

**ROBOT KESEIMBANGAN PADA SEBUAH BOLA
MENGUNAKAN SISTEM KENDALI PID**

PROYEK AKHIR

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



Diusulkan oleh :

Muhammad Fauzan NIRM: 1051815

Muhammad Ferdiansyah NIRM: 1051816

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2022**

LEMBAR PENGESAHAN

JUDUL PROYEK AKHIR
ROBOT KESEIMBANGAN PADA SEBUAH BOLA
MENGGUNAKAN SISTEM KENDALI PID

Oleh:

Muhammad Fauzan/1051815

Muhammad Ferdiansyah/1051816

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1



Muhammad Iqbal Nugraha, M.Eng

Pembimbing 2



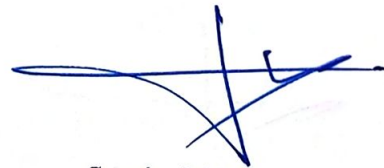
Indra Dwisaputra, M.T

Penguji 1



Ocsirendi, M.T

Penguji 2



Surojo, M.T

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1 : Muhammad Fauzan NIRM : 1051815

Nama Mahasiswa 2 : Muhammad Ferdiansyah NIRM : 1051816

Dengan Judul : Robot Keseimbangan Pada Sebuah Bola Menggunakan
Sistem Kendali PID

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, 14 Februari 2022

Nama Mahasiswa

Tanda Tangan

1. Muhammad Fauzan

..........

2. Muhammad Ferdiansyah

..........

ABSTRAK

Robot keseimbangan bola adalah salah satu robot penyeimbang dinamis yang harus dikontrol secara aktif untuk menstabilkannya tetap seimbang tegak lurus 0° diatas bola pada permukaan tanah atau lantai. Tujuan dari penelitian ini ialah mengimplementasikan sistem kendali PID secara lengkap, yaitu dengan menambahkan konstanta integral (Ki) pada robot keseimbangan bola yang diharapkan mampu mengeliminasi steady-state error dan melakukan perbandingan dari beberapa metode tuning sistem kendali PID untuk mencapai hasil yang optimal pada robot keseimbangan bola. Metode pelaksanaan meliputi pengerjaan desain konstruksi, menentukan flowchart kerja, penentuan kinematika dan penentuan sistem kendali. Sistem kendali yang digunakan ialah sistem kendali PID. Pengolahan data keseluruhan diproses oleh mikrokontroler Arduino Mega2560 yang menerima pembacaan sudut dari sensor MPU6050 sebagai masukan dan aktuasi pada motor stepper sebagai keluaran. Pengujian yang dilakukan diantaranya pengujian sensor, motor, arah putaran motor, respon robot, serta implementasi sistem kendali PID pada robot. Dari berbagai metode tuning yang diuji pada robot, metode manual adjustment CHR Overshoot 20% memberikan waktu jatuhnya robot sedikit lebih lama yaitu 1193ms, dibandingkan metode lainnya sebelum dilakukan manual adjustment yaitu Ziegler-Nichlos step response=281ms, CHR Overshoot 0%=421ms, CHR Overshoot 20%=525ms, dan Cohen-Coon=328ms.

Kata kunci: Kinematika, PID, robot keseimbangan bola

ABSTRACT

The ball balance robot is one of the dynamic balancing robots that must be controlled to stabilize it and remain balanced perpendicular to 0° above the ball on the ground or floor. The purpose of this study is to achieve a complete PID control system, namely by adding an integral constant (Ki) to the ball balance robot which is expected to be able to eliminate steady-state errors and to compare several methods of tuning the PID control system to achieve optimal results on the ball balancing robot. The implementation method includes construction design work, determining work flowcharts, determining kinematics and determining control systems. The control system used is the PID control system. Overall data processing is processed by the Arduino Mega2560 microcontroller which receives angle readings from the MPU6050 sensor as input and actuation on the stepper motor as output. Tests carried out include testing sensors, motors, motor rotation direction, robot response, and the implementation of the PID control system on the robot. Of the various tuning methods tested on the robot, the manual adjustment method CHR Overshoot 20% gives the robot a slightly longer fall time of 1193ms, compared to other methods before manual adjustment, namely Ziegler-Nichlos step response=281ms, CHR Overshoot 0%=421ms, CHR Overshoot 20%=525ms, and Cohen-Coon=328ms.

Keywords: Ball-balancing robot, kinematics, PID

KATA PENGANTAR

Bismillahirrohmanirrohiim,

Alhamdulillah, senantiasa kita panjatkan rasa syukur kepada Allah Yang Maha Esa atas semua limpahan rahmat dan karunia-Nya sehingga laporan proyek akhir ini yang berjudul “Robot Keseimbangan Pada Sebuah Bola Menggunakan Sistem Kendali PID” dapat diselesaikan tepat pada waktunya. Kami ucapkan terima kasih juga kepada semua pihak yang telah membantu kami selama pengerjaan proyek akhir dan dalam penyusunan laporan proyek akhir ini, yaitu:

1. Kedua orangtua dan keluarga yang selalu mendo’akan dan memberikan dukungan yang baik kepada penulis. Baik dalam pelaksanaan proyek akhir, perkuliahan, serta urusan-urusan lainnya.
2. Bapak Muhammad Iqbal Nugraha, M.Eng dan Bapak Indra Dwisaputra, M.T selaku dosen pembimbing yang bersedia membimbing, memberikan saran dan masukan, serta berbagi ilmunya selama pelaksanaan proyek akhir ini.
3. Dosen dan staf pengajar di Politeknik Manufaktur Negeri Bangka Belitung yang selama ini telah mendidik dan memberikan banyak pelajaran sehingga menambah wawasan serta pemahaman bagi penulis.
4. Teman sekelas serta pihak lainnya yang telah bersedia berbagi ilmunya.

Laporan ini dibuat sebagai salah satu syarat kelulusan Program Sarjana Terapan Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung. Harapan kami dengan adanya laporan ini dapat bermanfaat sebagai referensi pembelajaran bagi pembaca. Kami sadar bahwa masih banyak kekurangan yang terdapat di dalam penulisan laporan proyek akhir ini. Oleh karena itu, kami berharap kepada para pembaca untuk memberikan kritik dan saran yang positif kepada kami.

Sungailiat, 14 Februari 2022

Penulis

DAFTAR ISI

	Halaman
LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI	vi
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
DAFTAR LAMPIRAN	xi
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Perumusan dan Batasan Masalah	2
1.2.1. Perumusan Masalah	2
1.2.2. Batasan Masalah	2
1.3. Tujuan dan Manfaat Proyek Akhir	2
1.3.1. Tujuan Proyek Akhir	2
1.3.2. Manfaat Proyek Akhir	3
BAB II DASAR TEORI	4
2.1. Pengenalan Robot Keseimbangan Bola	4
2.2. Mekanisme Robot	5
2.3. Kinematika	7
2.4. Sistem Kendali	7
2.5. <i>Inertial Measurement Unit (IMU)</i>	9
2.5.1. Akselerometer	10
2.5.2. Giroskop	11
2.6. Kalman Filter	12
2.7. Mikrokontroler	13

2.7.1. Arduino Mega2560	13
BAB III METODE PELAKSANAAN	14
3.1. Desain Kontruksi	14
3.2. Blok Diagram <i>Hardware</i> dan Desain Elektrikal	15
3.2.1. Blok Diagram <i>Hardware</i>	15
3.2.2. Desain Elektrikal	16
3.3. Blok Diagram Sistem Kendali dan Desain <i>Flowchart</i>	16
3.3.1. Blok Diagram Sistem Kendali	16
3.3.2. Desain <i>Flowchart</i>	17
3.4. Konfigurasi Motor Stepper.....	18
3.5. Pergerakan Robot	18
3.5.1. Gerakan Maju dan Mundur	18
3.5.2. Gerakan Kanan dan Kiri.....	19
3.5.3. Gerakan Berputar	19
3.5.4. Kinematika	20
BAB IV PEMBAHASAN.....	21
4.1. Pengujian Motor Stepper	21
4.2. Pengujian Kinematika Robot.....	22
4.3. Pengujian Respon Robot	23
4.4. Penentuan Parameter PID.....	25
4.5. Pengujian PID Kontrol Keseimbangan	26
BAB V KESIMPULAN DAN SARAN.....	29
5.1. Kesimpulan.....	29
5.2. Saran	29
DAFTAR PUSTAKA	30

DAFTAR TABEL

Tabel	Halaman
2.1 Tanggapan Sistem Kontrol PID Terhadap Perubahan Parameter	9
4.1 Data Pengujian Kecepatan Motor (RPM)	22
4.2 Data Pengujian Kinematika Robot	23
4.3 Data Penarikan Garis Asimtot Grafik <i>Alpha</i> 0.7	25
4.4 Parameter PID Menggunakan Berbagai Metode	26

DAFTAR GAMBAR

Gambar	Halaman
2.1 Robot Keseimbangan Bola.....	4
2.2 Penggerak Bola	5
2.3 Definisi Sumbu dan Hubungan Antara Bola dan Roda	6
2.4 Bagan Sistem Kendali PID.....	8
2.5 Modul MPU6050	10
2.6 Orientasi Sumbu Penginderaan Akselerometer	10
2.7 Proses Integral Terhadap Suatu Sinyal	11
2.8 Orientasi Sumbu Penginderaan Giroskop	11
2.9 Giroskop Mekanik.....	12
2.10 Arduino Mega2560	13
3.1 Desain Kontruksi Keseluruhan Robot Keseimbangan Bola	14
3.2 Desain Robot Dari Sudut Pandang Tertentu	15
3.3 Blok Diagram <i>Hardware</i>	15
3.4 Desain Elektrikal	16
3.5 Blok Diagram Sistem Kendali Pada Robot.....	16
3.6 Desain <i>Flowchart</i> Program	17
3.7 Gerakan Maju dan Mundur	18
3.8 Gerakan ke Kanan dan ke Kiri	19
3.9 Gerakan Berputar	19
3.10 Definisi Sumbu dan Hubungan Antara Bola dan Roda Tampak Atas ...	20
4.1 Pengujian Kecepatan Masing-Masing Motor Stepper	21
4.2 Pengujian Respon Robot	24
4.3 Grafik Pengujian Respon Robot.....	24
4.4 Penarikan Garis Asimtot Pada Grafik <i>Alpha 0.7</i>	25
4.5 Grafik dari Berbagai Metode.....	26
4.6 Grafik <i>Manual Adjustment CHR Overshoot 20%</i>	27

DAFTAR LAMPIRAN

Lampiran 1: Daftar Riwayat Hidup

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Robot keseimbangan pada sebuah bola adalah salah satu robot penyeimbang dinamis, sama halnya seperti robot penyeimbang roda dua. Lal, et al. [1] menyatakan bahwa robot ini harus dikontrol secara aktif untuk menstabilkannya agar tetap seimbang tegak lurus diatas bola pada permukaan tanah atau lantai, dimana robot yang mereka buat mampu berosilasi seimbang pada kemiringan 5° pada sumbu Y dan 7° pada sumbu X. Salah satu keunggulan utama robot keseimbangan bola dibandingkan dengan robot penyeimbang roda dua yaitu robot mampu bergerak ke segala arah pada sumbu X dan sumbu Y, sedangkan robot penyeimbang roda dua hanya bisa berputar pada porosnya untuk mendapatkan arah tertentu [2].

