

# Pengembangan Teknik Navigasi Robot Berbasis PID untuk Robot Bergerak Menggunakan Mikrokontroler STM32F401

Muhammad Adri Yusril Izza, Dion Hayu Fandiantoro dan Arief Kurniawan  
Departemen Teknik Komputer, Institut Teknologi Sepuluh Nopember (ITS)  
*e-mail:* dion@its.ac.id; arifku@ee.its.ac.id

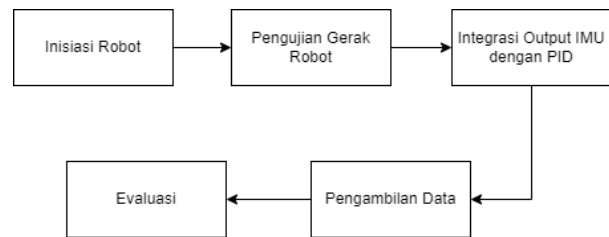
**Abstrak**—Robot adalah sebuah sistem yang menarik, fungsinya banyak serta menghibur. Oleh karena itu, penelitian di bidang teknologi robot menjadi topik yang memiliki daya tarik yang cukup tinggi bagi para peneliti. Terdapat salah satu masalah untuk merancang sebuah robot bergerak, yaitu bagaimana cara robot bernavigasi untuk mencapai tujuan. Problem ini disebut dengan sistem navigasi robot. Kontrol PID (*Proporsional-Integral-Derivatif*) pada robot bergerak adalah pendekatan yang umum digunakan untuk mengendalikan pergerakan robot secara presisi. Pergerakan robot biasanya di atur oleh keluaran dari sensor. Data dari sensor dihasilkan dari sensor IMU dan sensor penggerak robot yang mengukur kecepatan angular dari roda-roda robot. Kemudian, dengan memanfaatkan hukum gaya-gaya, momen inersia dari konstruksi robot serta hasil matematis dari integrasi pergerakan robot didapatkan posisi dan orientasi robot. Tentu akan muncul adanya error pada teknik ini yang berbasis pada sensor IMU. Oleh karena itu dibuatlah model pergerakan robot yang akan mereduksi error dari sensor agar robot dapat bergerak sesuai dengan jalur yang telah ditentukan. Penelitian ini menawarkan proses perancangan sebuah robot bergerak dengan penentuan lokasi berasarkan teknik tersebut dan akan diimplementasikan pada STM32F401.

**Kata Kunci**—Robot, PID, STM32F401.

## I. PENDAHULUAN

Di era modern ini atau bisa juga disebut era industri 4.0, semua sektor mengalami modernisasi dan digitalisasi. Di sektor industri sudah banyak pabrik atau kantor – kantor yang menggantikan jasa manusia menggunakan robot untuk melakukan sesuatu pekerjaan. Salah satu nya penggunaan Robot bergerak untuk membantu pekerjaan manusia di berbagai bidang, seperti bidang industrial, bidang militer, bidang konvensional dan bidang pendidikan. Robot dapat mengerjakan pekerjaan dengan sangat efektif dan dapat mengurangi resiko kecelakaan kerja. Salah satu contoh nya adalah robot pengantar barang di gudang atau pabrik. Cara kerja robot ini adalah dengan bergerak secara autonomous dengan bantuan berbagai sensor.

Selama ini kebanyakan teknologi mobile robot menggunakan sistem pergerakan differential drive. Permasalahan utama dari sistem pergerakan differential drive adalah terbatasnya pergerakan robot sehingga kurang efektif dan efisien. Roda mekanum menjawab permasalahan tersebut karena dapat bergerak kesegala arah. Robot differential driver merupakan suatu robot mobile yang memiliki sebuah roda disisi kanan dan kirinya yang tidak akan seimbang apabila tanpa adanya kontroler. Dengan menggunakan kontrol biasa open loop pada masing-masing roda, perpindahan robot menjadi tidak lurus.



Gambar 1. Alur kerja Penelitian

Agar robot dapat bergerak lurus, robot dengan empat roda membutuhkan sebuah control yang dapat mengatur kecepatan dan arah tiap roda. Ada berbagai solusi untuk membantu dalam kontrol pergerakan robot salah satunya adalah dengan menggunakan metode PID (*Proportional-Integral-Derivative*) dan IMU.

