

PROYEK AKHIR

**DESAIN SISTEM KONTROL *WHEELED HOLONOMIC*
ROBOT UNTUK ROBOT SEPAK BOLA**



Disusun Oleh:

Meta Saputri	NIRM	0031644
Suganda	NIRM	0031658

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2019**

**DESAIN SISTEM KONTROL *WHEELED HOLONOMIC*
ROBOT UNTUK ROBOT SEPAK BOLA**

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh:

Meta Saputri	NIRM	0031644
Suganda	NIRM	0031658

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2019**

LEMBAR PENGESAHAN

JUDUL PROYEK AKHIR

DESAIN SISTEM KONTROL *WHEELED HOLONOMIC* ROBOT UNTUK ROBOT SEPAK BOLA

Oleh:

Meta Saputri	0031644
Suganda	0031658

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

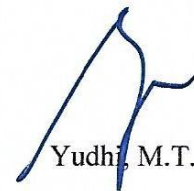
Menyetujui,

Pembimbing 1




M. Iqbal Nugraha, M.Eng.

Pembimbing 2



Yudhi, M.T.

Penguji 1



Indra Dwisaputra, M.T.

Penguji 2



Irwan, M.Sc., Ph.D.

Penguji 3



Eko Sulisty, M.T.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan dibawah ini:

Nama Mahasiswa 1 : Meta Saputri NIRM: 0031644

Nama Mahasiswa 2 : Suganda NIRM: 0031658

Dengan judul : Desain Sistem Kontrol *Wheeled Holonomic* Robot untuk
Robot Sepak Bola

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 30 Agustus 2019

Nama Mahasiswa

Tanda Tangan

1. Meta Saputri



.....

2. Suganda



.....

ABSTRAK

Sistem kendali kecepatan putaran roda pada robot sepak bola Sipak G3-Tim KRSBI Polmanbabel masih bersifat open loop. Hal ini mengakibatkan kestabilan gerak pada robot tersebut tidak bisa dijaga dengan baik, yang mana berdampak pada rendahnya kualitas dan repeatabilitas. . Proyek Akhir ini dibuat dengan tujuan untuk meningkatkan repeatabilitas pada sisi mobilitas robot sepak bola Polmanbabel dengan menerapkan sistem kendali tertutup berbasis PID. Penerapan sistem kontrol PID difokuskan tidak hanya pada menjaga kestabilan kecepatan putaran roda robot saja, tetapi juga dalam rangka untuk mengeliminasi atau memitigasi kesalahan keadaan tunak dan response time . Kinematika three omniwheels digunakan untuk memudahkan dalam transformasi kecepatan tiga buah roda omni menjadi satu kesatuan kecepatan robot.. Penentuan nilai awal konstanta PID pada proyek akhir ini menggunakan metode tuning Ziegler Nichols, yang kemudian disempurnakan secara manual dengan hasil akhir yakni $k_p=0,58$, $k_i=0,85$ dan $k_d=0,00002$. Hasil dari percobaan diperoleh bahwa respon kecepatan pada masing-masing roda sangat baik dengan overshoot = 4,167 %, rise time = 0,12 s, settling time = 0,38 s, dan steady state error= 3,3%. Adapun setelah dilakukan transformasi menggunakan kinematika tiga roda omni diperoleh bahwa rata-rata kesalahan orientasi robot adalah $\pm 5\%$, rata-rata kesalahan kecepatan adalah $\pm 1,45\%$ dan rata-rata kesalahan arah gerak adalah $\pm 0,6\%$.

Kata kunci: Robot Sepak Bola, Kontrol PID, *Wheeled Holonomic Robot*

ABSTRACT

System control of wheel's rotation speed on the robot soccer of Sipak G3-Team KRSBI Polmanbabel still open loop. This result in stability of motion on the tobot could not keep well, this is give impact to be low quality and repeatability. Purpose of the final project made for upgrading of reapeatability on mobility of the robot soccer Polmanbabel with apply close loop on system control base on PID. Application of the PID system control focused just not only at keep the stability of the wheel's rotation speed on the robot, but also on order for eliminated or ilimidating of steady state error and response time. Kinematic three omniwheels used to make it easy on transformation speed three omniwheels to be unity of robot's speed. Determination of the first value in PID constant on this final project use tuning method from Ziegler Nichols, which was then refined manually with the final result as $K_p=0.58$, $K_i=0.85$, and $K_d=0,00002$. The result of the experiment obtained that response of speed on each one wheel very well with overshoot=4.167%, rise time=0,12 s, settling time=0.38 s, and steady state error 3.3%. As for after the transformation use kinematic three omniwheels obtained that average error orientation of the robot is $\pm 5\%$, average error speed is $\pm 1,45\%$, average error direction of motion is $\pm 0,6\%$.

Keywords: Robot Soccer, PID control, Wheeled Holonomic Robot

KATA PENGANTAR

Assalamu'alaikum Warahmatulahi Wabarakatuh.

Puji dan syukur penulis panjatkan kehadiran Allah SWT, yang mana berkat rahmat dan karunia-Nya lah sehingga penulis dapat menyelesaikan Karya Tulis Proyek Akhir tepat pada waktunya. Karya Tulis Proyek Akhir ini disusun guna memenuhi salah satu persyaratan dan kewajiban mahasiswa untuk menyelesaikan kurikulum program Diploma III di Politeknik Manufaktur Negeri Bangka Belitung. Tentunya dalam penyusunan Karya Tulis Proyek Akhir dan melaksanakan Karya Tulis Proyek Akhir ini telah banyak pihak yang membantu penulis. Untuk itu penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Kedua orang tua dan keluarga yang terus memotivasi, memberikan dukungan dan terus mendoakan penulis dimanapun penulis berada.
2. Bapak Sugeng Ariyono, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Muhammad Iqbal Nugraha, M.Eng. selaku pembimbing I dan Bapak Yudhi, M.T. selaku pembimbing ke II yang telah meluangkan banyak waktu, tenaga serta pikiran dalam memberikan pengarahan pada penulisan Karya Tulis Proyek Akhir ini dan telah banyak pula memberi saran-saran dan solusi dari masalah-masalah yang penulis hadapi selama proses penyusunan Karya Tulis Proyek Akhir ini.
4. Bapak Eko Sulisty, M.T selaku kepala program studi D-III Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung.
5. Seluruh dosen-dosen Politeknik Manufaktur Negeri Bangka Belitung, yang telah mengajarkan banyak hal sehingga penulis menjadi seorang yang mempunyai wawasan dan ilmu, serta kepada seluruh staf administrasi Politeknik Manufaktur Negeri Bangka Belitung.
6. Seluruh rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah banyak membantu selama menyelesaikan Proyek Akhir.

Penulis menyadari bahwa Karya Tulis Proyek Akhir ini masih jauh dari kesempurnaan karena keterbatasan kemampuan penulis baik itu dari segi materi, maupun dalam penyampaian materi. Demikianlah Karya Tulis Proyek Akhir yang dapat penulis sampaikan semoga dapat bermanfaat dan berguna sebagaimana yang diharapkan. Akhir kata penulis ucapkan terimakasih.

Assalamu'alaikum Wr.Wb.

Sungailiat, 30 Agustus 2019

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	ii
ABSTRAK	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	3
BAB II DASAR TEORI	4
2.1 Robot Sepak Bola.....	4
2.2 <i>Holonomic Robot</i>	4
2.3 Kontrol PID (<i>Propositional Integral Derivatif</i>)	5
2.3.1 <i>Proportional Control</i>	6
2.3.2 <i>Integral Control</i>	6
2.3.3 <i>Derivative Control</i>	6
2.3.4 Metode Tuning parameter PID	7
2.4 Kinematika <i>Three Omniwheels</i>	10
BAB III METODE PELAKSANAAN	12
3.1 <i>Survey</i> , Pengumpulan data dan Pengelolaan data	13
3.2 Perancangan Mekanik Robot	13
3.3 Pembuatan Mekanik Robot.....	13

3.4	<i>Assembling</i> Mekanik Robot	13
3.5	Perancangan Elektrik Robot.....	13
3.6	Pembuatan Rangkaian Elektronika	14
3.7	<i>Assembling</i> Elektrik Robot.....	14
3.8	<i>Assembling</i> antara Mekanik dan Elektrik Robot.....	14
3.9	Pengujian antara Mekanik dan Elektrik Robot	14
3.10	Uji Coba keseluruhan Robot	14
3.11	Pembuatan Laporan Proyek Akhir	15
BAB IV PEMBAHASAN.....		16
4.1	Survei, Pengumpulan data dan pengelolaan data.....	16
4.2	Perancangan Mekanik Robot	16
4.3	Pembuatan Mekanik Robot.....	17
4.4	<i>Assembling</i> Mekanik Robot	18
4.5	Perancangan Elektrik Robot.....	18
4.6	Pembuatan Elektrik Robot	19
4.7	<i>Assembling</i> Elektrik Robot.....	20
4.8	<i>Assembling</i> Mekanik dan Elektrik Robot.....	20
4.9	Pengujian Mekanik dan Elektrik Robot	21
4.9.1	Blok Diagram.....	21
4.9.2	Pengujian Linearitas PWM terhadap RPM.....	22
4.9.3	Kontrol PID.....	24
4.10	Instruksi Robot.....	32
4.10.1	Instruksi Mode Biasa	33
4.10.2	Instruksi mode Koordinat	35
4.11	Pemakaian Daya Baterai	37
BAB V KESIMPULAN DAN SARAN.....		38
5.1	Kesimpulan	38
5.2	Saran.....	38
DAFTAR PUSTAKA		39

DAFTAR TABEL

Tabel 2.1 Respon Tuning Parameter PID	7
Tabel 2.2 Perhitungan Parameter	8
Tabel 4.1 Kalibrasi RPM.....	22
Tabel 4.2 Hasil keluaran sinyal PWM ke RPM	23
Tabel 4.3 Penentuan nilai T dan L	24
Tabel 4.4 Perhitungan Parameter PID Metode ZN	25
Tabel 4.5 Hasil Ujicoba Sistem Menggunakan $K_p=0,0038$, $K_i=0,21$, dan $K_d=0,000017$	26
Tabel 4.6 Perbandingan hasil pengujian tanpa beban	27
Tabel 4.7 Perbandingan hasil pengujian tanpa beban	28
Tabel 4.8 Perbandingan hasil pengujian dengan beban	28
Tabel 4.9 dengan menggunakan PID dan Tanpa PID	30
Tabel 4.10 Simulasi perhitungan nilai pulsa dengan nilai ppr encoder	31
Tabel 4.11 Percobaan dengan kecepatan robot terhadap sumbu x 0,5 m/s dan terhadap sumbu y 1 m/s.....	33
Tabel 4.12 Percobaan dengan kecepatan robot terhadap sumbu x 0 m/s dan terhadap sumbu y 1 m/s	34
Tabel 4.13 Percobaan dengan kecepatan robot terhadap sumbu x 1 m/s dan terhadap sumbu y 0 m/s	34
Tabel 4.14 Percobaan dengan kecepatan robot terhadap sumbu x 1 m/s dan terhadap sumbu y 0 m/s	34
Tabel 4.15 Percobaan dikoordinat (1.1).....	35
Tabel 4.16 Percobaan dikoordinat (3.4).....	36
Tabel 4.17 Percobaan dikoordinat (2.3).....	36
Tabel 4.18 Percobaan dikoordinat (1.4).....	36