Cukup banyak penelitian dan desain robot keseimbangan bola yang sudah diusulkan dalam literatur, diantaranya oleh Lal, et al. [1] seperti yang telah disebutkan diatas, kemudian oleh Kadam, et al. [3] yang membuat model matematika dan melakukan simulasi animasi robot keseimbangan bola yang menggunakan kombinasi sistem kendali LQR dan PID, namun belum mengimplementasikan ke dalam robot nyata, serta buatan Cai, et al. [4] yang menggunakan sistem kendali kaskade Fuzzy dan PD untuk kontrol keseimbangan, kontrol *yaw*, serta sistem kendali PI untuk kontrol posisi yang telah mencapai keseimbangan dinamis hingga $0,4^\circ$.

Untuk membuat robot tetap seimbang pada proyek akhir ini diperlukan sebuah sistem kontrol yang dapat mengendalikannya. Sistem kendali yang digunakan ialah sistem kendali PID. Pada penelitian ini, kontribusi yang diberikan ialah implementasi sistem kendali PID secara lengkap, yaitu dengan menambahkan konstanta integral (K_i) pada robot keseimbangan bola yang diharapkan mampu mengeliminasi *steady-state error* sesuai dengan teori PID [5] dengan robot keseimbangan roda dua yang dibuatnya yang telah mampu mengeliminasi *steady-*

state error pada simulasi menggunakan *software* Matlab. Kemudian kontribusi yang dilakukan selanjutnya ialah melakukan perbandingan dari beberapa metode *tuning* sistem kendali PID untuk mencapai hasil yang optimal pada robot keseimbangan bola pada proyek akhir ini.

1.2. Perumusan dan Batasan Masalah

1.2.1. Perumusan Masalah

Perumusan masalah yang timbul dari latar belakang diatas ialah:

- Bagaimana cara merancang sebuah robot keseimbangan pada sebuah bola?
- Bagaimana cara mengimplementasikan sistem kendali PID pada robot keseimbangan pada sebuah bola?

1.2.2. Batasan Masalah

Adapun batasan masalah dalam proyek akhir ini ialah:

- Sistem kendali yang digunakan ialah sistem kendali PID.
- Bola yang digunakan ialah bola berbahan PVC dengan diameter 13 cm.
- Sasis yang digunakan pada robot berbahan akrilik dengan ketebalan 3 mm.
- Motor yang digunakan ialah motor stepper bipolar.
- Mikrokontroler yang digunakan ialah Arduino Mega2560.
- Analisa kinematika yang digunakan ialah kinematika mundur.
- Proyek akhir ini hanya fokus pada sistem keseimbangan robot pada bola saja dan mengabaikan pergerakan mobilitas robot.

1.3. Tujuan dan Manfaat Proyek Akhir

1.3.1. Tujuan Proyek Akhir

Adapun tujuan dari pelaksanaan proyek akhir ini ialah:

- Merancang robot keseimbangan bola yang dapat mempertahankan posisi pada kondisi seimbang dan tegak lurus terhadap permukaan tanah atau lantai pada bidang datar.

- Mengimplementasikan kontroler PID berbasis Arduino pada robot keseimbangan pada sebuah bola.

1.3.2. Manfaat Proyek Akhir

Adapun manfaat dari pelaksanaan proyek akhir ini ialah:

- Diperolehnya rancangan sistem keseimbangan robot pada sebuah bola dari sisi kinematika dan kontrolnya.
- Diperolehnya rancangan sistem kontrol berbasis PID yang sesuai pada robot keseimbangan bola.

BAB II

DASAR TEORI

2.1. Pengenalan Robot Keseimbangan Bola

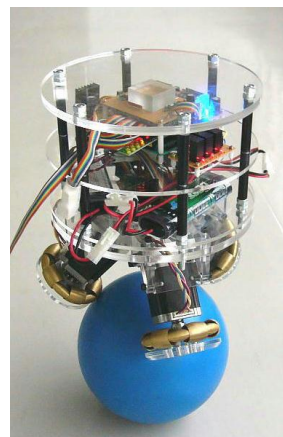
Robot keseimbangan pada sebuah bola atau lebih dikenal dengan istilah *Ball-Balancing Robot* (BBR) merupakan robot yang diharuskan mempertahankan dirinya agar tetap tegak lurus 0° diatas bola yang ada di bawahnya dengan permukaan tanah atau lantai [6]. Ketika sensor mendeteksi kemiringan antara sumbu X dan Y tidak sama dengan 0° , maka kontroler akan bekerja sedemikian rupa untuk menggerakkan motor stepper bergerak menuju sumbu X dan Y kembali pada titik 0° .

Robot keseimbangan bola sendiri sudah cukup banyak penelitian dan desain yang diusulkan dalam literatur, diantaranya sebagai berikut:

- BBR pertama ditunjukkan pada Gambar 2.1(a) yang dikembangkan pada tahun 2006 di Universitas Carnegie Mellon. Robot tersebut dibuat dengan tinggi seperti manusia yang memang diperuntukan untuk berinteraksi dengan manusia.
- BBR berikutnya dibuat oleh Universitas Tohoku Gakuin dengan nama robotnya ialah *BallIP* yang ditunjukkan pada Gambar 2.1(b). Pada robot ini roda digerakkan oleh tiga buah roda omni menggunakan motor stepper tanpa *gearbox*.



(a)

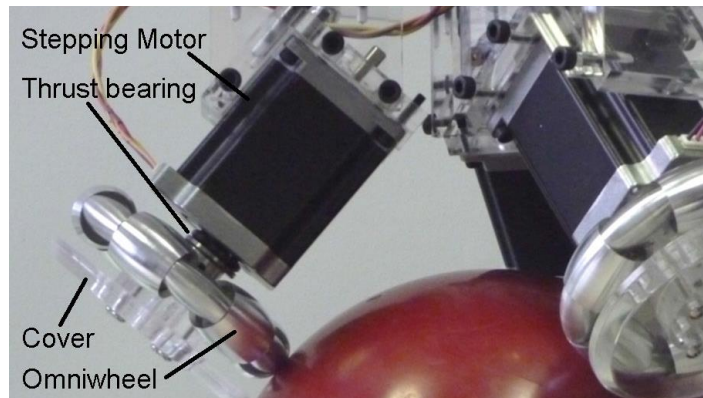


(b)

Gambar 2.1 Robot Keseimbangan Bola: (a) *BallBot* [7] dan (b) *BallIP* [8]

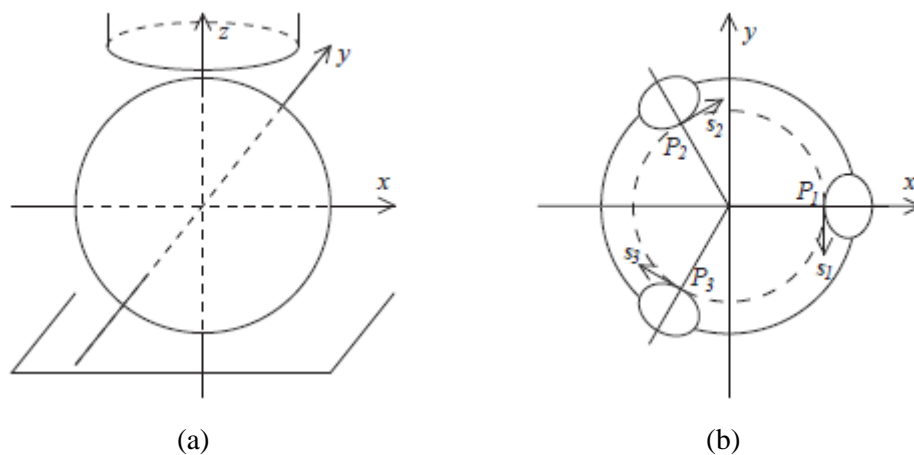
2.2. Mekanisme Robot

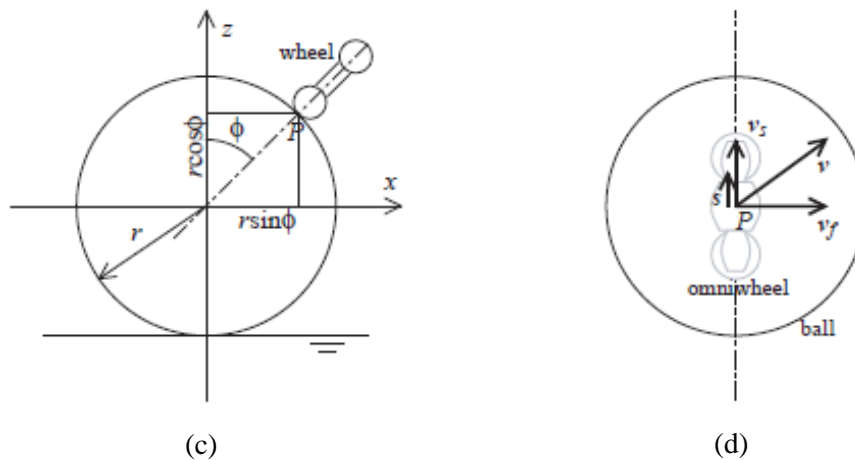
Pada bagian tubuh robot, terdiri dari sirkuit elektronik termasuk sensor, mikrokontroler, *driver* motor, serta tiga buah motor stepper yang menggunakan roda omni. Roda dihubungkan langsung dengan poros motor tanpa menggunakan *gearbox* seperti yang dilampirkan pada Gambar 2.2 berikut.



Gambar 2.2 Penggerak Bola [9]

Kelebihan penggunaan motor stepper menurut Masaaki, et al. [9]. ialah pengurangan biaya mekanisme, sirkuit penggerak, perangkat lunak kontrol karena karakteristik *loop* terbuka, serta torsi yang lebih besar bila dibandingkan motor servo DC. Mekanisme penggerak secara langsung ini memberikan osilasi yang halus dan juga rendah pada robot yang akan dikendalikan.