PID (*Proportional-Integral-Derivative*) adalah sebuah metode kontrol otomatis yang digunakan dalam sistem kontrol untuk mengatur variabel proses. Ini melibatkan tiga komponen utama: Proporsional (P), Integral (I), dan Derivatif (D). Metode PID digunakan untuk meminimalkan kesalahan antara nilai setpoint dan nilai aktual suatu variabel proses. Dengan menerapkan sistem ini, robot dapat mengontrol pergerakannya dengan mengatur kecepatan dari motor penggeraknya agar pergerakan robot stabil dan dapat bergerak sesuai dengan jalur atau tujuannya. biasanya menggunakan sensor IMU untuk menentukan Orientasi dalam navigasi. Namun keluaran dari sensor IMU memiliki noise atau error. Oleh sebab itu dibuatlah suatu implementasi teknik PID dan Kalman filter untuk mengurangi noise yang terdapat pada IMU sensor sehingga sistem navigasi menjadi lebih baik.

## II. PENELITIAN TERDAHULU

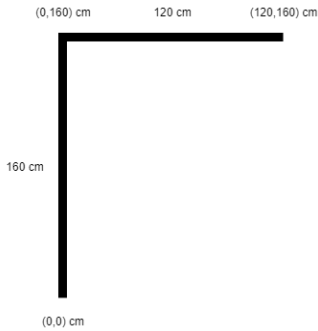
### A. Sistem Pengendali Robot Bergerak Lurus dengan Kontrol Proporsional Integral Derivatif (PID) Berbasis LabView

Penelitian yang dilakukan oleh Andrean George W dan Abdur Rahman pada tahun 2015, mengungkapkan bahwa pengendalian dan monitoring kecepatan putar suatu roda (motor DC) dalam sebuah sistem proses sangat penting perannya dalam implementasi industri [1]. Pengendalian dan monitoring PWM untuk kecepatan putar roda pada sepasang motor DC ini menggunakan perangkat antarmuka komputer dimana dalam industri hal ini diperlukan untuk memudahkan operator dalam mengendalikan dan memonitor kecepatan motor.

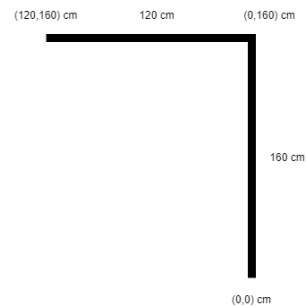
Agar diperoleh pengontrol yang terbaik, maka dilakukan tuning parameter pengontrol Proporsional Integral Derivatif



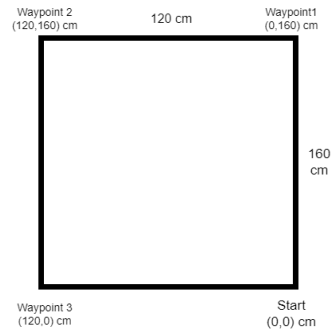
Gambar 2. Jalur Lurus.



Gambar 3. Jalur Lurus dan Belok Kanan



Gambar 4. Jalur Lurus dan Belok Kiri.



Gambar 5. Jalur Persegi.

(PID). Dalam tuning ini kita dapat mengetahui nilai dari proporsional gain ( $K_p$ ), waktu integral ( $T_i$ ) dan waktu derivative ( $T_d$ ). Pengontrol PID akan memberikan aksi kepada kontrol motor DC berdasarkan error yang diperoleh, nilai putaran motor DC yang diinginkan disebut dengan set point. Software LabVIEW digunakan sebagai pemonitor, kendali kecepatan motor.

**B. Movement Control of Two Wheels Balancing Robot using Cascaded PID Controller**

Penelitian yang dikakukan oleh Pratama, D., Binugroho, E.H., Ardilla, F. pada tahun 2015, mengungkapkan bahwa robot seimbang yang diusulkan dalam makalah ini adalah robot yang mengandalkan dua roda dalam proses pergerakannya [2]. Berbeda dengan robot mobile lain yang stabil secara mekanis dalam posisi berdirinya, robot seimbang membutuhkan kontrol seimbang yang memerlukan nilai sudut sebagai umpan balik kemiringan. Kontrol seimbang ini akan mengendalikan robot agar dapat mempertahankan posisi berdirinya. Selain kontrol seimbang itu sendiri, pergerakan robot seimbang memerlukan kontrolnya sendiri untuk mengontrol pergerakan sambil menjaga keseimbangan robot.