DAFTAR GAMBAR

Gambar 2.1 Robot Sepak Bola.....	4
Gambar 2.2 Blok diagram PID [4].....	5
Gambar 2.3 Kurva S [5].....	8
Gambar 2.4 Penentuan Parameter L dan T [5].....	8
Gambar 2.5 Blok Diagram Sistem Loop Tertutup [5].....	9
Gambar 2.6 Osilasi Stabil untuk Mencari Nilai Pcr [5].....	9
Gambar 2.7 Kinematik dari Sistem Penggerak Three Omni-Wheels [5].....	10
Gambar 2.8 Kinematik dari sebuah penggerak Three Omni-Wheels [5].....	11
Gambar 3.1 Flowchart Metode Pelaksanaan.....	12
Gambar 4.1 Detail Ukuran Robot.....	16
Gambar 4.2 Rancangan 2 dan 3 dimensi Robot.....	17
Gambar 4.3 Base Robot.....	17
Gambar 4.4 Pemasangan Motor DC.....	18
Gambar 4.5 Pemasangan Roda.....	18
Gambar 4.6 Rancangan Rangkaian Elektrik.....	19
Gambar 4.7 Rangkaian pada PCB.....	19
Gambar 4.8 Assembling Elektrik Robot.....	20
Gambar 4.9 Assembling Mekanik dan Elektrik Robot.....	20
Gambar 4.10 Blok Diagram Kontrol Robot.....	21
Gambar 4.11 Blok Diagram Hardware Robot.....	21
Gambar 4.12 Tampilan aplikasi: (a) Login; (b) Pilihan Mode.....	32
Gambar 4.13 Tampilan aplikasi: (a) Mode Biasa; (b) Mode Koordinat.....	32
Gambar 4.14 Tampilan layar instruksi mode biasa.....	33
Gambar 4.15 Tampilan layar instruksi mode koordinat.....	35

DAFTAR LAMPIRAN

Lampiran 1: Daftar Riwayat Hidup

Lampiran 2: Program

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kontes Robot Indonesia (KRI) adalah ajang kompetisi rancang bangun dan rekayasa dalam bidang robotika yang diselenggarakan oleh Direktorat Kemahasiswaan, Direktorat Jenderal Pembelajaran dan Kemahasiswaan Kementerian Riset, Teknologi, dan Pendidikan Tinggi Robotik Indonesia. Kompetisi ini dibagi menjadi beberapa divisi, yaitu divisi Kontes Robot Abu Indonesia (KRAI), divisi Kontes Robot Seni Tari Indonesia (KRSTI), divisi Kontes Robot Pemadaman Api Indonesia (KRPAI), dan divisi Kontes Robot Sepak Bola Indonesia (KRSBI). Tema dan aturan setiap divisi kontes pada KRI mengacu pada kompetisi internasional. KRAI mengacu pada *ABU Asia-Pacific Robot Contest*. KRPAI mengacu pada *Trinity College International Robot Contest*. Dan KRSBI mengacu pada *RoboCup*. [1]

Pada divisi KRSBI, tipe robot yang dilombakan terbagi menjadi dua, yaitu tipe beroda dan *humanoid*. Yang membedakan antara kedua tipe robot ini adalah penggerak utama robot tersebut. Pada KRSBI tipe *humanoid*, penggerak utama robot untuk berjalan, berlari dan menendang bola adalah kaki sehingga robot *humanoid* ini tampak seperti pemain bola sungguhan. Berbeda halnya dengan KRSBI beroda, dimana penggerak utamanya adalah roda, sedangkan untuk penendang dan penggiring bola digunakan sistem mekanis tambahan. Pada KRSBI, robot harus dapat bermain bola seperti merebut bola, menggiring bola, dan menendang bola. Khusus pada KRSBI beroda, robot tersebut membutuhkan mobilitas yang tepat dan stabil agar pengaturan strateginya lebih mudah. [2]

Saat ini, kontrol pergerakan robot yang digunakan pada tim robot Polmanbabel masih bersifat *open loop*. Artinya, pada saat diberikan instruksi kepada salah satu motor penggerak, maka tidak dapat diketahui secara akurat apakah gerakan atau putaran motor tersebut sudah sesuai atau belum dengan set point yang diinginkan. Sebagai contoh, ketika ingin menentukan kecepatan motor

pada roda, maka parameter yang digunakan hanya berdasarkan sinyal analog dari mikrokontroler atau sinyal *Pulse Width Modulation (PWM)* saja tanpa ada umpan balik dan bersifat statis. Kelemahan dari sistem kontrol seperti ini adalah sulitnya untuk menjaga kestabilan, mengingat apabila sumber tegangan berubah maka output kecepatan juga akan berubah.

Untuk dapat menerapkan strategi yang baik, maka minimal sistem kendali pada masing-masing *actuator* atau motor harus *close loop*. Selain itu, dengan menggunakan sistem *holonomic* maka diperlukan sistem kendali tersendiri yang dapat mengkonversi instruksi dari kontroler utama menjadi instruksi khusus ke masing-masing *actuator*. Dengan adanya sistem yang sudah mampu menjaga kestabilan gerak, maka pengaturan strategi menjadi lebih mudah atau sangat mungkin untuk diterapkan. Oleh karena itu, proyek akhir ini diusulkan dengan harapan dapat membantu tim robot Polmanbabel dalam membuat robot-robotnya agar siap berlaga di kontes robot nasional.

1.2 Rumusan Masalah

1. Bagaimana konstruksi mekanik *Wheeled Holonomic Robot*?
2. Bagaimana menerapkan sistem kontrol PID pada pengaturan kecepatan masing-masing roda?
3. Bagaimana konversi pada kecepatan masing-masing roda menjadi satu kesatuan kecepatan robot dan juga arah pergerakan robot atau sebaliknya?

1.3 Batasan Masalah

1. Permukaan lantai rata atau tidak bergelombang sesuai dengan kondisi ketika kontes robot sepak bola.
2. Dimensi robot tidak melebihi ketentuan yang ada pada aturan kontes robot sepak bola KRSBI beroda.
3. Instruksi yang diberikan kepada robot bersifat *Open-Loop*, kecuali pada masing-masing roda.
4. Beban pada robot tetap.

5. Beban pada masing-masing motor sama.

1.4 Tujuan

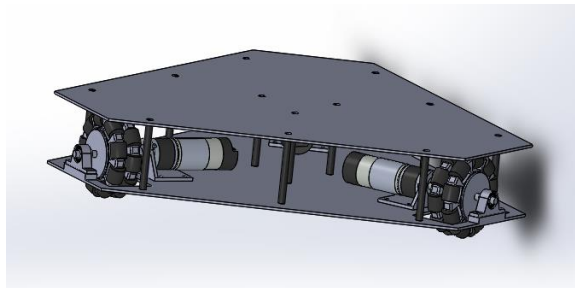
1. Dapat membuat konstruksi mekanik *Wheeled Holonomic Robot*
2. Dapat menerapkan sistem kontrol PID pada pengaturan kecepatan masing-masing roda
3. Dapat menerapkan kinematika robot 3 *omniwheels*

BAB II

DASAR TEORI

2.1 Robot Sepak Bola

Robot Sepak bola terdiri dari 2 tipe yaitu tipe beroda dan *humanoid*. Yang membedakan antara kedua tipe robot ini adalah penggerak utama robot tersebut. Pada KRSBI tipe *humanoid*, penggerak utama robot untuk berjalan, berlari dan menendang bola adalah kaki sehingga robot *humanoid* ini tampak seperti pemain bola sungguhan. Berbeda halnya dengan KRSBI beroda, dimana penggerak utamanya adalah roda, sedangkan untuk penendang dan penggiring bola digunakan sistem mekanis tambahan. Pada KRSBI, robot harus dapat bermain bola seperti merebut bola, menggiring bola, dan menendang bola. Khusus pada KRSBI beroda, robot tersebut membutuhkan mobilitas yang tepat dan stabil agar pengaturan strateginya lebih mudah. [2]



Gambar 2.1 Robot Sepak Bola

2.2 *Holonomic Robot*

Sistem gerak *holonomic* merupakan sebuah sistem yang jumlah derajat kebebasannya sama dengan jumlah koordinat yang dibutuhkan untuk menyatakan konfigurasi dari sistem tersebut. Di dalam bidang *mobile robot*, istilah *holonomic* diterapkan terhadap sebuah robot tanpa memperhatikan kesatuan bentuk fisik yang membentuk mekanisme yang sebenarnya. *Mobile robot* dengan sistem gerak yang memiliki tiga derajat kebebasan pada suatu bidang dapat dikatakan *holonomic mobile robot*. Sistem gerak *holonomic* dapat membantu dalam merancang gerakan

robot karena tidak ada keterbatasan arah robot dapat bergerak. Berdasarkan sifat dasarnya, robot *non-holonomic* memiliki derajat kebebasan lebih sedikit dibandingkan dengan robot *holonomic*. [3]

Ciri-ciri robot *holonomic* adalah sebagai berikut:

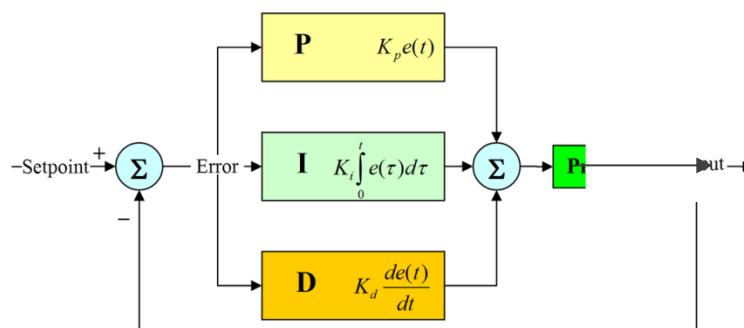
1. Konfigurasi robot dideskripsikan oleh tiga dimensi.
2. Robot memiliki tiga DOF tanpa *singularities*. *Singularities* adalah titik pada bidang yang tidak terjangkau.
3. Robot dapat bergerak memutar secara bebas pada absis x, ordinat y, dan orientasi Θ .
4. Robot dapat berakselerasi pada absis x, ordinat y, dan orientasi Θ .

Ciri-ciri robot *non-holonomic* adalah sebagai berikut:

1. Konfigurasi robot dideskripsikan oleh lebih dari tiga dimensi.
2. Robot memiliki dua DOF atau tiga DOF dengan *singularities* [3].

2.3 Kontrol PID (*Propositional Integral Derivatif*)

Kontrol *Proportional, Integral, Derivative* (PID) adalah sistem pengendali yang umum digunakan di industri maupun militer. Sekitar 90% dari peralatan industri menggunakan pengendali PID karena mudah digunakan dan paling sederhana. Gambar 2.2 menunjukkan blok diagram kontrol PID yang umum digunakan di industri. Kontrol *Proportional* (P), *Integral* (I), dan *Derivatif* (D) dapat digunakan bersamaan secara paralel ataupun digunakan terpisah dengan tidak menggunakan salah satu komponen P, I atau D. [4]



Gambar 2.2 Blok diagram PID [4]

Persamaan nilai keluaran dari kontrol PID, dirumuskan sebagai:

$$u(t) = K_P e(t) + K_I \int_0^t e(t)dt + K_D \frac{d}{dt} e(t) \dots\dots\dots (2.1)$$

Persamaan (1) menjelaskan bahwa nilai keluaran $u(t)$, merupakan jumlah dari *gain proportional* (K_p), *gain integral* (K_i), dan *gain derivative* (K_d) yang masing-masing dipengaruhi oleh error (e) dalam selang waktu (t) tertentu. [4]

2.3.1 *Proportional Control*

Kontrol proporsional memiliki *output* yang sebanding atau proporsional dengan besar sinyal kesalahan (selisih antara nilai yang diinginkan dengan nilai aktualnya, *error*). Setiap perubahan pada sinyal *input* menyebabkan sistem langsung mengeluarkan sinyal keluar sebesar konstanta pengalinya. Persamaan kontrol proporsional dirumuskan sebagai:

$$u(t) = K_P e(t) \dots\dots\dots (2.2)$$

K_p merupakan *gain proportional*, e adalah *error*, dan u merupakan nilai keluaran relatif terhadap waktu (t). [4]

2.3.2 *Integral Control*

Pengontrol *integral* berfungsi untuk menghilangkan *steady-state error* menjadi nol. Jika sebuah plant tidak mempunyai unsur *integrator* ($1/s$), pengontrol proporsional tidak mampu menjamin output sistem akan tepat sesuai respon yang diinginkan, sehingga dibutuhkan pengontrol *integral*. Persamaan kontrol *integral* dirumuskan sebagai:

$$u(t) = K_I \int_0^t e(t)dt \dots\dots\dots (2.3)$$

K_i merupakan *gain integral*, e adalah *error*, dan u merupakan nilai keluaran relatif terhadap waktu (t). [4]