Gambar 2.3 Definisi Sumbu dan Hubungan Antara Bola dan Roda [9]: (a) Sumbu Koordinat Lokal, (b) Bola dan Roda Tampak Atas, (c) Bola dan Roda Tampak Samping, dan (d) Kecepatan Relatif Antara Bola dan Roda

Berdasarkan robot yang dibuat oleh Masaaki, et al. [9], seperti yang ditunjukkan pada Gambar 2.3(b), ketiga roda dipasang secara simetris dengan jarak masing-masing roda ialah 120° melalui poros motor untuk membuatnya tegak lurus terhadap bidang singgung bola, seperti yang ditunjukkan pada Gambar 2.3(c). Sudut vertikal ϕ adalah 50° . Roda omni yang digunakan pada robot memudahkan untuk menggerakkan robot pada permukaan yang berbentuk bola, dengan diameter roda yang digunakannya ialah 100 mm.

Jenis motor yang digunakan ialah motor stepper *hybrid* KH56QM2-913 (Nidec Servo Co.) dengan konfigurasi bipolar untuk mendapatkan torsi sekitar 2 Nm. Setiap motor dihubungkan dengan *driver* TA8435 untuk membuat konfigurasi *full step* $1,8^\circ$ /langkah menjadi *microstepping* $1/8$ yaitu $0,225^\circ$ /langkah untuk mendapatkan putaran yang halus. Baterai yang digunakan ialah baterai Ni-MH 7,2V untuk dapat menjalankan ketiga motor dan pengontrol [9].

Sensor yang digunakan dalam sistem kontrol adalah sensor MEMS dan mikrokontroler 16-bit H8/3052 (Renesas). *Output* dari 2 buah sensor *gyro* kecepatan sudut ADXRS401 (perangkat analog) dan akselerometer ADXL203 (perangkat analog) dikonversikan menjadi sinyal kemiringan/sudut dan dikombinasikan menjadi satu sinyal dalam satuan frekuensi [10]. Sinyal sensor *gyro* digunakan untuk mendapatkan respon yang lebih tinggi dan sinyal dari sensor akselerometer untuk mendapatkan stabilitas sudut kemiringan mutlak [9].

2.3. Kinematika

Perhitungan kinematika dibutuhkan untuk dapat menghitung kecepatan roda untuk pengemudian bola. Seperti yang ditunjukkan pada Gambar 2.3(a), didefinisikan kerangka koordinat lokal dengan asalnya di pusat bola. Sumbu robot dipasang pada badan robot, yaitu sumbu Z sepanjang garis vertikal melewati pusat bola dan pusat massa robot, dan sumbu X dan Y tegak lurus terhadap sumbu Z.

Seperti pada Gambar 2.3(b), kecepatan perintah roda virtual (v_x, v_y) yang diturunkan dari percepatan sebagai *input* kontrol diubah menjadi kecepatan ketiga roda nyata pada persamaan 1 berikut:

$$\begin{pmatrix} v_{s1} \\ v_{s2} \\ v_{s3} \end{pmatrix} = \begin{pmatrix} -v_y + \cos \phi + K_z \omega_z \\ \frac{\sqrt{3}}{2} v_x \cos \phi + \frac{1}{2} v_y \cos \phi + K_z \omega_z \\ -\frac{\sqrt{3}}{2} v_x \cos \phi + \frac{1}{2} v_y \cos \phi + K_z \omega_z \end{pmatrix} \dots\dots\dots (1)$$

Dimana :

v_{s1} = Kecepatan motor s_1

v_{s2} = Kecepatan motor s_2

v_{s3} = Kecepatan motor s_3

$K_z = -r \sin \phi$

K_z merupakan koefisien putaran disekitar sumbu vertikal dari kontrol pendulum terbalik [9]. Sudut vertikal ϕ menentukan rasio putaran roda terhadap bola dan memengaruhi penopang badan robot [11].

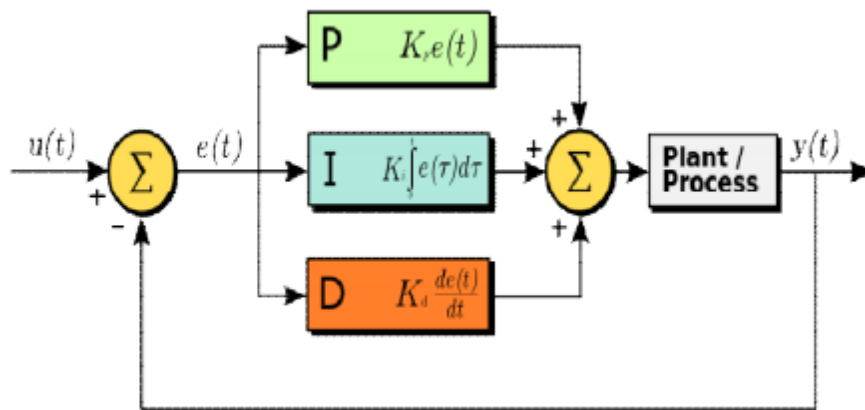
2.4. Sistem Kendali

Sistem kendali yang digunakan pada proyek akhir ini yang digunakan untuk mengendalikan kemiringan sudut dari *ball-balancing robot* ialah sistem kendali PID. PID merupakan singkatan dari *Proportional-Integral-Derivative*. Sistem kontrol PID akan terus melakukan proses pembacaan dan penyesuaian secara kontinyu untuk meminimalisir nilai *error* [12]. Nilai-nilai dari pembacaan PID akan dikirimkan ke mikrokontroler, kemudian diproses oleh mikrokontroler untuk menggerakkan aktuator agar dapat memperkecil nilai *error*.

Sistem kendali PID sendiri memiliki bermacam-macam struktur, yang biasa digunakan adalah PID ideal atau PID tipe A. Pada proyek akhir ini, sistem yang digunakan adalah sistem kendali PID tipe A bentuk independen. Persamaannya ialah sebagai berikut [13]:

$$c_o(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \dots \dots \dots (2)$$

Berikut bentuk blok diagram sistem kendali PID bentuk independen:



Gambar 2.4 Bagan Sistem Kendali PID [6]

Dimana:

- $C_o(t)$: Keluaran sistem kendali PID
- $e(t)$: *Error*
- $u(t)$: *Setpoint*
- $y(t)$: Keluaran sistem kendali (variabel proses)
- K_P : Konstanta proposional
- K_I : Konstanta integral
- K_D : Konstanta derivatif
- T_I : Waktu integral

Pada sistem kontrol PID terdapat 3 cara pengaturan yaitu kontrol P (Proporsional), D (Derivatif) dan I (Integral). Dalam penggunaannya bisa digunakan sendiri maupun gabungannya. Pada saat melakukan rancangan sistem kendali PID, hal penting yang harus diperhatikan ialah pengaturan parameter P, I, dan D agar keluaran dari sistem yang dirancang terhadap masukan sesuai dengan yang diinginkan.

Tabel 2.1 Tanggapan Sistem Kontrol PID Terhadap Perubahan Parameter

<i>Closed Loop Response</i>	<i>Rise time</i>	<i>Overshoot</i>	<i>Settling Time</i>	<i>Steady-state Error</i>
Propositional (Kp)	Menurun	Meningkat	Perubahan Kecil	Menurun
Integral (Ki)	Menurun	Meningkat	Meningkat	Hilang
Derivatif (Kd)	Perubahan Kecil	Menurun	Menurun	Perubahan Kecil

Untuk merancang sistem kontrol PID, kebanyakan dilakukan dengan metode coba-coba atau (*trial & error*). Hal ini dikarenakan masing-masing konstanta memberikan tanggapan yang berbeda-beda seperti yang ditunjukkan pada Tabel 2.1. Maka, untuk mendapatkan hasil kontrol yang baik diperlukan langkah coba-coba dengan mengubah konstanta P, I dan D hingga menemukan nilai yang diinginkan [14].

2.5. *Inertial Measurement Unit (IMU)*

Inertial Measurement Unit (IMU) merupakan sensor yang mampu mengumpulkan data kecepatan *angular* dan kecepatan linier, kemudian dikirim ke mikrokontroler untuk memperoleh informasi keberadaan dan pergerakan suatu benda. Sensor ini merupakan gabungan dari akselerometer (sensor percepatan) dan giroskop (sensor kecepatan *angular*) yang ditempatkan pada tiga sumbu yaitu sumbu X, sumbu Y, dan sumbu Z yang saling tegak lurus [15]. Sensor ini cocok diterapkan pada sistem navigasi dan sistem kontrol.

Sensor IMU yang digunakan pada proyek akhir ini yaitu sensor dengan tipe MPU6050 yang mempunyai 6 nilai data keluaran yaitu 3 nilai keluaran dari hasil pembacaan akselerometer dan 3 nilai keluaran dari hasil pembacaan giroskop. Selain mampu melakukan pembacaan akselerometer dan giroskop, kemampuan lainnya dari sensor ini yaitu mampu membaca temperatur di sekitar sensor.

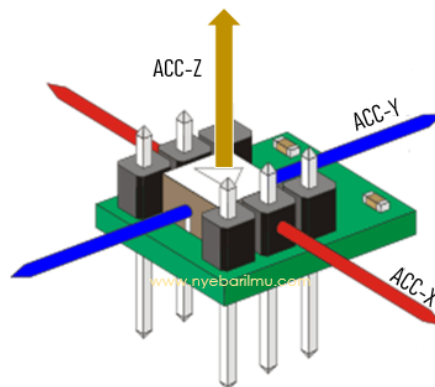


Gambar 2.5 Modul MPU6050 [16]

Sensor ini bekerja dengan mengonversi data analog menjadi data digital yang memiliki resolusi 16-bit pada masing-masing *channel* sehingga pembacaan data dari sensor dapat dikatakan akurat dan juga mampu membaca data dari *channel* X, Y, dan Z secara bersamaan dalam satu waktu. Komunikasi dari sensor MPU6050 ke mikrokontroler dihubungkan melalui standar komunikasi *Inter Integrated Circuit (I2C)* melalui pin *Serial Data (SDA)* dan *Serial Clock (SCL)*.

2.5.1. Akselerometer

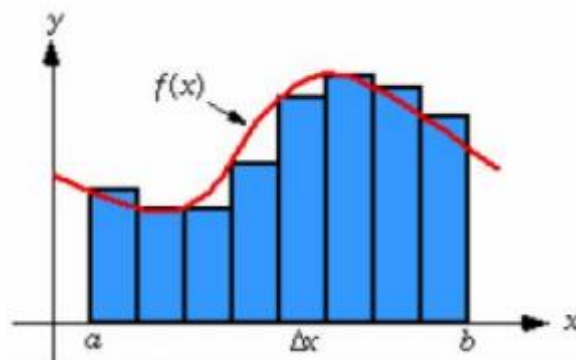
Akselerometer merupakan sensor yang berfungsi untuk mengukur percepatan suatu benda atau objek yang bergerak secara statis maupun dinamis.



Gambar 2.6 Orientasi Sumbu Penginderaan Akselerometer [16]

Pengukuran dinamis ialah pengukuran percepatan pada objek yang bergerak, sedangkan pengukuran statis ialah pengukuran percepatan objek terhadap gravitasi bumi.

