik B ke titik A menggunakan kontrol PID

No	MA	MB	MC	MD
1	85,56	93,34	101,11	101,11
2	85,56	101,11	101,11	93,34
3	87,06	102,89	94,87	87,06
4	98,61	96,67	104,73	96,67
5	94,97	94,97	94,97	87,06
6	87,06	94,97	94,97	94,97
7	98,61	96,67	104,73	98,61
8	94,97	102,89	94,97	87,06
9	87,06	94,97	94,97	94,97
10	90,23	98,43	106,63	90,23
Rata - rata	97,969	97,691	98,306	97,108

Gambar 4. 20 Eror pergerakan robot dengan kontrol PID

Pada gambar dapat dilihat bahwasanya eror dari pergerakan pergeseran robot dari sumbu garis Tengah dimana robot bergerak dari titik A menuju ke titik B menghasilkan bergeseran sebesar 0,3 derajat dari garis Tengah sedangkan saat di gerakkan dari titik B menuju titik A robot menghasilkan pergeseran sebesar 1,5 derajat, dilihat dari sudut yang dibentuk dari pergerakan robot menghasilkan angka yang lebih kecil dibandingkan dengan sudut eror yang dibentuk dengan kontrol *open loop*.

Tabel 4. 9 Hasil pengujian pergerakan robot dengan kontrol PID

DENGAN PID						
No	A-B	B-A	Sudut melenceng θ_1	Sudut melenceng θ_2	Eror θ_1	Eror θ_2
1	10 cm	5 cm	1,9	1,0	1,06 %	0,53 %
2	3 cm	10 cm	0,6	1,9	0,31 %	1,06 %
3	9 cm	6 cm	1,7	1,1	0,95 %	0,63 %
4	9 cm	15 cm	1,7	2,9	0,95 %	1,59 %
5	10 cm	5 cm	1,9	1,0	1,06 %	0,53 %
6	2 cm	10 cm	0,4	1,9	0,21 %	1,06 %
7	3 cm	13 cm	0,6	2,5	0,31 %	1,37 %
8	10 cm	8 cm	1,9	1,5	1,06 %	0,84 %
9	6 cm	14 cm	1,1	2,7	0,63 %	1,48 %
10	5 cm	6 cm	1,0	1,1	0,53 %	0,63 %
RATA2					0,71 %	0,97 %

Dari tabel 4.9 hasil pengujian gerak robot dengan kontrol PID samahalnya dengan pengujan sebelumnya dilakukan pengujian sebanyak sepuluh kali dengan hasil rata-rata nilai *Eror* θ_1 gerak robot dari sumbu garis lurus saat digerakkan dari titik A ke titk B sebesar 0,71 % dan nilai rata-rata *eror* θ_2 gerak robot saat di gerakkan dari titik B ke titik A yaitu sebesar 0,97 % dari sumbu garis lurus, maka dapat dikatakan kontrol PI yang dibuat dapat membuat gerak robot lebih lurus dan dapat menekan eror pergeserah robot, nilai eror tersebut dianggap sudah cukup baik walaupun dirasa masih cukup besar hal ini dikarenakan pengaruh dari jumlah ppr pada *rotary encoder* yang digunakan sebagai umpan balik dari kontrol PID .

BAB V
PENUTUP

5.1. Kesimpulan

Dari hasil penelitian yang telah dilakukan maka dapat disimpulkan sebagai berikut:

1. Robot *four omni-directional* dapat bergerak tidak lurus dan kurang stabil di karenakan pembagian beban robot yang tidak seimbang sehingga menyebabkan salah satu motor penggerak harus menerima beban lebih atau dapat disebabkan juga karena pembuatan *frame base* dan peletakan posisi motor yang kurang simetris sehingga membuat salah satu motor penggerak menerima beban yang lebih sehingga putaran motor berkurang yang menyebabkan robot membelok.
2. Untuk mengatasi permasalahan robot yang bergerak tidak lurus dan kurang stabil maka diperlukan kontrol *close loop* menggunakan PID, dari penelitian yang telah dilaksanakan didapatkan nilai perbandingan eror pergeseran sudut arah robot sebelum dan sesudah PID sebesar 1,7 % untuk arah hadap maju dan 1,5% untuk arah hadap mundur.
3. Untuk mendapatkan parameter PID yang optimal serta efisien untuk mendapatkanya maka diperlukan metode tunning PID, penelitian ini menggunakan metode tunning *zigler niclous method 1* dengan memanfaatkan sinyal step dari motor maka didapatkan parameter PID dengan kp = 3,2 , ki = 2, dan kd = 0 nilai tersebut dirasa sudah menghasilkan respon yang cukup baik, nilai kd 0 sendiri karena pada saat melakukan fine tunniing pada parameter kd tidak berpengaruh pada sistem maka kontrol yang digunakan adalah kontrol PI dengan nilai kd = 0.