2.3.3 *Derivative Control*

Besar *output* dari pengontrol *derivatif* memiliki sifat seperti operasi diferensial pada umumnya. Pengontrol *derivatif* menggunakan kecepatan perubahan sinyal kesalahan sebagai parameter pengontrol. Apabila tidak ada

perubahan sinyal *error*, maka *output* dari kontrol *derivatif* tidak akan berubah. Persamaan kontrol *derivatif* dirumuskan sebagai:

$$u(t) = K_D \frac{d}{dt} e(t) \dots\dots\dots (2.4)$$

Kd merupakan *gain derivatif*, e adalah *error*, dan u merupakan nilai keluaran relatif terhadap waktu (t). [4]

2.3.4 Metode Tuning parameter PID

Cara melakukan tuning parameter-parameter PID bisa dilakukan dengan melihat tabel karakteristik respon parameter PID. Dengan menganalisa respon yang dihasilkan, nilai-nilai Kp, Ki, dan Kd bisa diubah-ubah sesuai keluaran yang diinginkan. Tabel karakteristik respon parameter PID ditunjukkan pada tabel 2.1. [5]

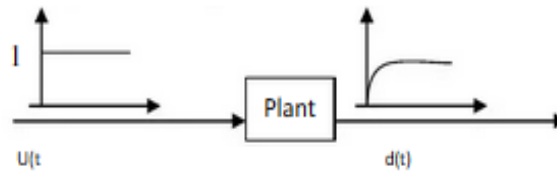
Tabel 2.1 Karakteristik Respon Parameter PID [5]

Parameter	<i>Rise Time</i>	<i>Overshoot</i>	<i>Settling Time</i>	<i>Steady-State</i>
Kp	Berkurang	Bertambah	<i>Minor Change</i>	Berkurang
Ki	Berkurang	Bertambah	Bertambah	Menghilangkan
Kd	<i>Minor Change</i>	Berkurang	Berkurang	<i>Minor Change</i>

Salah satu metoda mencari parameter-parameter PID adalah dengan metode *Ziegler-Nichols*. Metode ini memiliki 2 tipe yaitu metode *Ziegler Nichols* Tipe Osilasi dan Tipe Kurva Reaksi. [5]

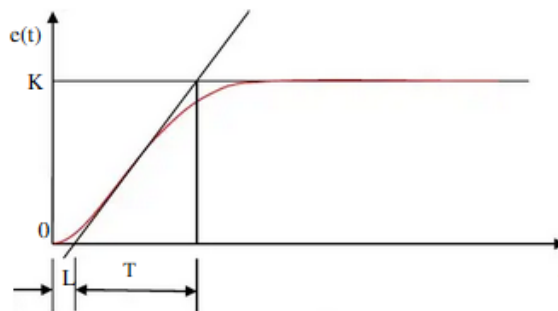
2.3.4.1 Metode *Ziegler Nichols* Tipe Kurva Reaksi

Pada metode *Ziegler Nichols* Tipe kurva reaksi dilakukan pada sistem yang bersifat terbuka (*open loop*) atau sistem yang tidak memiliki umpan balik. Plant yang akan dikendalikan tersebut diberi *input* step dan responnya akan dianalisa dengan metode *Ziegler Nichols* tipe kurva reaksi dimana perhitungannya akan menghasilkan parameter-parameter PID. [5]



Gambar 2.3 Kurva S [5]

Nilai PID diperoleh dari hasil percobaan dengan masukan ulit step, hasilnya akan membentuk kurva huruf S, seperti pada gambar 2.3. Kurva berbentuk S memiliki karakteristik dengan 2 buah konstanta, yaitu waktu tunda L dan time konstan T. [5]



Gambar 2.4 Penentuan Parameter L dan T [5]

Kedua parameter tersebut diperoleh dengan menggambar garis tangensial pada titik infleksi kurva S, seperti pada gambar 2.4. Garis tangensial tersebut akan berpotongan dengan garis time axis dan garis $c(t) = K$. berdasarkan parameter tersebut, didapatkan cara untuk menghitung parameter K_p , T_i , dan T_d . [5]

Tabel 2.2 Perhitungan Parameter [5]

PID Type	K_p	$T_i = K_p / K_i$	$T_d = K_d / K_p$
P	$\frac{T}{L \times U}$	∞	0
PI	$0.9 \frac{T}{L \times U}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L \times U}$	$2L$	$0.5L$

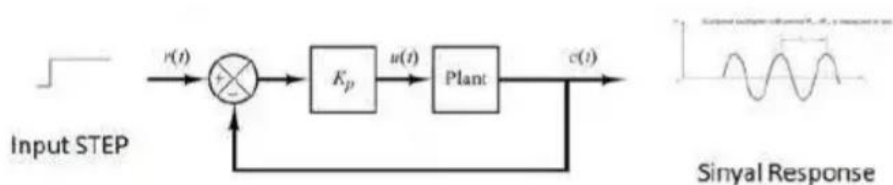
Apabila harga T_i dan T_d sudah diketahui, maka konstanta K_i dan K_d dapat ditentukan:

$$K_i = K_p/T_i \dots\dots\dots (2.5)$$

$$K_d = K_p \times T_d \dots\dots\dots (2.6)$$

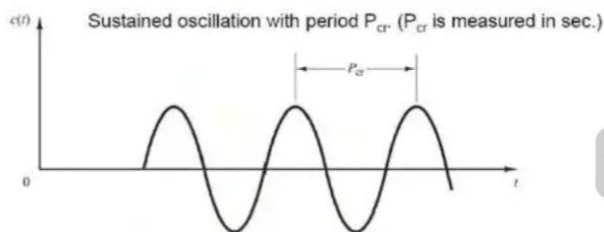
2.3.4.2 Metode Ziegler Nichols Tipe Osilasi

Pada metode *Ziegler Nichols* tipe Osilasi ini proses penentuan parameter PID dihitung menggunakan respon yang diatur menggunakan slider *gain* sehingga respon berosilasi. Metode ini diaplikasikan pada sistem *loop* tertutup (*close loop*) atau dengan umpan balik, (K_i & K_d di *set 0/off*), seperti gambar di bawah ini:



Gambar 2.5 Blok Diagram Sistem Loop Tertutup [5]

Pada metode kedua ini, percobaan dilakukan dengan menggunakan *proportional band*. Nilai K_p dinaikkan dari 0 hingga tercapai nilai K_p yang menghasilkan osilasi yang konsisten. Nilai slider *gain* ini disebut sebagai *critical gain* (K_{cr}). Jika harga K_p terlalu kecil, sinyal *output* akan teredam mencapai nilai titik keseimbangan setelah ada gangguan, Sebaliknya, jika harga K_p terlalu besar, osilasinya akan tidak stabil dan membesar, seperti gambar di bawah ini:



Gambar 2.6 Osilasi Stabil untuk Mencari Nilai P_{cr} [5]

Pada metode ini nilai penguatan K_p disebut K_{cr} . Nilai K_{cr} didapatkan dari tegangan output pada *setpoint* dibagi tegangan *input*, dan nilai P_{cr} didapatkan dari

mengurangkan kedua tinggi gelombang tersebut dan mengalikan dengan nilai Kcr. [5]

$$Kcr = \frac{Output}{Input} \dots\dots\dots(2.7)$$

$$Pcr = (X2-X1) \times Kcr \dots\dots\dots(2.8)$$

Berdasarkan parameter tersebut, didapatkan cara untuk menghitung parameter Kp, Ti dan Td. [5]

$$Kp = 0,6 \times Kcr \dots\dots\dots(2.9)$$

$$Ti = 0,5 \times Pcr \dots\dots\dots(2.10)$$

$$Td = 0,125 \times Pcr \dots\dots\dots(2.11)$$

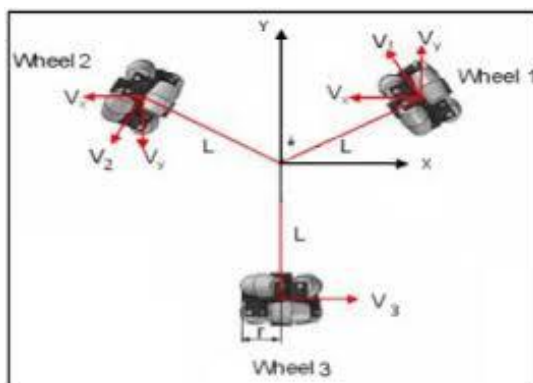
Apabila harga Ti dan Td sudah diketahui, maka konstanta Ki dan Kd dapat ditentukan. [5]

$$Ki = 2 \times \frac{Kp}{Ti} \dots\dots\dots(2.12)$$

$$Kd = Kp \times Kd \dots\dots\dots(2.13)$$

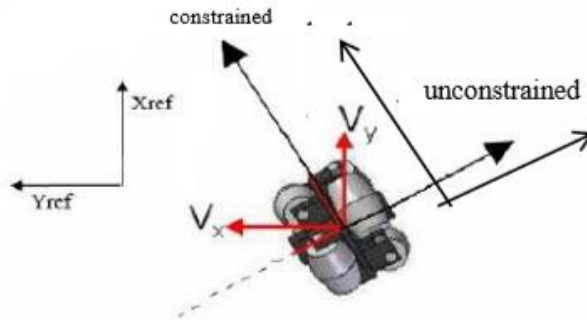
2.4 Kinematika Three Omniwheels

Untuk memahami dasar dari *omni-directional robot* yang dikembangkan, kita dapat meneliti sistem kinematika dari robot tersebut. Jika kita menginginkan pergerakan atau perpindahan robot terhadap posisi awalnya, kita harus mengetahui variabel-variabel yang berkaitan satu dengan yang lainnya yang dapat kita kontrol seperti posisi gerak dari robot dan kecepatan pada masing-masing roda. Perhitungan kinematik pada robot digunakan untuk menentukan perubahan posisi antar koordinat global dan posisi koordinat internal dari robot itu sendiri. [5]



Gambar 2.7 Kinematika dari Sistem Penggerak *Three Omni-Wheels* [5]

Untuk arah gerak masing-masing roda seperti gambar berikut.



Gambar 2.8 Kinematik dari sebuah penggerak *Three Omni-Wheels* [5]

Keterangan:

$X_B, Y_B,$ = Sistem koordinat kartesian.

$V_i, (1,2,3)$ = Arah putar roda *omni*.

L = Jarak pusat robot dengan titik *center* roda.

δ = Sudut roda *omni* terhadap sumbu Y = 30° . [5]

Dari gambar 1 dan 2 persamaan kinematik dari sistem bisa didapatkan.