Umumnya akselerometer digunakan untuk mengukur sudut kemiringan (*tilt*) dan prinsip kerja dari akselerometer menggunakan prinsip percepatan atau akselerasi. Perubahan arah pergerakan dari suatu benda akan menghasilkan percepatan. Untuk memperoleh data jarak dari sensor akselerometer, diperlukan proses akumulasi integral dari nilai keluaran sensor. Adapun ilustrasi proses integral suatu sinyal ditunjukkan pada Gambar 2.7.

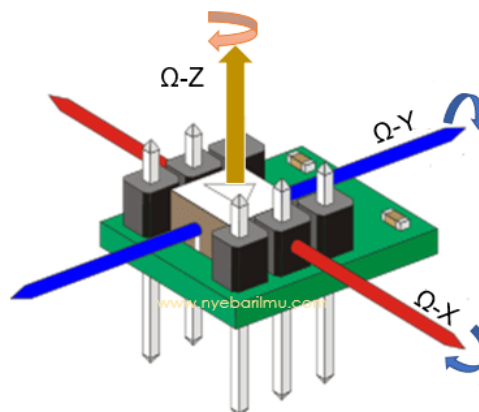


Gambar 2.7 Proses Integral Terhadap Suatu Sinyal [17]

Proses penghitungan ini dipengaruhi oleh waktu pengambilan sampel, sehingga jeda waktu pengambilan sampel (dt) harus selalu konstan dengan waktu pengambilan sampel dibuat sekecil mungkin.

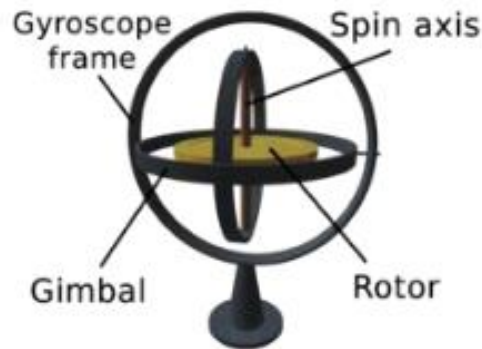
2.5.2. Giroskop

Giroskop merupakan sensor yang digunakan untuk melakukan pengukuran atau mempertahankan orientasi (putaran) berdasarkan pada prinsip momentum sudut secara mekanis.



Gambar 2.8 Orientasi Sumbu Penginderaan Giroskop [16]

Secara mekanis giroskop berbentuk seperti roda yang berputar atau seperti cakram dimana poros bebas berorientasi seperti ditunjukkan pada Gambar 2.9.



Gambar 2.9 Giroskop Mekanik [15]

Giroskop memberikan keluaran berupa kecepatan sudut dari arah 3 sumbu yaitu, sumbu X sebagai sudut ϕ (ϕ) yaitu di bagian kanan dan kiri, sumbu Y sebagai sudut θ (θ) yaitu di bagian atas dan bawah, dan sumbu Z sebagai sudut ψ (ψ) yaitu di bagian depan dan belakang. Sensor giroskop dapat mendeteksi gerakan dari objek yang diukur sesuai dengan gravitasi.

2.6. Kalman Filter

Pada subbab 2.5 yang membahas tentang sensor IMU, sensor tersebut mampu melakukan pembacaan dari pengukuran akselerasi dan rotasi. Sedangkan pada proyek akhir ini perlu penambahan pembacaan sudut kemiringan. Oleh karena itu dibutuhkan sebuah algoritma *filter* yang mampu memprediksi atau mengestimasi sudut kemiringan yang didapatkan dari hasil kombinasi pembacaan data akselerometer dan giroskop, dalam hal ini ialah Kalman Filter.

Kalman Filter merupakan algoritma *filter* yang mengambil data hasil pengukuran dari sensor yang masih terdapat sinyal gangguan. Dalam proses pengolahan data, Kalman Filter memiliki dua tahapan, yaitu memprediksi dan membarui. Tahapan pertama yaitu memprediksi data, Kalman Filter memberi perkiraan. Kemudian langkah kedua yaitu membarui, pada tahap ini Kalman Filter mengambil data yang telah diperbarui kemudian dibandingkan dengan hasil data perkiraan dengan *kalman gain* [18].

2.7. Mikrokontroler

2.7.1. Arduino Mega2560

Arduino Mega2560 adalah mikrokontroler berbasis ATmega2560 yang memiliki 54 pin digital *input* dan *output*, diantaranya 15 pin yang digunakan sebagai *output* PWM, 16 pin sebagai *input* analog, dan 4 pin sebagai UART (*port serial hardware*), kristal osilator berukuran 16 MHz, *port* USB untuk mengunggah kode program, *jack power* untuk memasukan daya, *header ICSP*, dan tombol *reset* [19].



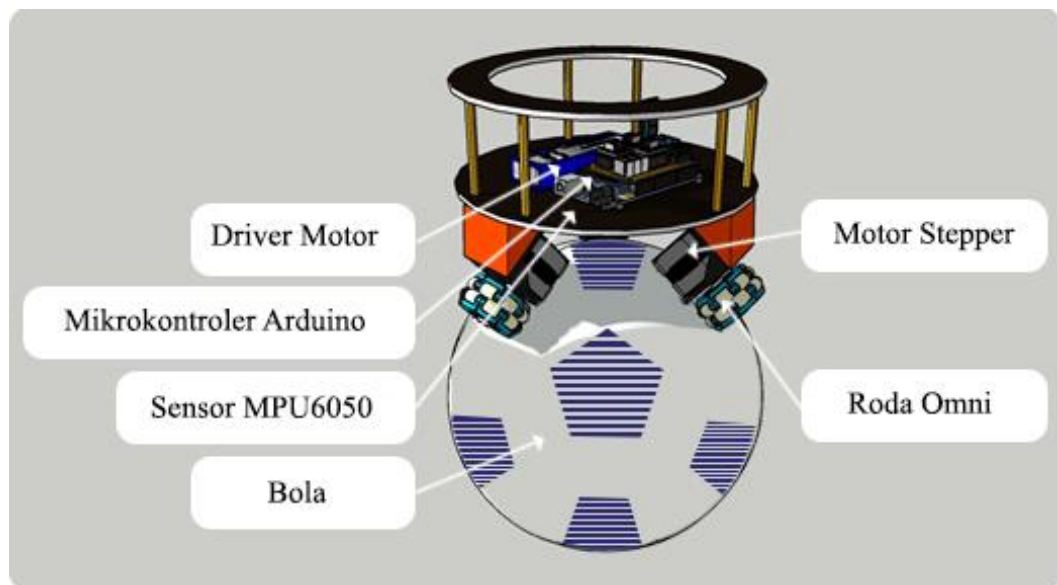
Gambar 2.10 Arduino Mega2560 [19]

Mikrokontroler Arduino Mega2560 berfungsi sebagai pusat kendali pada robot keseimbangan bola beroda tiga yang akan digunakan dalam proyek akhir ini. Alasan penggunaannya ialah dikarenakan penggunaan yang lebih mudah.

BAB III METODE PELAKSANAAN

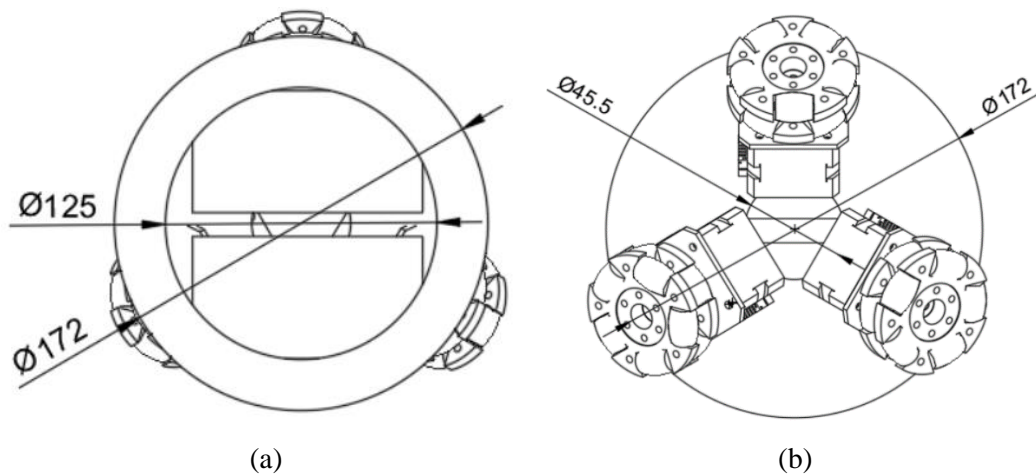
3.1. Desain Kontruksi

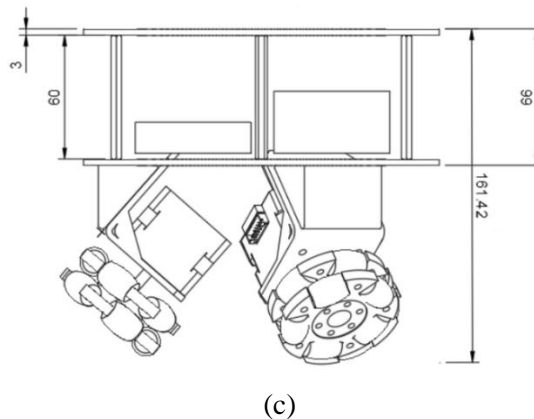
Desain kontruksi mencakup hasil rancangan kontruksi dari proyek akhir yang dibuat yang kemudian dilakukan perakitan dari keseluruhan perangkat sehingga menjadi robot keseimbangan pada sebuah bola.



Gambar 3.1 Desain Kontruksi Keseluruhan Robot Keseimbangan Pada Sebuah Bola

Kemudian untuk ukuran robot bila dilihat dari sudut pandang tertentu, dilampirkan pada Gambar 3.2 berikut.





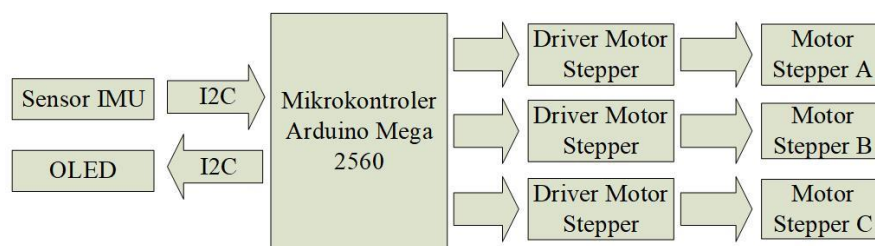
Gambar 3.2 Desain Robot Dari Sudut Pandang Tertentu: (a) Tampak Atas, (b) Tampak Bawah, dan (c) Tampak Samping

Desain dari robot keseimbangan pada sebuah bola ini menggunakan tiga buah motor stepper dengan masing-masing interval antar motor ialah sebesar 120°. Alasan penggunaan daripada tiga motor ialah agar robot dapat bergerak dengan dua derajat kebebasan untuk memperoleh keseimbangan, yaitu pada sumbu X dan sumbu Y, dengan kombinasi dari arah putar dan kecepatan masing-masing motor.