Kedua pengendali ini akan digabungkan karena keduanya mengontrol roda yang sama sebagai aktuator. Dalam makalah ini, kami mengusulkan algoritma kontrol PID berurutan untuk menggabungkan kontrol seimbang dan pengendali pergerakan atau jarak.

**III. METODOLOGI**

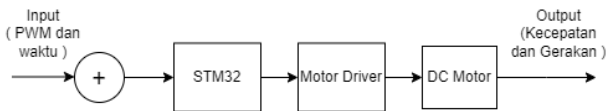
Dalam proses pengerjaannya, model yang digunakan untuk menjadi inti dari mekanisme pengembangan teknik navigasi berbasis kontroler PID akan dibuat melalui beberapa tahap seperti yang ditunjukkan pada Gambar 1. Proses dimulai dengan inisiasi robot yang akan digunakan sebagai objek penelitian. Dilanjutkan dengan proses pengujian gerak robot yang akan dipakai untuk proses pengambilan data.

Setelah pengujian robot telah selesai, dilakukan proses integrasi antara sensor IMU dan PID Controller, kedua tahapan ini dapat secara general disebut sebagai tahap inisiasi. Pada akhir proses tersebut akan dihasilkan sistem navigasi robot berbasis output dari IMU sensor untuk memprediksi keakuratan posisi robot saat bergerak mengikuti jalur dan pada step terakhir pengerjaannya akan dilakukan evaluasi untuk menguji proses yang dihasilkan. Berikut adalah metodologi sistem yang dibuat.

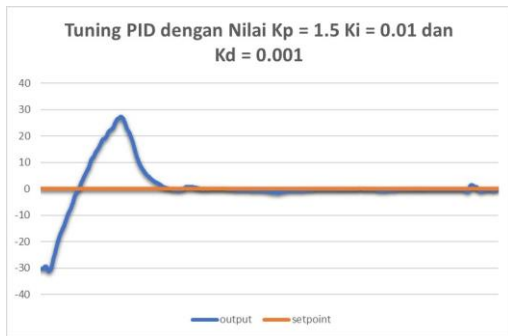
Pada Gambar 1 alur kerja tersebut menunjukkan proses kerja sistem robot. Proses kerja tersebut dapat dibagi menjadi beberapa tahap, yaitu: (1) *Tahap Inisiasi*, tahap inisiasi merupakan tahap awal dari proses kerja sistem robot. Pada tahap ini, robot akan diinisialisasi dengan cara menyiapkan semua komponen dan perangkat yang diperlukan untuk bekerja. (2) *Tahap Pengujian*, tahap pengujian merupakan tahap untuk menguji apakah robot dapat bekerja dengan baik sesuai dengan spesifikasi yang diinginkan. Pengujian ini dilakukan dengan cara memberikan input tertentu ke robot dan kemudian memeriksa output yang dihasilkan. (3) *Integrasi*, integrasi output IMU, yang dilakukan untuk menggabungkan output dari sensor IMU dengan output dari pengendali. Sensor IMU (*Inertial Measurement Unit*) merupakan sensor yang digunakan untuk mengukur pergerakan robot. Integrasi output IMU dengan output dari pengendali diperlukan untuk meningkatkan akurasi kontrol robot. (4) *Pengambilan Data*, tahap ini akan dilakukan proses pengambilan data penyimpangan posisi akhir robot pada tiap titik tujuan. (5) *Evaluasi*, pada blok evaluasi, output robot akan dibandingkan dengan hasil yang diinginkan. Jika output robot sesuai dengan hasil yang diinginkan, maka robot dapat dikatakan berfungsi dengan baik. Namun, jika output robot tidak sesuai dengan hasil yang diinginkan, maka perlu dilakukan perbaikan pada robot.