Persamaan yang digunakan adalah.

$$V_1 = -\cos 60^\circ \dot{x} + \sin 60^\circ \dot{y} + L\dot{\theta} \dots \dots \dots (2.14)$$

$$V_2 = \cos 60^\circ \dot{x} + \sin 60^\circ \dot{y} + L\dot{\theta} \dots \dots \dots (2.15)$$

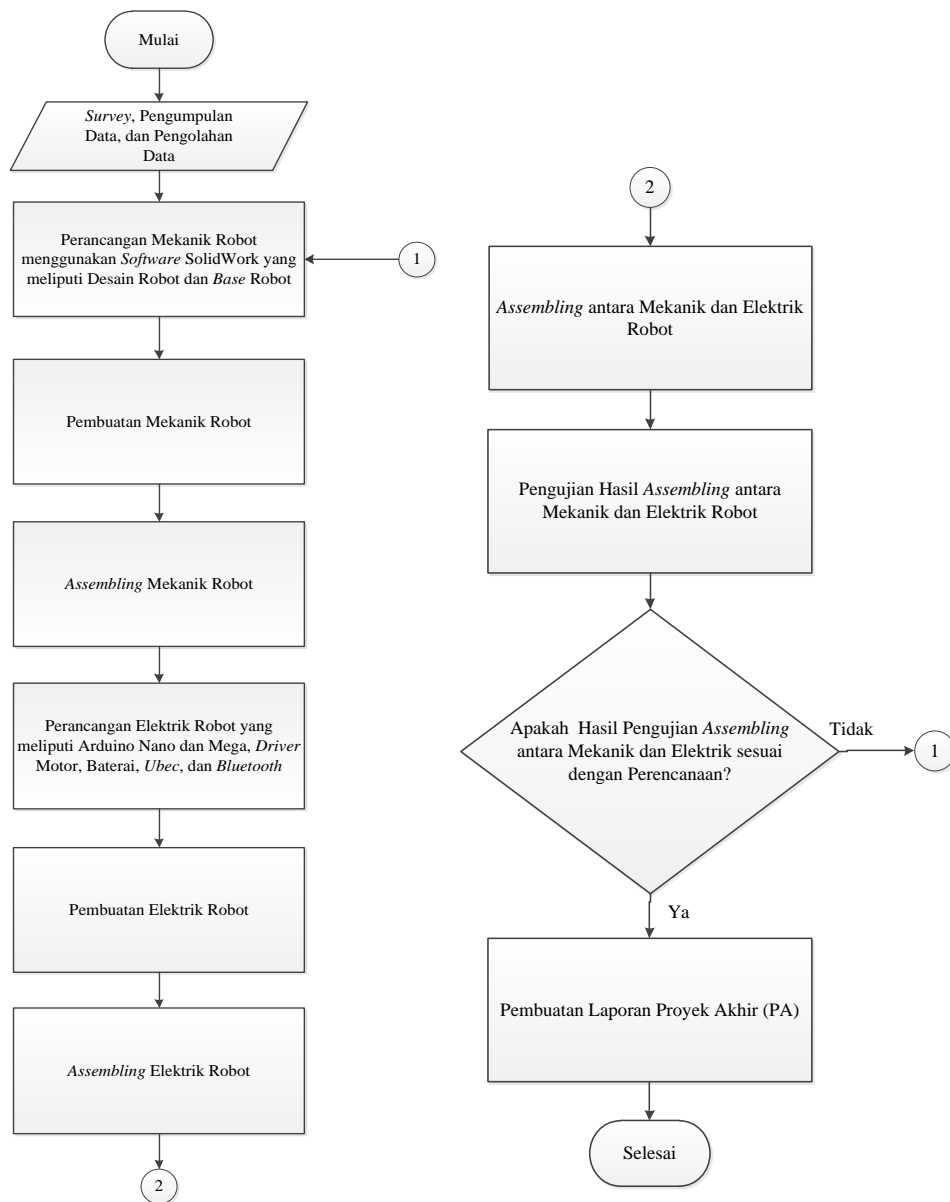
$$V_3 = \dot{x} + L\dot{\theta} \dots \dots \dots (2.16)$$

Untuk mendapatkan kecepatan dari masing-masing roda omni adalah dengan cara mengalihkan kecepatan dari motor dengan jari-jari dari roda *omni*. Dalam hal ini roda *omni* dirangkai secara simetris dengan perbedaan sudut antar roda sebesar 120° dan masing-masing roda memiliki sudut δ sebesar 30° . [5]

BAB III

METODE PELAKSANAAN

Pada Bab ini menjelaskan mengenai metode pelaksanaan yang penulis terapkan dalam perencanaan, pelaksanaan, dan penyelesaian proyek akhir serta penyusunan laporan ini. *Flowchart* metode pelaksanaan dapat dilihat pada gambar 3.1.



Gambar 3.1 *Flowchart* Metode Pelaksanaan

3.1 Survei, Pengumpulan data dan Pengelolaan data

Langkah awal sebelum merancang desain dan membuat robot, terlebih dahulu pengumpulan data referensi tentang robot sepak bola beroda. Adanya referensi ialah dapat membantu menganalisis ataupun memberi kejelasan tentang apa yang tidak kita ketahui pada tahap-tahap pengerjaan tugas akhir ini. Referensi dicari dari internet baik dari youtube, jurnal ataupun dari tulisan tim robot Polman Babel yang pernah mengikuti ajang kontes robot.

3.2 Perancangan Mekanik Robot

Pada tahap yang pertama penulis membuat rancangan robot menggunakan *software solidWork*. Rancangan tersebut meliputi bagian-bagian penting dari robot seperti *base* dan desain dari robot. Dimana rancangan tersebut tidak melebihi ukuran yang telah ditetapkan.

3.3 Pembuatan Mekanik Robot

Tahap selanjutnya adalah pembuatan konstruksi robot. Panduan pembuatan konstruksi robot ini adalah gambar kerja yang telah dibuat menurut rancangannya. Proses pengerjaannya dilakukan di bengkel mekanik Polman Babel, ruangan Ostatik atau di lab Elektronika.

3.4 Assembling Mekanik Robot

Pada tahap *assembling* bagian-bagian robot yang sudah selesai dibuat seperti dudukan robot akan dirakit dengan komponen-komponen mekanik lainnya seperti roda *omni* dan Motor Dc. Tahapan *assembling* terdiri dari perakitan dari roda ke motor dan dari motor ke base robot.

3.5 Perancangan Elektrik Robot

Pada tahap ini, penulis mulai mengumpulkan komponen yang akan digunakan untuk pembuatan robot, selain itu pada tahap ini juga penulis merancang rangkaian apa saja yang akan dibuat pada robot.

3.6 Pembuatan Rangkaian Elektronika

Selanjutnya pembuatan rangkaian elektronika mengikuti pada rancangan yang sudah dibuat. Proses pengerjaannya dilakukan di Lab Elektronika dan diruangan Ostatik.

3.7 *Assembling* Elektrik Robot

Assembling Elektrik robot merupakan proses perakitan komponen-komponen elektronika menjadi suatu rangkaian. Komponen-komponen elektronika yang digunakan yaitu Arduino Mega, Arduino Nano, Baterai Lipo, *Driver* Motor, Ubec, dan Modul Bluetooth.

3.8 *Assembling* antara Mekanik dan Elektrik Robot

Assembling mekanik dan elektrik robot merupakan proses penggabungan antara kontruksi mekanik yang sudah dibuat dan rangkaian elektrik yang sudah dirangkai. Penggabungan antara mekanik dan elektrik robot akan menghasilkan sebuah robot tetapi belum memiliki sistem kontrol.

3.9 Pengujian antara Mekanik dan Elektrik Robot

Pada tahap pengujian ini akan menggunakan program untuk membuat kontrol robot. Ada beberapa program yang digunakan untuk pengujian robot seperti program PWM ke RPM, Pengontrolan roda, Tuning, dan program utama. Program yang dibuat akan membentuk suatu sistem kontrol pada robot.

3.10 Uji Coba keseluruhan Robot

Pada tahap ini bertujuan untuk menguji robot secara keseluruhan baik mekanik maupun elektrik sudah sesuai dengan rancangan awal atau belum. Pengujian pada tahap ini yaitu apakah robot sudah bisa bergerak sesuai instruksi yang diberikan atau masih menghasilkan *error* yang melewati tuntutan. Apabila Robot sudah sesuai dengan rancangan dan tuntutan maka akan melanjutkan pada proses pembuatan Laporan Akhir, namun apabila robot belum sesuai dengan tuntutan maka dilakukan pengecekan dimulai dari point 1 dan seterusnya.

3.11 Pembuatan Laporan Proyek Akhir

Pembuatan Proyek Akhir merupakan tahap terakhir setelah selesai pembuatan Proyek akhir. Dengan adanya Laporan Poyek Akhir maka kita dapat mengetahui tahap-tahap dalam pembuatan robot dan juga dapat membantu sebagai referensi untuk ke depannya.

BAB IV

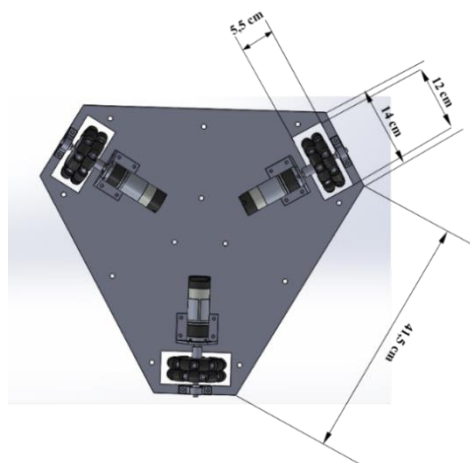
PEMBAHASAN

4.1 Survei, Pengumpulan data dan pengelolaan data

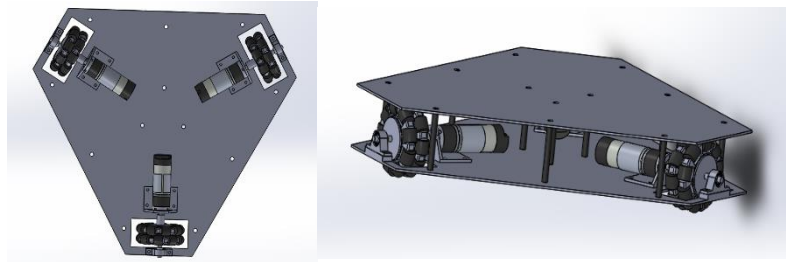
Survei, Pengumpulan data dan pengelolaan data dilakukan untuk mempermudah perancangan dan pembuatan robot seperti pemilihan komponen, referensi dari Internet, dari tulisan tim robot Polman Babel, serta referensi dari Dosen Pembimbing. Komponen mekanik yang digunakan berupa Plat aluminium, As aluminium, Plat dan siku, sedangkan komponen elektrik yang digunakan berupa Arduino Mega [7], Arduino Nano [8], *Driver Motor* [9], Ubec, Baterai Lipo dan Module Bluetooth.

4.2 Perancangan Mekanik Robot

Sebelum melakukan pembuatan robot tentunya kita harus melakukan perancangan robot terlebih dahulu. Tujuan dari perancangan tersebut ialah mempermudah untuk mencocokkan alat dengan pembuatan konstruksi robot. Perancangan Mekanik Robot dapat dilakukan dengan mendesain pada kertas atau buku terlebih dahulu kemudian diterapkan pada *Software SolidWork*. Dimana rancangan tersebut tidak melebihi ukuran yang telah ditetapkan. Berikut detail ukuran dan rancangan 2 dan 3 dimensi robot dapat dilihat pada Gambar 4.1 dan 4.2.



Gambar 4.1 Detail Ukuran Robot



Gambar 4.2 Rancangan 2 dan 3 dimensi Robot

4.3 Pembuatan Mekanik Robot

Pembuatan mekanik robot yang pertama yaitu pembuatan *base* robot, sebelum membuat desain *base* pada robot ada beberapa hal yang harus dipertimbangkan yaitu dalam hal memilih model dari *base* robot agar *base* robot dapat memenuhi beberapa *point* seperti mudah dalam pembuatannya dan agar pergerakan robot bisa fleksibel.

Jika proses desain selesai hal pertama yang dilakukan untuk proses pembuatan *base* robot yaitu dengan memotong plat aluminium sampai berbentuk segitiga, setelah itu *marking* untuk dudukan motor. Setelah proses *marking* selesai dilanjutkan dengan pemotongan untuk jarak pemasangan motor dc.

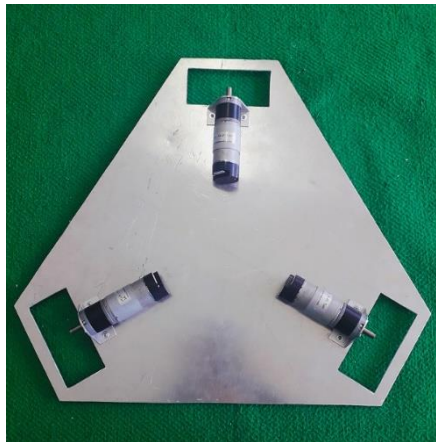
Base inilah yang nantinya akan menjadi dasar peletakan komponen-komponen elektrik seperti Arduino, *Driver* Motor dan dudukan baterai. *Base* robot dapat dilihat pada Gambar 4.3.