3.2. Blok Diagram *Hardware* dan Desain Elektrikal

3.2.1. Blok Diagram *Hardware*

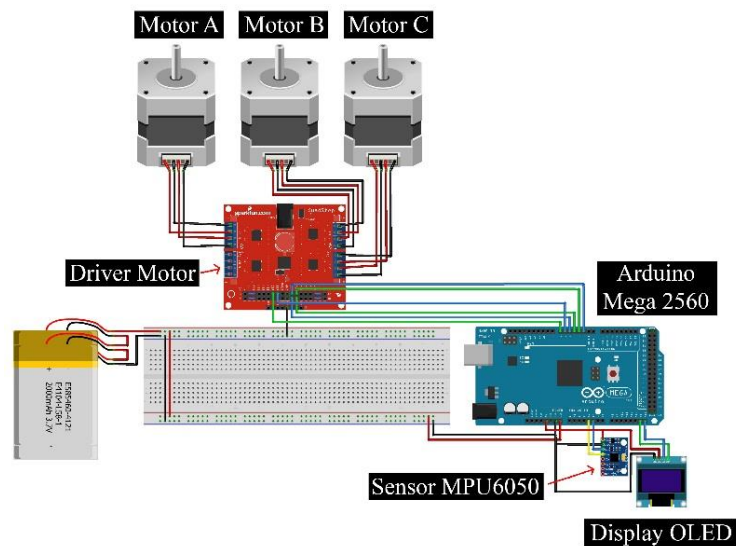
Blok diagram *hardware* dibuat untuk memetakan proses kerja dari proyek akhir yang dibuat yang menjabarkan hubungan antar perangkat *input* dan *output*.



Gambar 3.3 Blok Diagram *Hardware*

Sensor IMU berperan sebagai *input*, yaitu melakukan pembacaan sudut kemiringan, akselerasi, serta rotasi yang dihubungkan ke mikrokontroler menggunakan komunikasi I2C. OLED berperan sebagai *output* untuk menampilkan informasi sudut kemiringan yang dibaca oleh sensor IMU dan konstanta PID. Motor stepper berperan sebagai *output* yaitu aktuator untuk menyeimbangkan robot yang dihubungkan dengan mikrokontroler melalui *driver* motor stepper.

3.2.2. Desain Elektrikal



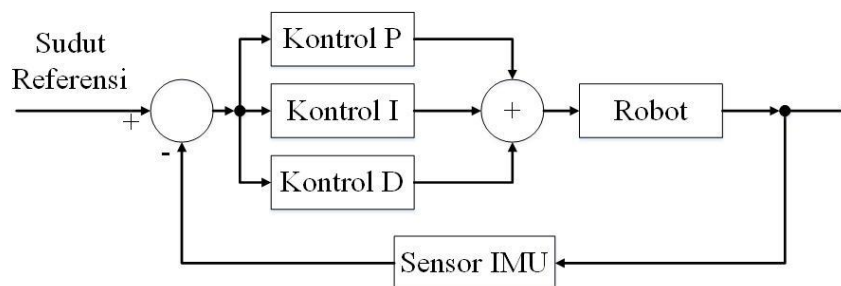
Gambar 3.4 Desain Elektrikal

Desain elektrikal menjabarkan keseluruhan koneksi antar perangkat agar dapat menjadi kendali atas robot keseimbangan pada sebuah bola. Dimulai dari koneksi sumber ke rangkaian sirkuit, ketiga motor stepper dengan *driver* motor, kemudian *driver* motor dihubungkan dengan mikrokontroler, serta sensor dan OLED yang terhubung dengan mikrokontroler.

3.3. Blok Diagram Sistem Kendali dan Desain *Flowchart*

3.3.1. Blok Diagram Sistem Kendali

Robot keseimbangan bola ini direalisasikan menggunakan sistem kendali PID. PID merupakan singkatan dari *Proportional–Integral–Derivative*. Sistem kontrol PID akan terus melakukan proses pembacaan dan penyesuaian secara kontinyu untuk meminimalisir nilai *error* [12]. Pada robot ini sistem kendali PID yang digunakan adalah PID ideal bentuk independen.

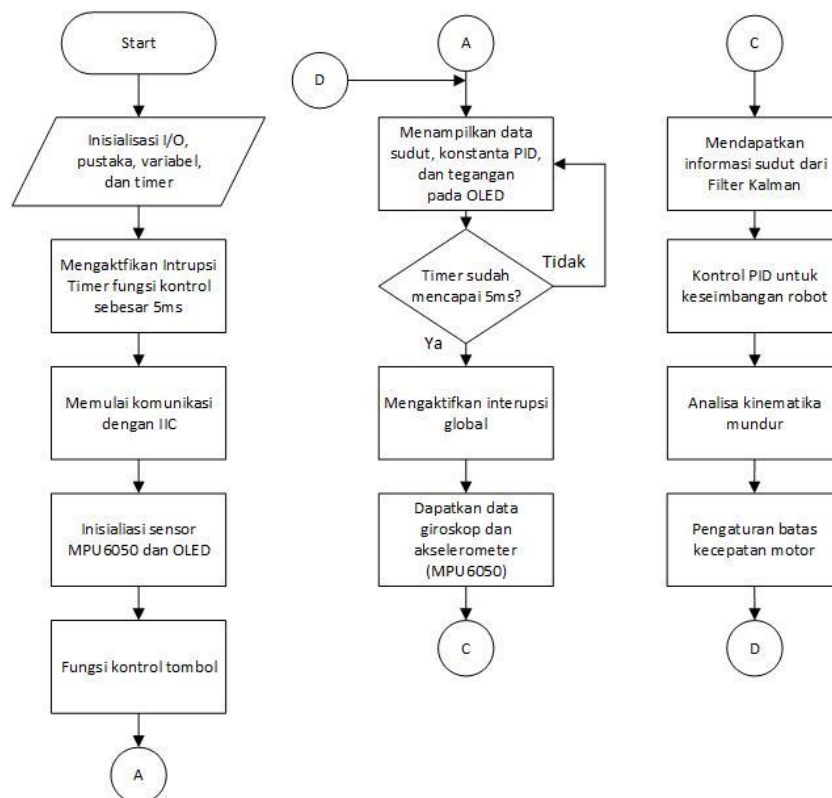


Gambar 3.5 Blok Diagram Sistem Kendali Pada Robot

Berdasarkan blok diagram kontrol robot pada Gambar 3.4, sistem dimulai dengan menginisialisasikan sudut referensi pada sensor MPU6050 sebesar 0° sebagai kondisi seimbang baik pada sumbu X maupun sumbu Y. Hasil pembacaan sudut dari sensor MPU6050 diproses oleh sistem kendali PID, dan hasil perhitungan dari PID diproses oleh mikrokontroler untuk menggerakkan motor stepper.

Dalam menentukan nilai parameter PID untuk sebuah sistem diperlukan pemilihan metode yang tepat. Dalam meriset metode yang tepat untuk sistem robot keseimbangan bola ini, dilakukan beberapa percobaan nilai parameter PID dari 3 metode yang berbeda, yaitu metode Ziegler-Nichols, CHR, Cohen-Coon.

3.3.2. Desain *Flowchart*



Gambar 3.6 Desain *Flowchart* Program

Gambar *flowchart* program diatas menjelaskan proses kerja dari robot keseimbangan bola. Sensor yang digunakan pada robot ini ialah sensor MPU6050, sensor ini mampu membaca kemiringan sudut berdasarkan data dari sensor akselerometer dan giroskop. Informasi sudut yang diperoleh dari sensor MPU6050 dikontrol menggunakan sistem kontrol PID untuk mengontrol keseimbangan robot.

3.4. Konfigurasi Motor Stepper

Motor yang digunakan pada proyek akhir ini ialah motor stepper yang memiliki resolusi $1.8^\circ/\text{langkah}$ atau bisa diartikan setiap langkah memberi putaran sebanyak 1.8° , sehingga apabila diasumsikan 1 putaran ialah 360° , maka pulsa yang harus diberikan untuk satu putaran penuh ialah sebanyak 200 langkah.

$$\frac{360^\circ/\text{putaran}}{1.8^\circ/\text{langkah}} = 200 \frac{\text{langkah}}{\text{putaran}}$$

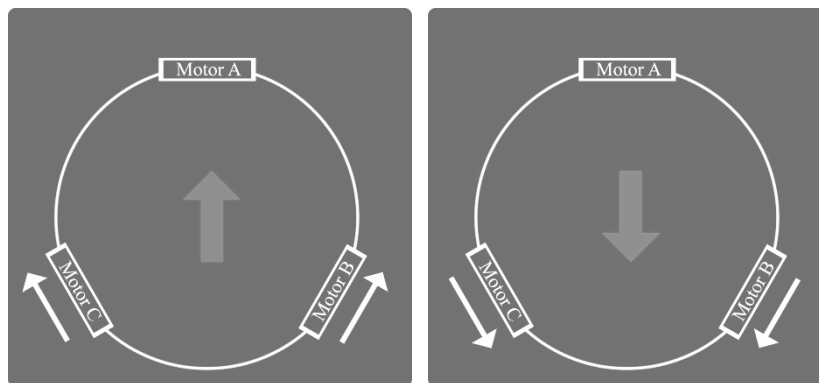
Pada proyek akhir ini, konfigurasi yang digunakan ialah *microstepping* 1/16 melalui *driver* motor LV8371V, sehingga untuk mendapatkan satu putaran penuh membutuhkan pulsa sebanyak 3200 langkah yang mampu menghasilkan resolusi hingga $0.1125^\circ/\text{langkah}$ yang memberikan efek putaran motor menjadi lebih halus.

$$\frac{360^\circ/\text{putaran}}{1.8^\circ/\text{langkah} \times \frac{1}{16}} = 3200 \frac{\text{langkah}}{\text{putaran}} = \frac{0.1125^\circ}{\text{langkah}}$$

3.5. Pergerakan Robot

3.5.1. Gerakan Maju dan Mundur

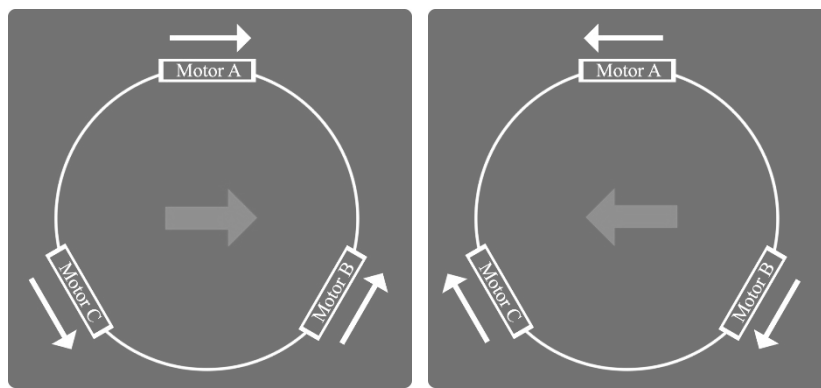
Ketika robot miring ke arah belakang, robot harus bisa membuat gerakan maju agar dapat mencapai titik keseimbangannya. Maka untuk dapat membuat robot bergerak maju, Motor B dan Motor C digerakkan ke arah depan dengan kecepatan yang sama, sedangkan Motor A diam. Dan apabila robot miring ke arah depan, robot harus bisa membuat gerakan mundur agar dapat mencapai titik keseimbangannya, maka untuk dapat melakukan gerakan mundur ialah kebalikan dari gerakan maju.