**A. Inisiasi Robot**

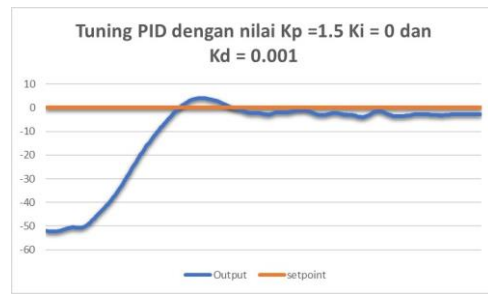
Pada tahap ini, proses perakitan dan inisiasi robot yang akan digunakan dalam fase pengujian dilakukan. Alat-alat



Gambar 6. Diagram Blok Pengujian dengan Metode Open Loop.



Gambar 7. Tuning Percobaan 1.



Gambar 8. Tuning Percobaan 2.



Gambar 9. Tuning Percobaan 3

yang digunakan akan dirakit dan diprogram ke dalam perangkat pengujian, membentuk robot yang dimodifikasi yang akan berfungsi sebagai objek penelitian. Robot ini terdiri dari beberapa komponen seperti sasis 4WD, empat motor DC, pengendali motor L298N, STM32F401xx, baterai atau power bank, dan sensor IMU. Selama tahap ini, juga akan dilakukan instalasi beberapa perangkat lunak yang diperlukan, termasuk instalasi perangkat lunak, kalibrasi sensor IMU, dan pemrograman robot yang akan digunakan. Begitu robot yang dimodifikasi ini dirakit dan diinstal, maka akan digunakan untuk tahap berikutnya, yaitu fase Pengujian Robot.

**B. Pengujian**

**1) Pengujian Gerak Robot Pada Jalur Lurus**

Pengujian pertama merupakan evaluasi performansi penentuan navigasi robot saat menjalankan gerakan lurus. Dalam rangka pengujian ini, robot ditempatkan pada titik awal koordinat kartesius (0,0) cm pada sumbu (X,Y).

Pada Gambar 2 menunjukkan lintasan robot yang bergerak mengikuti garis lurus. Lintasan tersebut memiliki panjang 160 cm dan digambarkan dengan angka-angka pada gambar. Angka-angka tersebut menunjukkan jarak robot dari titik awal. Berdasarkan gambar tersebut, pergerakan robot dapat dijelaskan sebagai berikut:

1. Robot memulai pergerakannya dari titik (0,0) cm.
2. Robot bergerak ke arah sumbu x positif.
3. Robot bergerak dengan kecepatan konstan.
4. Robot berhenti pada titik (0,160) cm

**2) Pengujian Gerak Robot Pada Jalur Lurus dan Belok Kanan**

Pengujian kedua ini merinci serangkaian pergerakan yang lebih kompleks untuk menguji kemampuan robot. Awalnya, robot ditempatkan pada titik koordinat (0,0) cm dalam sistem kartesius (X,Y). Setelah mencapai fase pergerakan lurus, robot kemudian diuji dengan tujuan mencapai titik pertama pada koordinat (0,160) cm. Selama perjalanan ini, kemampuan robot untuk berbelok ke kiri akan diuji, memeriksa responsnya terhadap instruksi untuk mengubah arah dan mencapai koordinat yang ditentukan.

Pada Gambar 3 tersebut menunjukkan lintasan robot yang bergerak sepanjang garis lurus. Panjang lintasan adalah 160 cm, dengan titik awal di (0,0) cm dan titik akhir di (120,160) cm. Berikut adalah detail pergerakan robot:

1. Robot dimulai dari titik (0,0) cm.
2. Robot bergerak sepanjang garis lurus hingga titik (0,160) cm,
3. Kemudian robot berbelok kanan
4. Robot menempuh jarak 120 cm.
5. Robot tiba di titik (120,160) cm

**3) Pengujian Gerak Robot Pada Jalur Lurus dan Belok Kiri**

Pengujian ketiga melibatkan serangkaian pergerakan yang kompleks untuk menguji kemampuan robot dalam berbelok ke kiri pada suatu kemiringan tertentu. Awalnya, robot ditempatkan pada titik koordinat (0,0) cm dalam sistem kartesius (X,Y), dengan kecepatan roda kiri dan roda kanan diatur sedemikian rupa sehingga robot dapat menjalankan pergerakan lurus. Setelah mencapai fase pergerakan lurus, robot memiliki tujuan untuk mencapai titik pertama pada koordinat (0,-160) cm.