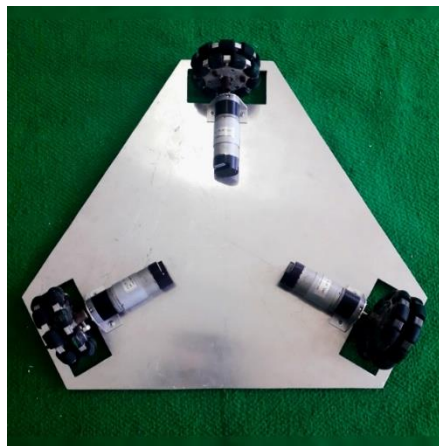
Gambar 4.3 Base Robot

4.4 *Assembling* Mekanik Robot

Setelah pembuatan *base* selesai maka selanjutnya ke tahap *assembling* mekanik robot. *Assembling* mekanik robot merupakan penyusunan komponen mekanik robot menjadi satu kesatuan. *Assembling* mekanik robot disusun atau dipasang dengan kencang sehingga tidak mudah goyang ataupun tidak merubah kesetabilan gerak robot. Penyusunan konstruksi robot dimulai dengan pemasangan dudukan Motor DC, pemasangan Motor DC dan pemasangan Roda.



Gambar 4.4 Pemasangan Motor DC

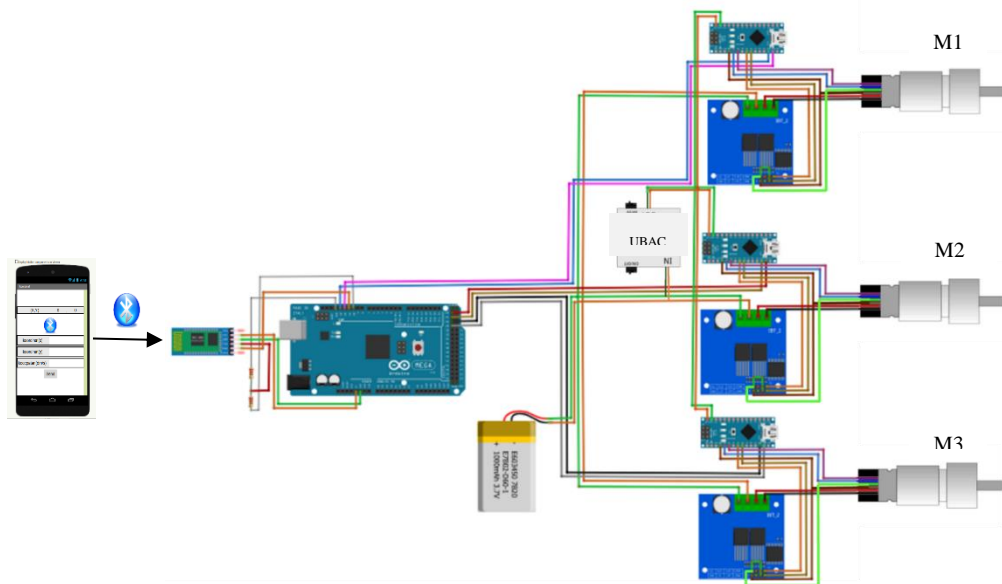


Gambar 4.5 Pemasangan Roda

4.5 Perancangan Elektrik Robot

Perancangan Elektrik Robot dilakukan untuk mempermudah dalam melakukan pembuatan rangkaian elektronika. Tujuan dilakukan perancangan terlebih dahulu agar tidak terjadi kesalahan dalam pengkoneksian antar komponen

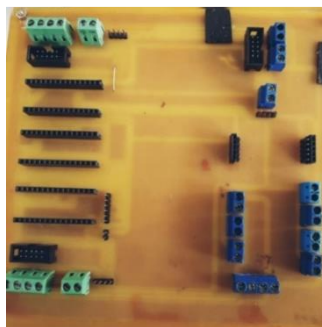
saat pembuatan rangkaian Elektronika. Berikut racangan elektrik Robot dapat dilihat pada Gambar 4.6.



Gambar 4.6 Rancangan Rangkaian Elektrik

4.6 Pembuatan Elektrik Robot

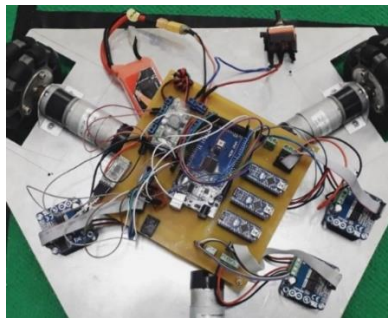
Pembuatan Elektrik Robot merupakan proses peletakan *port-port* untuk peletakan komponen elektrik pada *Project Circuit Board* (PCB) sebelum proses *assembling*. Peletakan *port-port* yaitu seperti penyolderan kaki pin pada PCB, pemasangan terminal blok, kemudian penyusunan dan pemasangan komponen pada PCB. Tujuan pembuatan elektrik robot ini bertujuan sebagaiudukan komponen seperti Arduino dan juga digunakan untuk sambungan antar *port*. Berikut rangkaian pada PCB yang dibuat dapat dilihat pada Gambar 4.7.



Gambar 4.7 Rangkaian pada PCB

4.7 *Assembling* Elektrik Robot

Setelah menentukan komponen yang dibutuhkan untuk pembuatan robot maka selanjutnya ke tahap perakitan rangkaian elektronika. *Assembling* Elektrik Robot merupakan pengkoneksian antar komponen elektronika agar menjadi suatu rangkaian yang berfungsi untuk menggerakkan robot. Perakitan rangkaian elektronika dirangkai dengan kebutuhan yang diperlukan yaitu mengkoneksikan antar komponen tanpa adanya *Short* pada rangkaian, karena apabila terjadi *Short* pada rangkaian akan berakibat fatal atau kemungkinan akan terjadi kerusakan fungsi pada komponen yang terkena *Short*. Perakitan rangkaian elektronika juga harus menyesuaikan pada program *software*nya seperti pin berapa yang digunakan sebagai *input* dan *output* pada komponen yang memiliki fungsi *input* dan *output*. Perakitan Rangkaian Elektrik dapat dilihat pada Gambar 4.8.



Gambar 4.8 *Assembling* Elektrik Robot

4.8 *Assembling* Mekanik dan Elektrik Robot

Assembling mekanik dan elektrik robot merupakan proses penggabungan antara konstruksi mekanik yang sudah dibuat dan rangkaian elektrik yang sudah dirangkai. Berikut *assembling* mekanik dan elektrik robot dapat dilihat pada Gambar 4.9.



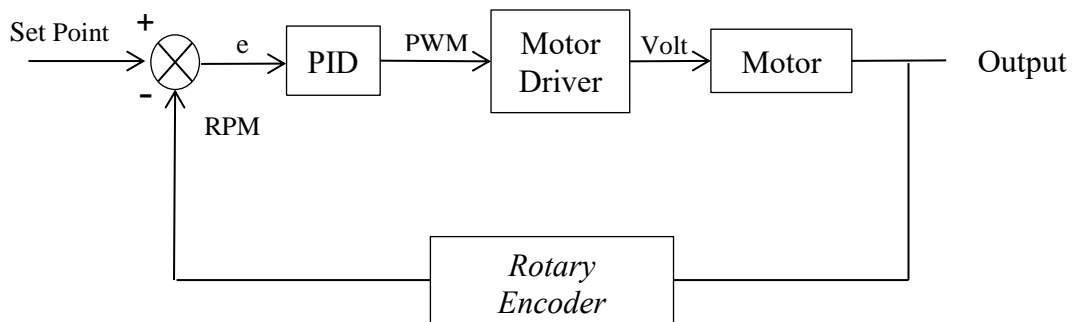
Gambar 4.9 *Assembling* Mekanik dan Elektrik Robot

4.9 Pengujian Mekanik dan Elektrik Robot

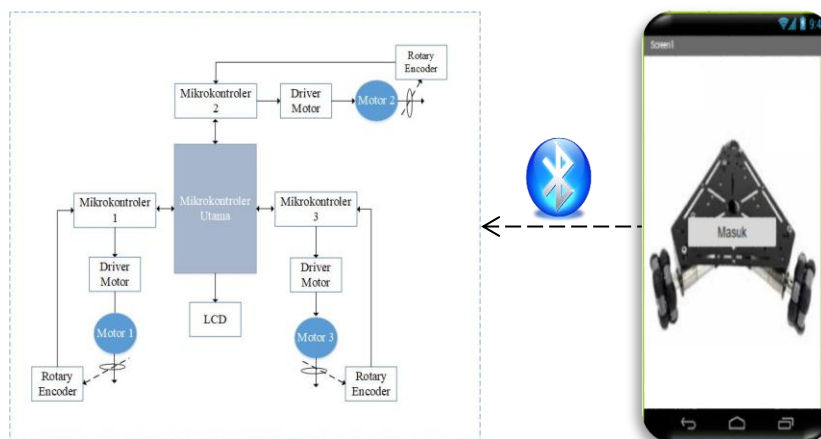
Pada tahap pengujian ini akan menggunakan program untuk membuat kontrol robot. Ada beberapa program yang digunakan untuk pengujian robot seperti program konfigurasi PWM ke RPM, Pengontrolan roda, Tuning, dan program utama. Program yang dibuat akan membentuk suatu sistem kontrol pada robot.

4.9.1 Blok Diagram

Pada pengujian robot tentunya sangat dibutuhkan blok diagram untuk menentukan tahap-tahap sebuah sistem kontrol pada robot. Berikut blok diagram kontrol dan blok diagram *hardware* robot dapat dilihat pada Gambar 4.10 dan Gambar 4.11.



Gambar 4.10 Blok Diagram Kontrol Robot



Gambar 4.11 Blok Diagram *Hardware* Robot

4.9.2 Pengujian Linearitas PWM terhadap RPM

Sebelum melakukan pengujian linieritas PWM terhadap RPM terlebih dahulu melakukan kalibrasi RPM. Berikut adalah Tabel kalibrasi RPM menggunakan *Tachometer* dengan pembacaan *rotary encoder*.

Tabel 4.1 Kalibrasi RPM

<i>Tachometer</i>	<i>Rotary Encoder</i>	<i>Error (%)</i>
126,90	127,98	0,8
126,90	127,98	0,8
126,90	127,98	0,8
126,90	127,98	0,8
126,90	127,98	0,8
58,30	59,81	0,8
58,30	59,81	0,8
58,30	59,81	0,8
58,30	59,81	0,8
58,30	59,81	0,8
200,70	201,56	0,4
200,70	201,56	0,4
200,70	201,56	0,4
200,70	201,56	0,4
Rata-rata <i>error</i> :		0,6

Pada table 4.1 dapat dilihat persentase *error* dari pembacaan dengan *rotary encoder* dengan pembacaan dengan dengan *tachometer* adalah 0,6 % Sehingga penggunaan *rotary encoder* ini masih dapat digunakan.

Pada gambar 4.1 block diagram kontrol, dapat dilihat bahwa *Driver Motor* menerima sinyal PWM untuk menggerakkan Motor Dc, sedangkan *set point* dari kontrol ini berupa nilai RPM. Maka pada tahap ini akan dilakukan pengujian linearitas terhadap sinyal PWM terhadap RPM. Berikut Tabel data perubahan dari sinyal PWM ke RPM:

Tabel 4.2 Hasil dan grafik linearitas PWM terhadap RPM

PWM	RPM	GRAFIK
1	0	
15	19,64	
30	41,07	
45	64,29	
60	87,5	
75	108,93	
90	132,14	
105	153,57	
120	173,21	
135	198,21	
150	221,43	
165	224,64	
180	267,86	
195	287,5	
210	308,93	
225	332,14	
240	353,57	
255	373, 21	

Pada grafik linearitas PWM terhadap RPM diatas, dapat dilakukan peninjauan terhadap persamaan garis lurus seperti berikut.

$$y = mx+b \dots\dots\dots(4.1)$$

$$m=\Delta y/\Delta x \dots\dots\dots(4.2)$$

Sehingga didapatkan nilai m dari tabel 4.2 adalah 1,45 dan b=-5,46. Dari nilai *gradient* (m) dan konstanta (b) yang telah didapatkan, maka persamaan RPM terhadap PWM dapat ditulis sebagai berikut:

$$RPM=m(PWM)+b \text{ atau } RPM=1,45(PMW) -5,46 \dots\dots\dots(4.3)$$

Persamaan ini yang digunakan untuk mensinkronisasikan nilai PWM terhadap RPM pada pengontrolan PID.