Gambar 3.7 Gerakan Maju dan Mundur

3.5.2. Gerakan Kanan dan Kiri

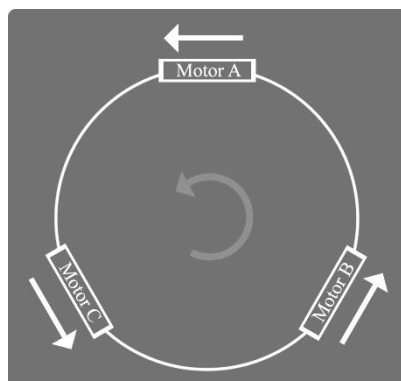
Ketika robot miring ke arah kiri, robot harus bisa membuat gerakan ke kanan agar dapat mencapai titik keseimbangannya. Maka untuk membuat gerakan ke kanan pada robot maka Motor B bergerak ke arah depan, Motor C bergerak ke arah belakang, sedangkan Motor A bergerak ke arah kanan dengan kecepatan yang berbeda pada setiap roda. Dan apabila robot miring ke arah kanan, robot harus bisa membuat gerakan ke kiri agar dapat mencapai titik keseimbangannya, maka untuk dapat melakukan gerakan ke kiri ialah kebalikan dari gerakan ke kanan.



Gambar 3.8 Gerakan ke Kanan dan ke Kiri

3.5.3. Gerakan Berputar

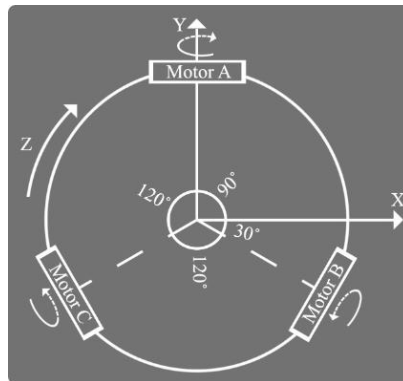
Untuk membuat gerakan berputar pada robot, maka Motor B bergerak ke arah depan, Motor C bergerak ke arah belakang, sedangkan Motor A bergerak ke arah kiri dengan kecepatan yang sama pada setiap roda.



Gambar 3.9 Gerakan Berputar

3.5.4. Kinematika

Arah kemiringan pada robot berada pada ruang sumbu X maupun sumbu Y dengan derajat kebebasan sebesar 360^0 . Hal ini membutuhkan sebuah perhitungan matematis yang dapat menghitung dan mengkonversi dari besarnya kemiringan robot menjadi respon kecepatan dan arah putaran motor yang sesuai. Kombinasi arah putaran dan kecepatan dari ketiga motor pada robot didapat berdasarkan perhitungan analisa kinematika mundur (*Inverse Kinematic*) [4] yang dilampirkan pada persamaan 3 dengan definisi sumbu dan hubungan antara bola dan roda dari tampak atas yang ditunjukkan pada gambar 3.10.



Gambar 3.10 Definisi Sumbu dan Hubungan Antara Bola dan Roda Tampak Atas

$$\begin{pmatrix} V_A \\ V_B \\ V_C \end{pmatrix} = \begin{pmatrix} V_X + V_Z \\ -\sin 30^\circ V_X - \cos 30^\circ V_Y + V_Z \\ -\cos 60^\circ V_X + \sin 60^\circ V_Y + V_Z \end{pmatrix} \dots\dots\dots (3)$$

Keterangan:

V_A : Kecepatan motor A

V_B : Kecepatan motor B

V_C : Kecepatan motor C

V_X : Kecepatan motor terhadap sumbu X

V_Y : Kecepatan motor terhadap sumbu Y

V_Z : Kecepatan motor terhadap sumbu Z

BAB IV PEMBAHASAN

Pembahasan pada bab 4 berikut mencakup uraian, analisis, hasil percobaan, pengukuran, serta observasi dari robot yang dibuat yang berkaitan pada topik yang terdapat pada bab 3.

4.1. Pengujian Motor Stepper

Motor stepper pada proyek akhir ini merupakan penggerak robot secara keseluruhan untuk dapat memosisikan robot dalam keadaan tegak dan seimbang. Oleh karena itu, perlu dilakukan pengujian pada masing-masing motor untuk memastikan keluaran kecepatan motor mampu berputar sesuai dengan kecepatan yang diberikan yang akan diimplementasikan pada robot. Pengujian motor stepper diuji menggunakan alat ukur *Tachometer*. Salah satu data pengujian kecepatan motor stepper ditunjukkan pada Gambar 4.1 berikut.



Gambar 4.1 Pengujian Kecepatan Masing-Masing Motor Stepper dengan Target Kecepatan 70 RPM

Dari pengujian kecepatan yang dilakukan, didapatkan data pengujian kecepatan masing-masing motor yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Data Pengujian Kecepatan Motor (RPM)

No.	Target	Pengukuran			<i>Error (%)</i>		
		Motor A	Motor B	Motor C	Motor A	Motor B	Motor C
1	20	19,5	19,5	19,5	2,5641	2,5641	2,5641
2	70	67,1	67,1	67,1	4,3219	4,3219	4,3219
3	100	94,5	94,5	94,5	5,8201	5,8201	5,8201
4	125	115,7	115,7	115,7	8,038	8,038	8,038
5	150	138,2	138,2	138,2	8,5384	8,5384	8,5384
Rata-Rata <i>Error (%)</i>					5,8565	5,8565	5,8565

Berdasarkan Tabel 4.1 diatas, dengan percobaan target kecepatan diantaranya 20 RPM, 70 RPM, 100 RPM, 125 RPM, dan 150 RPM, didapatkan persentase *error* ketiga motor tersebut berurutan-turut sebesar 2,5641%, 4,3219%, 5,8201%, 8,038%, dan 8,5384% sehingga didapatkan nilai persentase rata-rata *error* sebesar 5,865%. Maka, dengan data pengujian motor tersebut, didapatkan kesimpulan bahwa semakin besar target kecepatan yang diinginkan, berbanding lurus dengan persentase *error* yang didapatkan. Kesalahan tersebut bisa saja disebabkan dari kondisi pabrikan dari motor tersebut, yakni toleransi *error*. Meski demikian, ketiga motor stepper tersebut memiliki persentase *error* yang sama. Maka ketiga motor tersebut dapat diatur dengan satu sistem kontrol, dengan asumsi bahwa hanya ketika ketiga motor tersebut memiliki *output* kecepatan yang sama bila diberikan kecepatan tertentu seperti yang terdapat pada Tabel 4.1 sebelumnya. Oleh karena itu sistem kendali keseimbangan harus diimplementasikan pada sistem agar dapat menyesuaikan antara kecepatan yang diperintahkan oleh mikrokontroler dengan kecepatan yang keluar agar dapat mengatur keseimbangan pada robot.

4.2. Pengujian Kinematika Robot

Pengujian kinematika robot dilakukan untuk membandingkan antara perhitungan dan pengujian sebenarnya dari arah putaran motor sehingga robot mampu menggerakkan motor sesuai dengan arah kemiringan yang terbaca. Adapun perhitungan kinematika robot didapatkan dari analisa kinematika mundur yang

dibahas pada subbab 3.5.4 pada persamaan 3. Pada perhitungan dan pengukuran, parameter yang diukur ialah arah putaran motor dengan asumsi masing-masing sudut diberikan nilai kombinasi kemiringan 0° , 10° , dan -10° pada sumbu X dan Y. Pengujian arah putaran motor ditunjukkan pada Tabel 4.2 berikut.

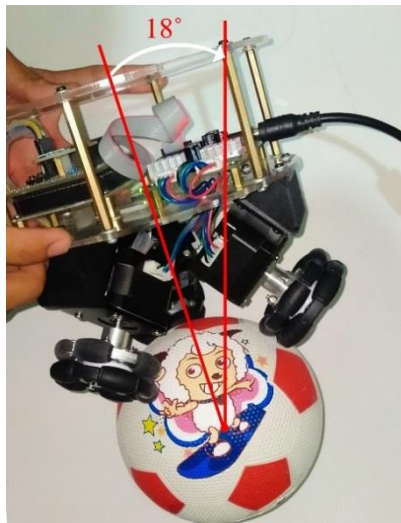
Tabel 4.2 Data Pengujian Kinematika Robot

No.	Sudut		Perhitungan			Pengukuran			Keterangan
	X	Y	Motor A	Motor B	Motor C	Motor A	Motor B	Motor C	
1	0	Y	0	CCW	CW	0	CCW	CW	Sesuai
2	X	0	CW	CCW	CCW	CW	CCW	CCW	Sesuai
3	X	Y	CW	CCW	CW	CW	CCW	CW	Sesuai
4	0	-Y	0	CW	CCW	0	CW	CCW	Sesuai
5	X	-Y	CW	CW	CCW	CW	CW	CCW	Sesuai
6	-X	0	CCW	CW	CW	CCW	CW	CW	Sesuai
7	-X	Y	CCW	CCW	CW	CCW	CCW	CW	Sesuai
8	-X	-Y	CCW	CW	CCW	CCW	CW	CCW	Sesuai

Berdasarkan hasil pengujian arah putaran motor pada Tabel 4.2, didapatkan perbandingan antara perhitungan dan pengujian masing-masing motor mampu berputar dengan arah yang sesuai dengan kemiringan robot. Dengan demikian, didapatkan kesimpulan bahwa perhitungan kinematika mundur sangat berpengaruh terhadap robot sehingga robot mampu menyesuaikan arah putaran motor pada ruang sumbu X maupun sumbu Y dengan derajat kebebasan sebesar 360° .