5.2. Saran

Saran yang dapat ditarik dari penelitan tentang kendali gerak robot four omni directional ini untuk pengembangan berikutnya adalah sebagai berikut:

Penelitian ini menggunaka rotary encoder internal PG45 dengan nilai PPR (Pulse Per Rotation) yaitu 7 ppr yang menyebabkan feedback terhadap sistem

cukup lamban dan sinyal yang di hasilkan terlalu rengang, maka perlu menggunaka *rotary encoder* internal dengan nilai PPR tinggi atau dengan menambah rotary external dengan nilai tinggi pula agar feddback terhadap sistem dapat lebih cepat maka dapat menekan peresentase nilai eror. Adapun penambahan sensor *gyroscope* untuk orientasi sumbu x pada robot agar arah hadap robot dapat lebih stabil dan tidak terkendala akibat selip roda.

DAFTAR PUSTAKA

[1] T. A. S. S. S. Wahyuono, H. S. S. S. Widyantara, and M. C. S. S. S. Wibowo, "KENDALI ROBOT MANUAL 4WD MECANUM WHEEL BERBASIS PID DENGAN MENGGUNAKAN ARM–CORTEX M4.," *Journal JCONES*, vol. 4, no. 1, 2015.

[2] A. Priyambudi, B. Firman, and S. Kristiyana, "Kendali Kecepatan Motor Pada Robot Dengan Empat Roda Omni Menggunakan Metode Pid," *Jurnal Teknologi Technoscientia*, pp. 209–217, 2017.

[3] D. A. Karnadi, "TA: Rancang Bangun Kontrol Kecepatan Robot Three Omni-Directional Menggunakan Metode PID (Proportional Integral Derivative)." Institut Bisnis dan Informatika Stikom Surabaya, 2018.

[4] S. K. Kholiqin, I. Sirajuddin, and S. Sungkono, "Sistem kendali Kecepatan Putaran Motor Robot Omnidirectional dengan 4 Penggerak Mekanum Menggunakan PID Berbasis myRIO," *Jurnal Elkolind: Jurnal Elektronika dan Otomasi Industri*, vol. 8, no. 3, pp. 293–299, 2021.

[5] R. Mochammad, "Menggunakan Kontroler Pid Pada Robot Kontes Robot Abu Indonesia (Krai)." Brawijaya University.

[6] M. A. Johnson and M. H. Moradi, *PID control*. Springer, 2005.

[7] Katsuhiko. Ogata, *Modern control engineering*. Prentice-Hall, 2010.

[8] N. N. Rokhmah, "Kendali Kecepatan Motor DC Dengan Metode PID Berbasis Arduino Uno," *Jurnal Teknik Elektro dan Vaksional*, 2018.

[9] A. Rachmawan, "Penentuan Posisi Robot Sepak Bola Beroda Menggunakan Rotary Encoder dan Kamera," *Undergraduate Thesis, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya*, 2017.

[10] Hidayat Khusnul, "Incremental Rotary Encoder," Malang, 2021. Accessed: Feb. 10, 2023. [Online]. Available: https://www.youtube.com/watch?v=UTRka_3nDkA&ab_channel=KhusnulHidayat

[11] W. S. Pambudi, "Rancang Bangun 3 Wheels Omni-Directional Mobile Robot Menggunakan Sensor Position Sensitive Device (PSD) Serta Sensor Vision Dengan Metode Kendali Fuzzy Logic Controller (FLC) Untuk Menghindari Halangan," *Semantik*, vol. 1, no. 1, 2011.

[12] A. Amilah, "SIMULASI SISTEM KENDALI KESTABILAN MOTOR DC," 2016.

[13] A. I. Putra, "Sistem Pengaturan Posisi Sudut Putar Motor DC Pada Model Rotary Parking Menggunakan Kontroler PID Berbasis Arduino MEGA 2560," *Website: elektro Studentjournal. ub. ac. idindex. php/teub/article/download/104/72*, 2013.

[14] N. L. Husni, S. Rasyad, M. S. Putra, Y. Hasan, and J. Al Rasyid, “Pengaplikasian Sensor Warna pada Navigasi Line Tracking Robot Sampah Berbasis Mikrokontroler,” *Jurnal Ampere*, vol. 4, no. 2, pp. 297–306, 2020.