4.9.3 Kontrol PID

Kontrol PID digunakan untuk mengontrol kecepatan putaran motor DC dengan bantuan sensor *rotary encoder* yang terpasang pada motor DC. Pada proyek akhir ini pengontrolan PID digunakan untuk mengontrol tiap-tiap roda, sehingga dibutuhkan juga 3 kontrol PID untuk mengontrol 3 motor DC.

Metode yang digunakan untuk mendapatkan konstanta PID itu sendiri adalah metode *Ziegler Nichols*. Seperti penjelasan pada BAB II, metode *Ziegler Nichols* memiliki dua metode, yaitu osilasi dan kurva reaksi. Pada proyek akhir ini metode tuning yang digunakan adalah metode kurva reaksi *Ziegler Nichols*.

4.9.3.1 Prosedur FOTD (*First Order Time Delay*)

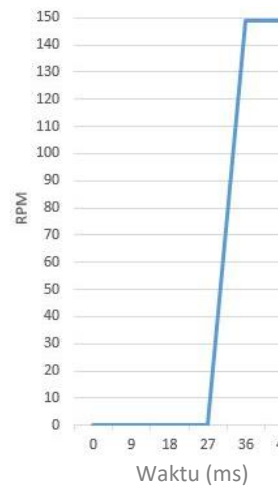
Hal pertama yang dilakukan pada metode kurva reaksi *Ziegler Nichols* ini adalah menentukan nilai unit step. Unit step adalah *setpoint* RPM yang digunakan untuk melihat reaksi. Pada proyek akhir ini nilai unit step yang digunakan adalah 150 rpm. Namun, pada hal ini nilai RPM tidak linear dengan nilai PWM maka, pada perhitungan untuk mencari nilai K_p (seperti pembahasan pada landasan teori pada BAB II), nilai U yang digunakan pada perhitungan adalah nilai PWM yang diberikan untuk menggerakkan motor DC dengan 150 rpm. Sehingga nilai U yang digunakan nanti adalah 101 PWM.

Seperti pembahasan pada BAB II mengenai tuning dengan metode kurva reaksi yang diperhatikan adalah reaksi motor DC dari awal hingga kondisi stabilnya saja, sehingga didapat grafik seperti pada table 4.3. Berikut adalah grafik reaksi motor DC terhadap waktu dengan *sampling time* 9 ms.

Tabel 4.3 Reaksi motor DC terhadap waktu

Waktu (ms)	RPM	GRAFIK
0	0	
9	0	
18	0	
27	0	
36	148.80	
45	148.80	

54	150.65
63	148.80
72	148.80
81	148.80
90	148.80
99	150.65
108	148.80
117	148.80
126	148.80
135	148.80
144	150.65
153	150.65
162	148.80



Nilai T=9 ms, Nilai L=27 ms

Dengan menggunakan rujukan persamaan pada tabel *Zigeler Nichols* (ZN) pada bab II maka diperoleh tabel hasil perhitungan parameter seperti tampak pada Tabel 4.4 berikut:

Tabel 4.4 Perhitungan Parameter PID Metode ZN

PID Type	K_p	$T_i = K_p / K_i$	$T_d = K_d / K_p$
P	0,0011	∞	0
PI	0,00097	0,09	0
PID	0,0012	0,018	0,0135

Berdasarkan Tabel 4.4, maka konstanta pada masing-masing kontroler yaitu K_p , K_i , dan K_d ditetapkan. Selanjutnya adalah proses ujicoba sistem dengan menerapkan konstanta tersebut. Adapun hasil dari ujicoba sistem tersebut dapat dilihat pada Tabel 4.5, dimana sistem sudah berjalan cukup baik tanpa *overshoot*, tetapi sedikit lebih lama untuk mencapai *settling time*.

Tabel 4.5 Hasil Ujicoba Sistem Menggunakan $K_p=0,0038$, $K_i=0,21$, dan $K_d=0,000017$

Hasil	Grafik
Overshoot = 0 %	
Rise Time = 0,82 s	
Settling Time = 0,90 s	
Steady State = 151 rpm	

Berdasarkan hasil ujicoba sistem pada Tabel 4.5, maka diperlukan perbaikan respon terutama untuk mengurangi waktu naik dan waktu tenang (*settling time*) dengan cara ditala (*adjust*) secara perlahan. Harapannya adalah untuk mendapatkan respon terbaik atau nilai konstanta PID terbaik. Rujukan peran masing-masing kontroler pada PID adalah tabel 2.1 pada Bab II. Setelah dilakukan penalaan secara perlahan, diperoleh hasil atau respon yang baru yang dapat dilihat pada tabel 4.6 tentang perbandingan hasil dari pengujian program kontrol PID dengan set point sebesar 150 rpm.

Pada Tabel 4.6 ditunjukkan sebanyak tiga (3) buah hasil penalaan kontroler PID beserta responnya masing-masing. Dari ketiga percobaan tersebut dapat disimpulkan bahwa data percobaan ke-3 adalah yang paling baik. Hal ini dikarenakan respon yang diberikan sangat baik dengan *rise time* dan *settling time* yang paling kecil sebesar 0,12 s dan 0,29 s walaupun *overshoot* dan *steady state error* nya tidak mengalami perubahan.

Tabel 4.6 Perbandingan hasil pengujian tanpa beban

No	Nilai	Hasil	Grafik
1.	$K_p = 0,0181$ $K_i = 0,431$ $K_d = 0,00019$	Overshoot = 3,17 % Rise Time = 0,14 s Settling Time = 0,64 s Steady State = 151 rpm	
2.	$K_p = 0,3$ $K_i = 0,5$ $K_d = 0,005$	Overshoot = 3,17 % Rise Time = 0,46 s Settling Time = 0,71 s Steady State = 151 rpm	
3.	$K_p = 0,15$ $K_i = 0,8$ $K_d = 0,2$	Overshoot = 3,17 % Rise Time = 0,12 s Settling Time = 0,29 s Steady State = 151 rpm	

Pengujian Sistem Kontrol Dengan Beban

Pada pengujian ini, robot diletakkan dilantai dengan pembebanan pada robot standar yang digunakan ketika robot beraktivitas. Pengujian ini bertujuan untuk melihat bagaimana respon kecepatan roda, apakah berubah atau tidak dengan pengujian sebelumnya (pengujian tanpa beban).

Dengan menggunakan konstanta PID yang paling baik pada percobaan sebelumnya atau percobaan ke-3 pada Tabel 4.6 diperoleh bahwa respon yang

dihasilkan berubah dan tidak sesuai yang diharapkan. Adapun respon tersebut dapat dilihat pada Tabel 4.7 berikut, dimana terjadi perubahan yang signifikan pada *rise time*.

Tabel 4.7 Perbandingan hasil pengujian tanpa beban

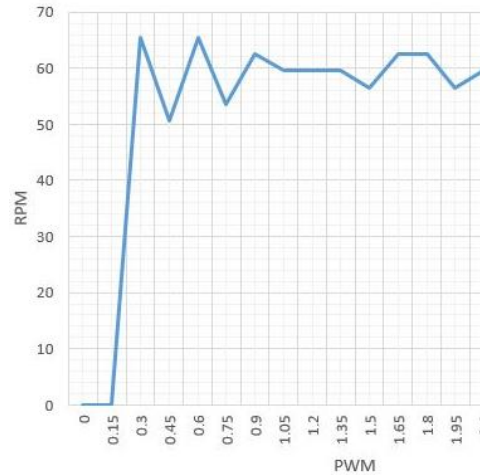
Hasil	Grafik
Overshoot = 28,22 % Rise Time = 0,3 s Settling Time = 0,85 s Steady State = 62 rpm	

Berdasarkan Tabel 4.7, maka konstanta PID yang ada harus ditala kembali untuk mendapatkan respon yang lebih baik. Tabel 4.8 berikut adalah hasil penalaan kembali pada kontroler PID dengan set point sebesar 60 rpm. Percobaan atau penalaan yang dilakukan adalah sebanyak tiga kali, dimana pada percobaan yang ke-3 diperoleh hasil atau respon yang paling baik dengan *rise time* sebesar 0,12ms dan *settling time* sebesar 0,38ms. Namun, ada sedikit sekali kenaikan pada *overshoot* dan *steady state error*.

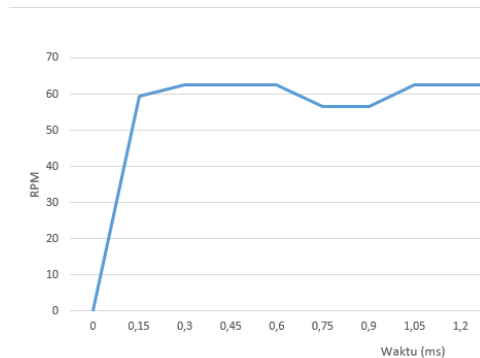
Tabel 4.8 Perbandingan hasil pengujian dengan beban

No	Nilai	Hasil	Grafik
1.	$K_p = 0,18$ $K_i = 1,05$ $K_d = 0,4$	Overshoot = 13,67 % Rise Time = 0,207 s Settling Time = 0,76 s Steady State = 62 rpm	

2. $K_p = 0,7$ Overshoot = 9,667 %
 $K_i = 0,8$ Rise Time = 0,07 s
 $K_d = 0,4$ Settling Time = 0,67 s
Steady State = 62 rpm



3. $K_p = 0,51$ Overshoot = 3,33 %
 $K_i = 0,89$ Rise Time = 0,12 s
 $K_d = 0,5$ Settling Time = 0.14 s
Steady State = 62 rpm



Melihat dari hasil percobaan yang ditunjukkan pada Tabel 4.6 dan Tabel 4.8, maka ada perbedaan yang cukup berpengaruh ketika roda pada robot diuji dalam kondisi bebas atau tanpa beban dengan ketika roda diberi beban berupa bodi robot tersebut. Dengan adanya data ini, maka konstanta PID tidak serta merta bisa digunakan pada kondisi yang berbeda. Pada Proyek Akhir ini, data atau konstanta PID yang digunakan adalah konstanta PID ketika roda diberi beban.

Perbandingan Respon Antara Sistem Dengan Kontrol PID dan Tanpa PID

Percobaan ini bertujuan untuk melihat sejauh mana dampak sistem ketika menerapkan kontrol PID. Dengan adanya percobaan ini bisa memperkuat atau meyakinkan bahwa kontrol PID itu dibutuhkan atau tidak pada sistem saat ini. Pada percobaan ini, sistem kontrol tanpa PID menggunakan teknik pengendalian *error* secara konvensional atau logika percabangan sederhana. adapun hasil dari percobaan tersebut dapat dilihat pada Tabel 4.9.

Pada tabel 4.9 dapat dilihat bahwa ada perbedaan respon yang sangat signifikan untuk parameter penting seperti *overshoot* dan *settling time*. Dengan adanya data perbandingan ini tampak jelas bahwa kontroler PID sangat berpengaruh dalam menghasilkan respon yang lebih baik. Dimana selisih dari keduanya adalah *overshoot* 20 % dan *settling time* 1,81 s.

Tabel 4.9 dengan menggunakan PID dan Tanpa PID

No	PID	Hasil	Grafik
1.	Tanpa PID	<p>Overshoot = 23,3 %</p> <p>Rise Time = 0,12 s</p> <p>Settling Time = 1,95 s</p> <p>Steady State = 62 rpm</p>	
2.	Dengan PID	<p>Overshoot = 3,33 %</p> <p>Rise Time = 0,12 s</p> <p>Settling Time = 0,14 s</p> <p>Steady State = 62 rpm</p>	

Steady State Error Tidak Nol

Pada proyek akhir ini didapatkan bahwa nilai *steady state error* itu tidak hilang atau masih ada (tidak nol). Secara teori, dengan adanya PID khususnya bagian integral (I) unsur *steady state error* harusnya bisa dieliminasi, akan tetapi kenyataannya tidak demikian. Hal ini setelah diamati disebabkan oleh spesifikasi *rotary encoder* yang memiliki jumlah pulsa per rotasi sebesar 134 ppr. Pengaruhnya adalah terhadap resolusi kecepatan putaran roda, dimana dengan *time sampling* sebesar 0,15 s maka perubahan terkecil setiap 1 pulsa *encoder* adalah 2,97 rpm.