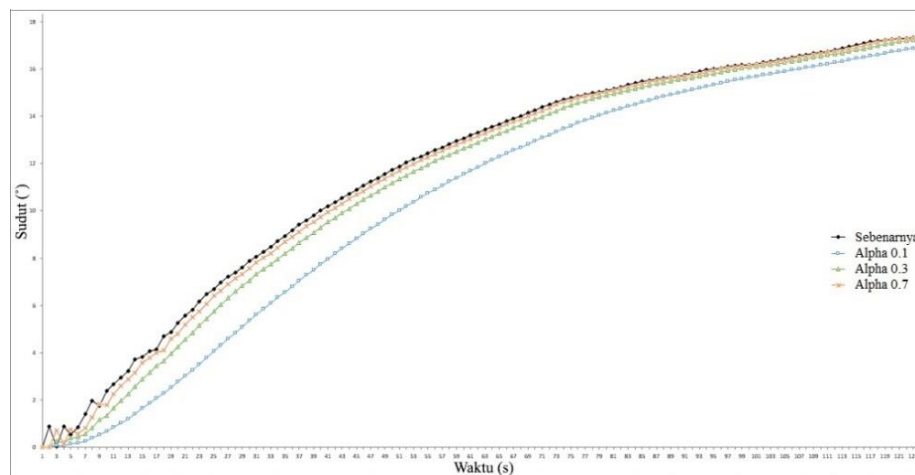
4.3. Pengujian Respon Robot

Tahapan pengujian ini dilakukan untuk mengetahui respon robot pada kemiringan tertentu, kemudian data yang telah didapatkan digunakan untuk mendapatkan *starting point* parameter PID. Pengujian dilakukan pada sistem *loop* terbuka terhadap sistem [20], dengan kecepatan motor yang diberikan ialah 93,75 RPM dari kemiringan sudut posisi robot ialah 18° menuju 0° terhadap sumbu Y seperti yang ditunjukkan pada Gambar 4.2.



Gambar 4.2 Pengujian Respon Robot

Berdasarkan pengujian tersebut, didapatkan data kemiringan sudut yang kemudian divisualisasikan kedalam bentuk grafik respon robot terhadap kemiringan seperti yang ditunjukkan pada Gambar 4.3.



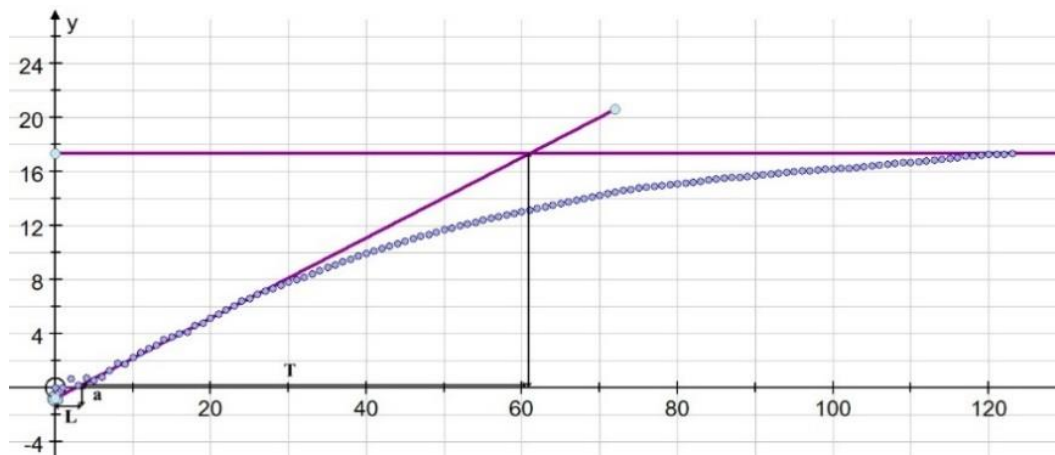
Gambar 4.3 Grafik Pengujian Respon Robot

Berdasarkan hasil pengujian respon robot pada grafik Gambar 4.1, didapatkan data respon robot sebenarnya dari kemiringan 18° menuju 0° yang masih terdapat sinyal *noise*. Oleh karena itu, dibutuhkan penghalusan data dari sudut yang terukur untuk dapat memudahkan dalam pengolahan data yang akan digunakan untuk penentuan parameter PID. Penghalusan grafik yang digunakan ialah metode penghalusan eksponensial dengan diberikan masing-masing nilai *alpha* 0.1, 0.3, dan 0.7 yang ditampilkan pada Gambar 4.3.

Berdasarkan grafik pada Gambar 4.3, dapat disimpulkan bahwa metode penghalusan eksponensial mampu menghilangkan sinyal *noise* pada grafik respon robot sehingga data yang telah dihaluskan lebih mudah digunakan dalam penentuan parameter PID pada robot.

4.4. Penentuan Parameter PID

Setelah mendapatkan data grafik yang telah dihaluskan menggunakan penghalusan eksponensial pada subbab 4.3, tahapan selanjutnya ialah penarikan garis asimtot pada salah satu data grafik. Pada pengujian ini, dipilih data nilai grafik *alpha* 0.7 seperti yang ditunjukkan pada Gambar 4.4.



Gambar 4.4 Penarikan Garis Asimtot Pada Grafik *Alpha* 0.7

Berdasarkan Gambar 4.4, didapatkan data hasil penarikan garis asimtot yang ditunjukkan pada Tabel 4.3.

Tabel 4.3 Data Penarikan Garis Asimtot Grafik *Alpha* 0.7

Konstanta	$a(^{\circ})$	L (ms)	L (s)	L+T (s)	T (s)	τ (s)
<i>Alpha</i> 0.7	0,8	2,69159	0,002692	0,060974	0,058283	0,04414288

Data yang telah didapatkan yang ditunjukkan pada Tabel 4.3 digunakan untuk perhitungan konstanta PID dari berbagai metode, diantaranya Ziegler-Nichols *step-response*, *CHR Overshoot 0%*, *CHR Overshoot 20%*, dan Cohen-Coon. Sehingga didapatkan hasil perhitungan parameter PID yang dilampirkan pada Tabel 4.4.

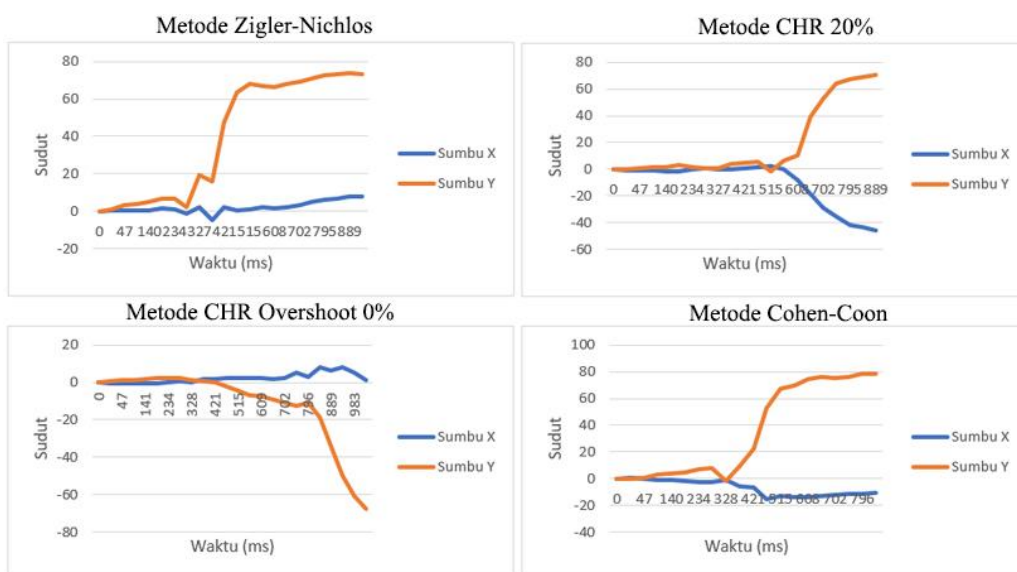
Tabel 4.4 Parameter PID Menggunakan Berbagai Metode

Metode	Kp	Ti	Td	Ki	Kd
ZN-step response	1,5	0,00538	0,00135	278,646	0,00202
CHR Overshoot 0%	1,1875	0,00646	0,00113	183,829	0,00134
CHR Overshoot 20%	1,5	0,00538	0,00113	278,646	0,00169
Cohen-Coon	1,7015	0,00661	0,00099	257,61	0,00168

Dari beberapa data yang telah didapatkan pada Tabel 4.4, data yang akan digunakan untuk pengujian kontrol keseimbangan diantaranya ialah nilai Kp (konstanta proporsional), Ki (konstanta integral), dan Kd (konstanta derivatif).

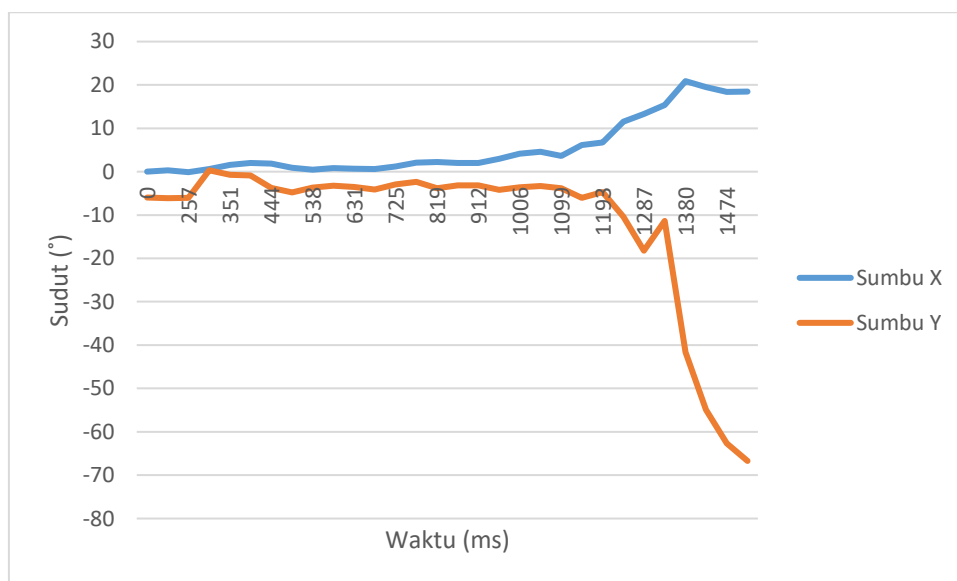
4.5. Pengujian PID Kontrol Keseimbangan

Setelah mendapatkan parameter PID yang ditunjukkan pada Tabel 4.4, tahap berikutnya ialah mengimplementasikan konstanta PID ke dalam program robot untuk dapat menguji hasil keseimbangan yang mampu diberikan oleh masing-masing parameter. Pengujian dimulai dari sudut kemiringan sumbu X dan Y masing-masing sebesar 0°, yaitu robot berada pada posisi tegak. Berikut ialah hasil pengujian konstanta PID pada robot dengan menggunakan konstanta yang telah didapat dari berbagai metode:



Gambar 4.5 Grafik Pengujian Keseimbangan dari Berbagai Metode

Dari beberapa hasil percobaan pada Gambar 4.5 diatas, terlihat bahwa konstanta PID yang telah didapat dari beberapa metode belum dapat menyeimbangkan robot sebagaimana yang diinginkan. Waktu jatuh robot dari masing-masing metode yaitu Ziegler-Nichlos-*step response*=281ms, CHR *Overshoot 0%*=421ms, CHR *Overshoot 20%*=525ms, dan Cohen-Coon=328ms. Sehingga perlu dilakukan *manual adjustment* dari salah satu metode yang mendapatkan hasil keseimbangan yang lebih baik. Metode yang dipilih untuk dilakukan *manual adjustment* ialah metode CHR *Overshoot 20%*, karena dari hasil pengujian yang ditunjukkan pada Gambar 4.5, metode tersebut mendapatkan hasil keseimbangan selama 525ms. Selanjutnya ialah melakukan penyesuaian parameter PID dengan melihat respon robot secara langsung dari konstanta yang diberikan. Pada pengujian ini, konstanta PID yang didapatkan dari hasil *manual adjustment* ialah $K_p=660$, $K_i=0.3$, dan $K_d=0.00003$ yang kemudian diuji dari sudut kemiringan sumbu $X=0^\circ$ dan $Y=6^\circ$.