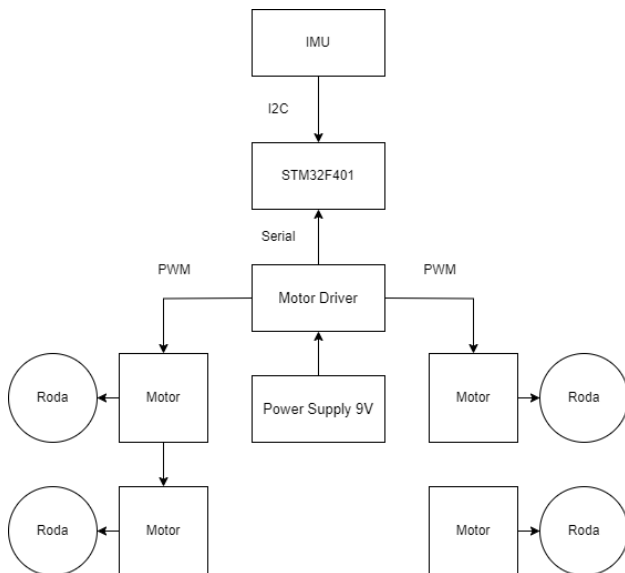
Lintasan tersebut memiliki panjang 160 cm dan lebar 120 cm. Robot mulai bergerak dari titik (0,0) cm dan bergerak sepanjang garis lurus hingga mencapai titik (160,0) cm, kemudian robot bergerak ke kiri hingga mencapai titik (120,160) cm. Pergerakan robot tersebut dapat digambarkan seperti pada Gambar 4.

Pada Gambar 4 tersebut menunjukkan lintasan robot yang bergerak sepanjang garis lurus. Panjang lintasan adalah 160 cm, dengan titik awal di (0,0) cm dan titik akhir di (120,160) cm. Berikut adalah detail pergerakan robot:

1. Robot mulai bergerak dari titik (0,0) cm.
2. Robot terus bergerak sepanjang garis lurus hingga mencapai titik (160,0) cm.
3. Robot berhenti di titik (160,0) cm.
4. Robot berbelok ke arah kiri dan bergerak lurus hingga titik (120,160) cm.

**4) Pengujian Gerak Robot Pada Jalur berbentuk Persegi**

Pengujian keempat melibatkan serangkaian pergerakan yang kompleks untuk menguji kemampuan robot. Awalnya, robot ditempatkan pada titik koordinat (0,0) cm dalam sistem kartesius (X,Y), dengan kecepatan roda kiri dan roda kanan diatur sedemikian rupa sehingga robot dapat menjalankan pergerakan lurus. Setelah mencapai fase pergerakan lurus,



Gambar 10. Sketsa Robot untuk Pengambilan Data.

robot memiliki tujuan untuk mencapai titik pertama pada koordinat (0,160) cm, kemudian robot bergerak ke titik (120,160) cm, kemudian bergerak kembali ke titik (120,0) cm dan terakhir robot bergerak ke titik awal yaitu (0,0) cm.

Pada Gambar 5 menunjukkan lintasan robot yang bergerak di sepanjang persegi dengan panjang 160 cm dan lebar 120 cm. Waypoint 1 terletak di titik (0,160) cm, waypoint 2 terletak di titik (120,160) cm, waypoint 3 terletak di titik (120,0) cm dan waypoint 4 pada titik (0,0) cm. Pergerakan robot dapat digambarkan sebagai berikut:

1. Robot mulai bergerak dari titik (0,0) cm menuju waypoint 1
2. Robot bergerak ke kiri sepanjang garis lurus hingga mencapai titik (120,160) cm.
3. Robot kemudian berbelok kiri ke bawah sepanjang garis lurus hingga mencapai titik (120,0) cm.
4. Robot kemudian berbelok kiri sepanjang garis lurus dan robot berhenti di titik (0,0) cm.