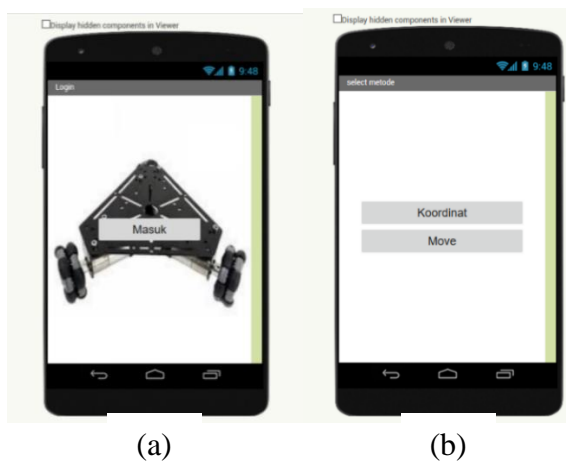
Atau dengan kata lain, dengan *time sampling* terkecil yang bisa digunakan menggunakan Arduino diperoleh kecepatan putaran roda terkecil adalah 2,97rpm. Ini menunjukkan bahwa kelipatan terkecil dalam penambahan kecepatan putaran roda adalah 2,97 rpm bukan 1 rpm. Hal ini mengakibatkan sulitnya untuk mencapai kondisi *set point* kecuali *set point* yang digunakan adalah kelipatan 2,97 rpm. Tabel 4.10 berikut memperlihatkan bagaimana pengaruh jumlah ppr terhadap kecepatan putaran roda:

Tabel 4.10 Simulasi perhitungan nilai pulsa dengan nilai ppr *encoder*

Pulses	Putaran	RPM	Selisih
0	0	0	0
1	0.00744	3	2.976192
2	0.014881	6	2.976192
3	0.022321	9	2.976192
4	0.029762	12	2.976192
5	0.037202	15	2.976192
6	0.044643	18	2.976192
7	0.052083	21	2.976192
8	0.059524	24	2.976192
9	0.066964	27	2.976192
10	0.074405	30	2.976192
11	0.081845	33	2.976192
12	0.089286	36	2.976192
13	0.096726	39	2.976192
14	0.104167	42	2.976192
15	0.111607	45	2.976192
16	0.119048	48	2.976192
17	0.126488	51	2.976192
18	0.133929	54	2.976192
19	0.141369	57	2.976192
20	0.14881	60	2.976192
21	0.15625	63	2.976192
22	0.16369	65	2.976192
23	0.171131	68	2.976192
24	0.178571	71	2.976192
25	0.186012	74	2.976192

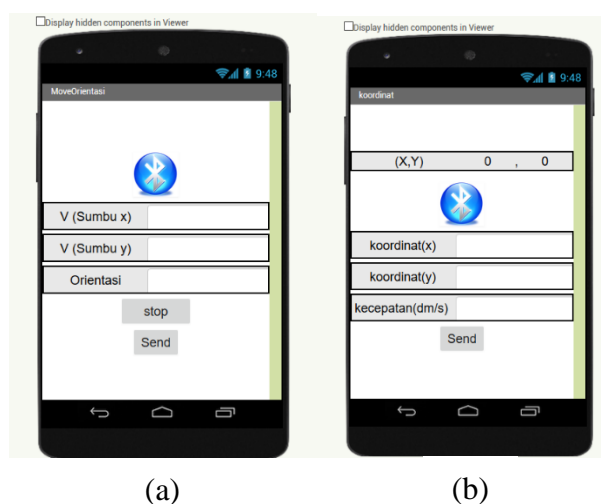
4.10 Instruksi Robot

Instruksi Robot merupakan perintah yang harus dijalankan oleh robot seperti perintah maju, mundur, kiri, kanan, berputar dll. Instruksi robot menggunakan sebuah aplikasi yang dibuat sendiri dari aplikasi MIT AppInventor. Pemberian instruksi dibagi menjadi dua mode, yaitu mode biasa dan koordinat. Berikut merupakan tampilan dari aplikasi yang dimaksud.



Gambar 4.12 Tampilan aplikasi: (a) Login; (b) Pilihan Mode

Pada proyek akhir ini, intruksi yang diberikan kepada robot terdiri dari dua mode, yaitu mode biasa dan mode koordinat.



Gambar 4.13 Tampilan aplikasi: (a) Mode Biasa; (b) Mode Koordinat

4.10.1 Instruksi Mode Biasa

Instruksi mode biasa merupakan memberikan instruksi kepada robot untuk bergerak dengan instruksi berupa kecepatan terhadap sumbu x dan atau sumbu y.



Gambar 4.14 Tampilan layar instruksi mode biasa

Gambar 4.14 merupakan tampilan dari aplikasi yang memberikan intruksi kepada robot atau aplikasi untuk menjalankan robot. Berikut perbandingan percobaan pengujian robot biasa:

Tabel 4.11 Percobaan dengan kecepatan robot terhadap sumbu x 0,5 m/s dan terhadap sumbu y 1 m/s

Jarak sumbu X (m)	Jarak sumbu Y (m)	Waktu Sebenarnya (s)	Waktu Tempuh (s)	Error %
1	2	2,23	2,20	1,3
1	2	2,23	2,20	1,3
1	2	2,23	2,18	2,2
Rata-rata <i>Error</i> : 1,6 %				

Tabel 4.12 Percobaan dengan kecepatan robot terhadap sumbu x 0 m/s dan terhadap sumbu y 1 m/s

Jarak sumbu X (m)	Jarak sumbu Y (m)	Waktu Sebenarnya (s)	Waktu Tempuh (s)	Error %
0	1	1	1,03	3
0	1	1	1	0
0	1	1	1,02	2
Rata-rata <i>Error</i> : 1,6%				

Tabel 4.13 Percobaan dengan kecepatan robot terhadap sumbu x 1 m/s dan terhadap sumbu y 1 m/s

Jarak sumbu X (m)	Jarak sumbu Y (m)	Waktu Sebenarnya (s)	Waktu Tempuh (s)	Error %
1	1	1,4	1,38	1,4
1	1	1,4	1,35	3,5
1	1	1,4	1,40	0
Rata-rata <i>Error</i> : 1,6 %				

Tabel 4.14 Percobaan dengan kecepatan robot terhadap sumbu x 1 m/s dan terhadap sumbu y 0 m/s

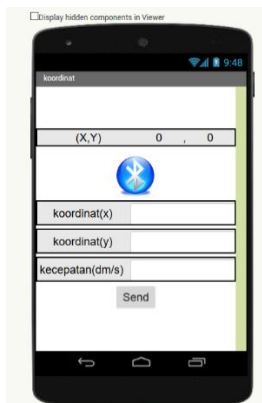
Jarak sumbu X (m)	Jarak sumbu Y (m)	Waktu Sebenarnya (s)	Waktu Tempuh (s)	Error%
1	0	1	1	0
1	0	1	0,99	1
1	0	1	0,98	2
Rata-rata <i>Error</i> :1%				

Pada tabel 4.11 sampai tabel 4.14 diatas merupakan beberapa contoh pergerakan yang dapat dilakukan oleh robot. Pada tabel diatas juga dapat disimpulkan bahwa robot dapat bergerak ke segala arah tanpa merubah arah hadapnya, dengan instruksi kecepatan terhadap sumbu x dan atau sumbu y. Dari

tabel hasil percobaan diatas juga dapat dilihat bahwa rata-rata kesalahan kecepatan robot adalah $\pm 1,45\%$.

4.10.2 Instruksi mode Koordinat

Instruksi Koordinat merupakan pemberian instruksi kepada robot berupa kecepatan robot dan tujuan robot yang bergerak pada sebuah koordinat, dimana koordinat memiliki grib 1 meter x 1 meter.



Gambar 4.15 Tampilan layar instruksi mode koordinat

Gambar 4.15 merupakan tampilan dari aplikasi yang memberikan intruksi kepada robot atau aplikasi untuk menjalankan robot. Pada pengujian robot dengan mode intruksi biasa telah diketahui bahwa rata-rata kesalahan kecepatan robot adalah $\pm 1,45\%$, sehingga pada pengujian dengan instruksi koordinat ini diambil dengan kecepatan yang sama, yaitu 0,5 m/s. Berikut tabel percobaan robot dengan koordinat:

Tabel 4.15 Percobaan dikoordinat (1.1)

Arah Hadap	Sumbu X			Sumbu Y		
	Jarak (mm)	Jarak Tempuh (mm)	Error%	Jarak (mm)	Jarak Tempuh (mm)	Error %
0%	1000	1010	1	1000	1000	0
0%,	1000	1000	0	1000	1000	0
0%	1000	1000	0	1000	1000	0
0%	1000	1010	1	1000	990	1
Rata-rata Error: 0,50%				Rata-rata Error: 0,33%		

Tabel 4.16 Percobaan dikoordinat (3.4)

Arah Hadap	Sumbu X			Sumbu Y		
	Jarak (mm)	Jarak Tempuh (mm)	Error %	Jarak (mm)	Jarak Tempuh (mm)	Error %
5%	3000	3040	1,3	4000	4000	0
5%,	3000	3045	1,5	4000	4040	1
5%	3000	3030	1	4000	4040	1
5%	3000	3030	1	4000	4040	1
Rata-rata <i>Error</i> :1,20%				Rata-rata <i>Error</i> : 0,75%		

Tabel 4.17 Percobaan dikoordinat (2.3)

Arah Hadap	Sumbu X			Sumbu Y		
	Jarak (mm)	Jarak Tempuh (mm)	Error %	Jarak (mm)	Jarak Tempuh (mm)	Error %
5%	2000	2020	1	3000	3000	0
0%,	2000	2010	0,5	3000	3030	1
5%	2000	2020	1	3000	3030	1
0%	2000	2010	0,5	3000	3010	0,3
Rata-rata <i>Error</i> : 0,75%				Rata-rata <i>Error</i> : 0,57%		

Tabel 4.18 Percobaan dikoordinat (1.4)

Arah Hadap	Sumbu X			Sumbu Y		
	Jarak (mm)	Jarak Tempuh (mm)	Error %	Jarak (mm)	Jarak Tempuh (mm)	Error %
0%	1000	1000	0	4000	4040	1
0%,	1000	1010	1	4000	4040	1
5%	1000	1010	1	4000	4000	0
0%	1000	1000	0	4000	4040	1
Rata-rata <i>Error</i> : 0,5%				Rata-rata <i>Error</i> : 0,75%		

Pada tabel 4.15 sampai tabel 4.18 diatas merupakan beberapa contoh koordinat yang dicapai robot. Pada tabel diatas juga dapat disimpulkan bahwa robot

dapat bergerak ke segala arah tanpa merubah arah hadapnya, dengan rata-rata perubahan arah hadap adalah $\pm 5\%$ dan rata-rata kesalahan jarak tempuh adalah $\pm 0,6\%$.

4.11 Pemakaian Daya Baterai

Pada proyek akhir ini baterai yang digunakan adalah baterai lipo dengan kapasitas tegangan output 12 volt DC dan arus 4200 mAh. Arus yang terbaca saat pengukuran ketika robot dijalankan adalah 0,45 A.

$$\text{Waktu Pemakaian Baterai} = \frac{4,2 \text{ Ah}}{0,45 \text{ A}} = 9,3 \text{ jam}$$

Sehingga dapat diketahui bahwa daya tahan baterai yang digunakan pada robot adalah sekitar 9,3 jam.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun Kesimpulan dari Proyek Akhir ini adalah:

1. Kinematika robot 3 roda omni diperlukan dalam kendali pergerakan robot yang bersifat holonomic.
2. Metode tuning Ziegler Nichols digunakan untuk mendapatkan nilai awal konstanta PID (sebagai starting point). Artinya, nilai tersebut masih bisa di-adjust untuk mendapatkan hasil yang lebih baik. Berdasarkan hasil pengujian, ada perbedaan konstanta PID pada roda yang dibeban dan tidak dibeban. Atas dasar ini, pada proyek akhir ini, konstanta PID pada roda yang dibebankan menjadi rujukan dalam kontrol roda. Perbedaan nilai tersebut adalah $k_p=0,55$, $k_i=0,85$, dan $k_d=0,55$ untuk roda yang dibebankan, sedangkan tanpa beban nilai $k_p=0,01$; $k_i=0,8$; dan $k_d=0,2$.
3. Dengan spesifikasi encoder sebesar 134 ppr berpengaruh terhadap resolusi kecepatan putaran roda, dimana dengan time sampling sebesar 0,15 s maka perubahan terkecil setiap 1 pulsa encoder adalah 2,97 rpm.
4. Dengan kinematika robot 3 omniwheels ini terlihat rata-rata kesalahan kecepatan robot sebesar 1,45% dan kesalahan rata-rata arah hadap robot adalah 5%.