Gambar 4.6 Grafik *Manual Adjustment* CHR *Overshoot 20%*

Berdasarkan grafik pada Gambar 4.6, didapatkan bahwa waktu jatuh robot sedikit lebih lama yaitu 1193ms dibandingkan dengan keempat data sebelumnya yaitu kurang dari 530ms. Maka dapat disimpulkan bahwa *manual adjustment* lebih efektif untuk mengatur konstanta PID pada robot keseimbangan bola ini, karena pengujian dilakukan dengan melihat respon robot secara langsung. Dimulai dari

membuat sistem mendapatkan kondisi *Overshoot* yang tidak jauh dari *setpoint*, menurunkan *Overshoot* secara perlahan, menghilangkan *steady-state error*, serta memperkecil *settling time* dan *rise time*. Meski sudah berusaha melakukan hal tersebut, nilai tuning PID yang didapatkan belum dapat menyeimbangkan robot secara sempurna. Hal ini dikarenakan faktor *tuning* konstanta PID yang belum sesuai dengan sistem. Oleh karena itu, diperlukan *tuning* lebih lanjut agar mendapatkan hasil sistem robot yang benar-benar seimbang.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, didapatkan hasil waktu jatuh robot diantaranya Ziegler Nichols-*step response*=281ms, CHR *Overshoot* 0%=421ms, CHR *Overshoot* 20%=525ms, dan Cohen-Coon=328ms. Dilakukan *manual adjustment* pada metode CHR *Overshoot* 20% untuk memperbaiki respon robot menjadi 1193ms. Secara keseluruhan dapat disimpulkan bahwa robot belum bisa dikatakan seimbang sempurna. Diantara faktor penyebabnya ialah *tuning* PID yang belum tepat, tidak ada pemodelan sistem secara keseluruhan sehingga pengujian keseimbangan harus dilakukan secara langsung tanpa membandingkan dengan simulasi, tidak ada *encoder* sebagai *feedback* untuk mengukur kecepatan nyata dari motor yang diinginkan dengan kecepatan motor yang dihasilkan, serta rancangan mekanis robot yang perlu direncanakan secara matang.

5.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan, didapatkan beberapa saran sebagai berikut:

- Pentingnya perencanaan matang terkait desain konstruksi robot yang akan dibuat. Karena robot tidak hanya tentang desain elektrik, melainkan juga rancangan mekanis robot yang sesuai.
- Pentingnya pengukuran dan pengecekan kelayakan sebuah perangkat sebelum diimplementasikan agar sesuai dengan hasil yang diinginkan.
- Perhitungan dan implementasi kinematika pada robot sangat dibutuhkan untuk dapat membuat sistem bergerak dengan arah yang sesuai.
- Pentingnya pemodelan sistem secara keseluruhan agar dapat membandingkan antara simulasi dengan pengujian sebenarnya dan mengurangi resiko kerusakan pada *hardware*.
- Pentingnya penentuan sebuah metode *tuning* sistem kendali agar mendapatkan kinerja yang lebih baik sesuai dengan sistem yang dikendalikan.

DAFTAR PUSTAKA

- [1] I. Lal, M. Nicoara, A. Codrean dan L. Busoniu, “*Hardware and control design of a ball balancing robot,*” *IEEE*, 2019.
- [2] C. Verdier, “*Geometric control of an under-actuated balancing robot,*” *Master of Science Thesis in Systems and Control at Delft University of Technology*, p. 1, 2015.
- [3] S. S. Kadam, G. N. Thokal dan S. Kadam, “*Mathematical Modeling and Simulation with Animation of Ball Balancing Robot,*” *IOSR Journal of Engineering (IOSRJEN)*, p. 17, 2018.
- [4] C. Cai, J. Lu dan Z. Li, “*Kinematic Analysis and Control Algorithm for the Ballbot,*” *IEEE Access*, vol. 7, p. 38317, 2019.
- [5] D. Jayakody dan K. Sucharitharathna, “*Control Unit for a Two-Wheel Self-Balancing Robot,*” *Global Journal of Researches in Engineering (J)*, vol. 19, no. 1, 2019.
- [6] G. D. F. D. S. Edwin C. Mone, “*Skema Kendali Ball-Balancing Robot Secara Nirkabel,*” *Jurnal JNTETI*, vol. 3, no. 2, pp. 1-2, 2014.
- [7] T. B. Lauwers, G. A. Kantor dan R. L. Hollis, “*A Dynamically Stable Single-Wheeled Mobile Robot with Inverse Mouse-Ball Drive,*” *IEEE*, pp. 2884-2889, 2006.
- [8] M. Kumagai dan T. Ochiai, “*Development of a Robot Balancing on a Ball,*” dalam *International Conference on Control*, COEX, Seoul, 2008.

- [9] M. Kumagai dan T. Ochiai, “*Development of a Robot Balanced on a Ball – First Report, Implementation of the Robot and Basic Control –*,” *Journal of Robotics and Mechatronics*, vol. 22, no. 3, pp. 349-351, 2010.
- [10] T. Emura, M. Kumagai dan K. Ogawa, “*Expansion of Frequency Response Using Multiple Sensors and Subtraction Type of Filter*,” *Proc. Machatronics*, vol. 96, no. 2, pp. 197-202, 1996.
- [11] M. Kumagai dan T. Ochiai, “*Development of a robot balanced on a ball – Application of passive motion to transport –*,” *Proc. ICRA 2009*, pp. 4106-4111, 2009.
- [12] S. A. A. Ilham dan M. I. Iftitah, “Penggunaan Kontrol PID dengan Berbagai Metode Untuk Analisis Pengaturan Kecepatan Motor DC,” *Jurnal Fisika dan Terapannya*, vol. 7, no. 1, pp. 78-79, 2020.
- [13] I. Setiawan, *Kontrol PID Untuk Proses Industri*, Jakarta: Flex Media Komputindo, 2008, pp. 69-98.
- [14] A. Yani, “Pembelajaran Perancangan Sistem Kontrol PID Dengan Menggunakan *Software* Matlab,” *Jurnal SAINTIKOM*, vol. 13, no. 1, pp. 57-59, 2014.
- [15] F. Palaha, J. Teles dan Y. , “Analisa Rancangan Keseimbangan Menggunakan Sensor IMU *Type*–MPU6050 pada *Quadcopter*,” *JURNAL SAINSTEK STT PEKANBARU*, vol. 8, no. 2, p. 98, 2020.
- [16] A. Faudin, "NyebarIlmu.com," 21 Januari 2019. [Online]. Available: <https://www.nyebarilmu.com/tutorial-arduino-mengakses-module-accelorometer-gyroscope-mpu6050/>. [Diakses 9 April 2021].
- [17] V. R. Alma'i, W. dan I. Setiawan, “Aplikasi Sensor *Accelerometer* Pada Deteksi Posisi,” *ResearchGate*, 2016.

- [18] R. Zhi, "A Drift Eliminated Attitude & Position Estimation Algorithm In 3D," *In Partial Fulfillment of the Requirements for the Degree of Master of Science Specializing in Electrical Engineering*, p. 12, 2016.
- [19] E. A. Prastyo, "Arduino Indonesia," 5 Januari 2019. [Online]. Available: <https://www.arduinoindonesia.id/2019/01/arduino-mega-2560.html>. [Diakses 30 Agustus 2021].
- [20] Y. Z. Maulana dan H. Pujiharsono, "Perbandingan Kinerja Pengontrol PID menggunakan Antarmuka OPC pada PLC dan MATLAB untuk Sistem Pasteurisasi Susu," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika.*, vol. 9, no. 2, p. 436, 2021.

LAMPIRAN 1

DAFTAR RIWAYAT HIDUP

DAFTAR RIWAYAT HIDUP

1. Data Diri

Nama lengkap : Muhammad Fauzan
Tempat, tanggal lahir : Bekasi, 8 Oktober 2000
Alamat rumah : Jl. Angsana III No.55 Perumnas
Bukit Merapin, Pangkalpinang
Telp : -
Hp : 081213068319
Email : excfauzan@gmail.com
Jenis kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

SD Negeri 11 Bojong Rawalumbu, Bekasi (2006 - 2008)
SD Negeri 4 Padang Baru, Bangka Tengah (2008 - 2010)
SD Negeri 55 Pangkalpinang (2010 - 2012)
SMP Negeri 3 Pangkalpinang (2012 - 2015)
SMK Negeri 2 Pangkalpinang (2015 - 2018)

3. Pendidikan Non Formal

-

Sungailiat, 14 Februari 2022

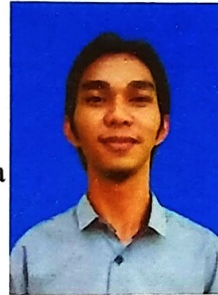


Muhammad Fauzan

DAFTAR RIWAYAT HIDUP

1. Data Diri

Nama lengkap : Muhammad Ferdiansyah
Tempat, tanggal lahir : Pangkalpinang, 9 Oktober 2000
Alamat rumah : Jl. Al Hayati 1, No.7, Desa Kace
Timur, Mendo Barat, Kab. Bangka
Telp : -
Hp : 085381140091
Email : mferdiansyah0910@gmail.com
Jenis kelamin : Laki-laki
Agama : Islam




2. Riwayat Pendidikan

SD Negeri 23 Mendo Barat (2006 - 2012)
SMP Negeri 1 Pangkalpinang (2012 - 2015)
SMK Negeri 2 Pangkalpinang (2015 - 2018)

3. Pendidikan Non Formal

-

Sungailiat, 14 Februari 2022



Muhammad Ferdiansyah