C. Pengujian Pergerakan Robot dengan Metode Open Loop

Pada tahapan ini dilakukan pengujian pergerakan robot menggunakan robot modifikasi yang terdiri dari STM32F401, motor driver L298N, 4 buah motor DC dan 2 buah baterai 18650. Ada 2 parameter yang di jadikan poin khusus untuk pergerakan robot yaitu kecepatan motor dan waktu, 2 hal ini yang dijadikan parameter robot bergerak mengikuti jalurnya. Kecepatan motor dan waktu di kalibrasi agar robot dapat bergerak akurat.

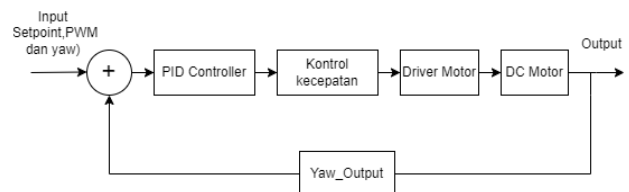
Pada Gambar 6 Diagram blok tersebut menunjukkan sistem penggerak motor DC yang terdiri dari tiga blok utama, yaitu:

1) Input

Blok input menerima sinyal masukan dari luar sistem, yaitu: PWM (*Pulse-width modulation*) dan Waktu. Sinyal PWM digunakan untuk mengontrol kecepatan putaran motor, sedangkan sinyal waktu digunakan untuk mengontrol waktu putar motor.

2) Proses

Blok proses memproses sinyal masukan dari blok input dan menghasilkan sinyal keluaran untuk menggerakkan motor. Proses ini dilakukan oleh *driver motor*. *Driver motor*



Gambar 11. Diagram Blok sistem kontrol PID

Tabel 1.

Perbandingan Rata - rata penyimpangan mutlak (Cm)		
Metode	Sumbu X	Sumbu Y
<i>Open Loop</i>	3,21 cm	0,78 cm
<i>PID</i>	1,12 cm	0,56 cm

Tabel 2.

Perbandingan Rata - rata penyimpangan mutlak (Cm)				
Koordinat (cm)	Sumbu X Open Loop	Sumbu Y Open Loop	Sumbu X PID	Sumbu Y PID
(0,160)	2,5 cm	0 cm	1,09 cm	0 cm
(120,160)	2,25 cm	7,78 cm	1,31 cm	3,71 cm

mengubah sinyal PWM dan waktu menjadi sinyal kendali yang dapat menggerakkan motor.

3) Output

Blok output memberikan sinyal keluaran dari sistem, yaitu: Kecepatan dan arah putaran motor. Sinyal kecepatan dan arah putaran motor digunakan untuk menggerakkan motor. Kecepatan dan arah putaran motor ditentukan oleh sinyal kendali dari driver motor. Kecepatan putaran motor ditentukan oleh lebar pulsa sinyal PWM. Semakin lebar pulsa sinyal PWM, maka kecepatan putaran motor semakin cepat. Arah putaran motor ditentukan oleh polaritas sinyal kendali. Jika polaritas sinyal kendali positif, maka motor akan berputar searah jarum jam. Jika polaritas sinyal kendali negatif, maka motor akan berputar berlawanan arah jarum jam.

D. Integrasi Output Sensor IMU dengan PID

Kontrol PID adalah metode umpan balik yang umum digunakan dalam otomatisasi untuk mengatur sistem sesuai dengan target. Ini terdiri dari tiga komponen utama: P (Proportional), I (Integral), dan D (Derivative). P merespons selisih saat ini, I mengatasi kesalahan tetap seiring waktu, dan D mengendalikan laju perubahan kesalahan. Sistem ini digunakan dalam berbagai aplikasi untuk mengoptimalkan performa sistem. Input pengontrol adalah sinyal kesalahan dan output diberikan ke proses.