5.2 Saran

Jika dilakukan pengembangan lebih lanjut, dibutuhkan nilai ppr encoder yang lebih besar agar pembacaan atau pun sinkronisasi lebih presisi.

DAFTAR PUSTAKA

- [1] K. official, "Kontes Robot Indonesia," [Online]. Available: <https://kontesrobotindonesia.id/kontes-robot.html>. [Accessed 06 Mei 2019].
- [2] R. A. Rochmanto, "Implementasi Robot There Omni-Directional menggunakan kontrol PID pada Kontes Robot Abu Indonesia(KRAI)," p. 1, 2014.
- [3] G. A. Mulyady, "Implementasi sistem gerak Holonomic pada robot KRSBI beroda 2017," *jurnal teknik dan ilmu komputer*, vol. 07, p. 11, 2018.
- [4] A. W. d. H. N. Danu Wisnu, "Perancangan Sistem Kontrol PID untuk Pengendali Sumbu Azimuth Turret pada Turret-gun Kaliber 20mm," *jurnal teknik ITS*, vol. 5, no. 2, p. 1, 2016.
- [5] K. Simamora, "Desain Kendali Pada Plant Debit Air dengan metode Ziegler-Nichols dan Cohen-Coon menggunakan Matlab dan Arduino," Politeknik Negeri Bandung, Bandung, 2015.
- [6] M. M. B. B. S. B. D. Iwan Kurniawan Wibowo, "penentuan Posisi Robot menggunakan Odometry Omniwheel," no. ISBN, pp. 11 - 12, 2018.
- [7] A. Official, "introduction to the arduino board," Arduino, 2019. [Online]. Available: <http://www.mantech.co.za/datasheets/products/A000047.pdf>. [Accessed 11 september 2019].
- [8] A. Official, "introduction to the arduino board," Arduino, 2019. [Online]. Available: <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>. [Accessed 11 September 2019].
- [9] D. M. Official, "introduction to the driver motor," driver motor, 2019. [Online]. Available: [https://www.handsontec.com/dataspecs/module/BTS7960%20Motor%200 Driver .pdf](https://www.handsontec.com/dataspecs/module/BTS7960%20Motor%200%20Driver.pdf). [Accessed 11 September 2019].

- [10] E. R. W. A. M. Angger Dimas Bayu Sadewo, "Perancangan Pengendali Rumah menggunakan Smartphone Android dengan konektivitas bluetooth," vol. 1, no. 5, 2017.
- [11] M. E. I. N. M. D. P. Yosua D. Widiarto, "Sistem penggerak Robot Beroda Vacuum Cleaner berbasis Mini Computer Raspberry pi," *jurnal teknik elektro dan komputer*, vol. 7, p. 26, 2018.
- [12] A. P. A. H. Dedy Junaidy, "Sistem kerja Robot Penjaga Gawang (Keeper) Divisi KRSBI beroda Tim ROTA," no. ISBN, p. 31, 2018.
- [13] A. E. Wahyudi, "Perancangan dan Pembuatan alat ukur jarak digital berbasis Arduino menggunakan Rotary Encoder," p. 4, 2015.
- [14] M. L. Akbar Iskandar, "Sistem keamanan pintu berbasis Arduino Mega," *Jurnal Informatika*, vol. 3, p. 100, 2017.
- [15] A. W. Setyamwamn, "Kendali Pergerakan Robot Sepak bola dengan PID (Proposional Integral Derivatif) Kontrol," Fakultas Teknologi dan Informatika Institut Bisnis dan Informatika Stikom Surabaya, Surabaya, 2017.
- [16] A. Raikhani, "Studi Pengaruh Pembebanan Pada Motor DC Penguatan Shunt Terhadap arus Stator," *jurnal@trisula LP2M undar*, vol. 1, no. ISSN, pp. 39-40, 2015.

LAMPIRAN

LAMPIRAN 1
DAFTAR RIWAYAT HIDUP

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Meta Saputri
Tempat & Tanggal Lahir : Belinyu, 21 Juli 1998
Alamat Rumah : Kp. Telang Luar,
Ds. Gunung Muda
Kec. Belinyu, Kab. Bangka,
Prov. Kep. Bangka Belitung
Hp : 082218269625
Email : metasaputri21@gmail.com
Jenis Kelamin : Perempuan
Agama : Islam



2. Riwayat Pendidikan

SD Negeri 22 Belinyu Lulus 2010
SMP Negeri 2 Belinyu Lulus 2013
SMK YPN Belinyu Lulus 2016

3. Pendidikan Non Formal

Pendidikan Sistem Ganda di Kantor Desa Gunung Muda
Praktik Kerja Lapangan di PT. Jembo Cable Company Tbk

Sungailiat, 30 Agustus 2019



Penulis

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Suganda
Tempat & Tanggal Lahir : Kelapa, 30 Januari 1999
Alamat Rumah : Ds. kelapa, Kec. Kelapa,
Kab. Bangka Barat,
Prov. Kep. Bangka Belitung
Hp : 081363834507
Email : sugandaklp@gmail.com
Jenis Kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

SD Negeri 2 Kelapa Lulus 2010
SMP Negeri 1 Kelapa Lulus 2013
SMA Negeri 1 Kelapa Lulus 2016

Sungailiat, 30 Agustus 2019



Penulis

LAMPIRAN 2
PROGRAM

PROGRAM KESELURUHAN

```
//bluetooth to mega
#include <SoftwareSerial.h>
SoftwareSerial mySerialbm(10,11); //rx tx
String dataIn;
String dt[10];
int i;
boolean parsing=false;

//mega to nano
#include <SoftwareSerial.h>
SoftwareSerial mySerialmn(12,13); // RX, TX motor1
#include <SoftwareSerial.h>
SoftwareSerial mySerialmn2(22,24); // RX, TX motor2
#include <SoftwareSerial.h>
SoftwareSerial mySerialmn3(26,28); // RX, TX motor3

//umum
float vx,vy,v1,v2,v3,o,s,s1,s2,s3,rpm1,rpm2,rpm3;
float v,t,n,wak;
int
darihp1,darihp2,darihp3,jarak1,jarak2,jarak3,ksudut;
int a=1,x,y;
double sudut,sudutoke,nsudut;
#define pb1 7
#define pb2 8

void setup()
{
```

```

pinMode(pb1, INPUT);
pinMode(pb2, INPUT);
Serial.begin(9600);
    //mega to nano
mySerialmn.begin(9600);
mySerialmn2.begin(9600);
mySerialmn3.begin(9600);

//bluetooth to mega
mySerialbm.begin(9600);
dataIn="";

}

void loop()
{
    //pemilihan intruksi
    if(digitalRead(pb1)==HIGH) //change to koordinat
    {
        a=1;
    }
    if(digitalRead(pb2)==HIGH) //change to biasa
    {
        a=0;
    }
    //bluetooth to mega
    if(mySerialbm.available())
    {
        delay(1);
        char inChar=mySerialbm.read();
        dataIn+=inChar;
    }
}

```

```

        if (inChar == ' ')
        {
            parsing = true;
        }
    }
    if(parsing)
    {
        parsingData();
        mySerialmn.println(darihp1);
        mySerialmn.println(jarak1);
        mySerialmn.println(ksudut);
        mySerialmn2.println(darihp2);
        mySerialmn2.println(jarak2);
        mySerialmn2.println(ksudut);
        mySerialmn3.println(darihp3);
        mySerialmn3.println(jarak3);
        mySerialmn3.println(ksudut);
        parsing=false;
        dataIn="";
    }
}
void parsingData()
{
    int j=0;

    //kirim data yang telah diterima sebelumnya
    Serial.print("data masuk : ");
    Serial.print(dataIn);
    Serial.print("\n");

    //inisialisasi variabel, (reset isi variabel)

```

```

dt[j]="";
//proses parsing data
for(i=1;i<dataIn.length();i++)
{
    //pengecekan tiap karakter dengan karakter (#) dan
(,)
    if ((dataIn[i] == '#') || (dataIn[i] == ','))
    {
        //increment variabel j, digunakan untuk merubah
index array penampung
        j++;
        dt[j]=""; //inisialisasi variabel array dt[j]
    }
    else
    {
        //proses tampung data saat pengecekan karakter
selesai.
        dt[j] = dt[j] + dataIn[i];
    }
}
//data hasil parsing
if(a==0) //biasa
{
    vx=dt[0].toInt();
    vy=dt[1].toInt();
    o=dt[2].toInt();
    s=dt[3].toInt();
}
if(a==1) //koordinat
{
    x=dt[0].toInt();

```

```

y=dt[1].toInt();
v=dt[2].toInt();
n=(x*x)+(y*y);
t=sqrt(n);
sudut=(acos(x/t)*180/3.14);
if(sudut>=60.02)
{
    sudutoke=sudut-60.02;
}
if(sudut<60.02)
{
    sudutoke=60.02-sudut;
}
wak=t*10/v;
vx=v*x/t;
vy=v*y/t;
}
//kinematik
v1=((0.5*vx)+(0.866*vy)+(0.247*o))/10;
v2=(-0.5*vx)+(0.866*vy)+(-0.247*o))/10;
v3= ((vx)+(-0.247*o))/10;
rpm1=v1*191.08;
rpm2=v2*191.08;
rpm3=v3*191.08;
nsudut=sin(sudutoke*0.017453295199433);
ksudut=nsudut*100;
s1=v1*wak;
s2=v2*wak;
s3=v3*wak;
darihp1=(float) rpm1*100;
darihp2=(float) rpm2*100;

```



```
darihp3=(float) rpm3*100;
jarak1=(float) s1*100;
jarak2=(float) s2*100;
jarak3=(float) s3*100;

Serial.print("t = ");
Serial.print(t);
Serial.print("\t hp1 = ");
Serial.print(darihp2);
Serial.print("\t j1 = ");
Serial.print(jarak2);
Serial.print("\t n = ");
Serial.print(n);
Serial.print("\t vx = ");
Serial.print(vx);
Serial.print("\t vy = ");
Serial.print(vy);
Serial.print("\t v1 = ");
Serial.print(v1);
Serial.print("\t v2 = ");
Serial.print(v2);
Serial.print("\t v3 = ");
Serial.println(v3);
Serial.print("\t s1 = ");
Serial.print(s1);
Serial.print("\t s2 = ");
Serial.print(s2);
Serial.print("\t s3 = ");
Serial.println(s3);
}
```

```

/*invers trigonometri */

float asin(float c){
    float out;
    out=
((c+(c*c*c)/6+(3*c*c*c*c*c)/40+(5*c*c*c*c*c*c*c*c)/112+
(35*c*c*c*c*c*c*c*c*c*c)/1152
+(c*c*c*c*c*c*c*c*c*c*c*c*c*c*0.022)+

(c*c*c*c*c*c*c*c*c*c*c*c*c*c*.0173)+(c*c*c*c*c*c*c*c*c*c*c*c*c*
c*c*c*c*c*c*.0139)+

(c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*0.0115)+(c*c*c*c*c*c*c*
*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*c*0.01)
));

//asin
if(c>=.96 && c<.97){out=1.287+(3.82*(c-.96)); }
if(c>=.97 && c<.98){out=(1.325+4.5*(c-.97));} //
arcsin
if(c>=.98 && c<.99){out=(1.37+6*(c-.98));}
if(c>=.99 && c<=1){out=(1.43+14*(c-.99));}
return out;}

float acos(float c)
{float out;
out=asin(sqrt(1-c*c));
return out;}

float atan(float c)
{float out;

```

```
out=asin(c/(sqrt(1+c*c)));  
return out;}
```