Sinyal keluaran pengendali dibangkitkan sedemikian rupa sehingga keluaran dari plant berusaha untuk mencapai nilai yang diinginkan. Kontroler PID adalah sistem loop tertutup yang memiliki sistem kontrol umpan balik dan membandingkan variabel Proses (variabel umpan balik) dengan set Point dan menghasilkan sinyal kesalahan dan karenanya menyesuaikan output sistem. Proses ini berlanjut hingga kesalahan ini mencapai nol atau nilai variabel proses menjadi sama dengan titik setel.

$$Output = K_p \cdot Error + K_i \cdot \int Error dt + K_d \cdot dError \quad (1)$$

K<sub>p</sub>, K<sub>i</sub> dan K<sub>d</sub> adalah bobot (koefisien) proporsional, integral, dan derivatif masing-masing. Error adalah selisih antara setpoint dan nilai aktual sistem, integral Error dt adalah

Tabel 3.

Perbandingan Rata - rata penyimpangan mutlak (Cm)

Koordinat (cm)	Sumbu X Open Loop	Sumbu Y Open Loop	Sumbu X PID	Sumbu Y PID
(0,160)	1,56 cm	3,81 cm	2 cm	0 cm
(120,160)	3,37 cm	11,12 cm	0,93 cm	3,5 cm

integral dari kesalahan sepanjang waktu.  $dError/dt$  adalah turunan dari kesalahan terhadap waktu.

E. Tuning PID

Tuning PID (Proporsional, Integral, Derivatif) adalah proses penyesuaian parameter PID pada sistem kontrol untuk mencapai kinerja yang diinginkan. PID adalah metode kontrol umum yang digunakan dalam otomatisasi untuk memastikan bahwa sistem dapat mengikuti setpoint atau merespons gangguan dengan baik. Proses tuning PID melibatkan penyesuaian tiga komponen utama:

Proporsional (P), Integral (I), dan Derivatif (D). Pada penerapan ini akan menggunakan metode trial and error. trial and error adalah pendekatan praktis yang melibatkan eksperimen dan pengamatan untuk menemukan parameter PID yang optimal untuk sistem kontrol tertentu [3]. Berikut adalah langkah-langkah umum yang dapat diikuti dalam metode ini:

1) Inisiasi Parameter Awal

Tetapkan nilai awal untuk  $K_p$  (proporsional),  $K_i$  (integral), dan  $K_d$  (derivatif). Mulailah dengan nilai-nilai yang kecil dan tingkatkan secara bertahap. Uji Coba Pertama.

2) Implementasikan kontrol PID pada sistem.

Amati respons sistem terhadap perubahan setpoint atau gangguan. Jika respons sistem terlalu lambat, tingkatkan nilai  $K_p$ .

3) Optimasi Integral ( $K_i$ )

Jika terdapat offset atau kesalahan statis, tingkatkan nilai  $K_i$ . Perhatikan agar tidak terlalu tinggi yang dapat menyebabkan overshoot atau instability.

4) Optimasi Derivatif ( $K_d$ )

Jika ada overshoot atau osilasi yang berlebihan, tambahkan komponen derivatif (tingkatkan nilai  $K_d$ ). Hindari nilai  $K_d$  yang terlalu tinggi yang dapat menyebabkan sistem menjadi tidak stabil.

5) Iterasi dan Pengecekan Lanjutan

Lakukan uji coba berulang dengan memodifikasi parameter PID. Perhatikan respons sistem terhadap setiap perubahan.

6) Optimasi Keseluruhan

Sesuaikan parameter secara bersamaan untuk mencapai keseimbangan yang optimal antara respons cepat, tanpa osilasi berlebihan, dan tanpa kesalahan statis yang signifikan.

Telah dilakukan proses trial and error sebanyak 3 kali percobaan dengan nilai parameter  $K_p, K_i$  dan  $K_d$  yang berbeda.

Gambar 7 menunjukkan respon output dari sistem kontrol PID dengan nilai parameter  $K_p = 1.5, K_i = 0.01, K_d = 0.001$ . Grafik tersebut memiliki dua garis, yaitu garis biru yang menunjukkan output dari sistem kontrol PID dan garis

Tabel 4.

Perbandingan Rata - rata penyimpangan mutlak (Cm)

Koordinat (cm)	Sumbu X Open Loop	Sumbu Y Open Loop	Sumbu X PID	Sumbu Y PID
(0,160)	2,12 cm	1,31 cm	1,40 cm	1,93 cm
(120,160)	3,25 cm	12,12 cm	2,65 cm	3,62 cm
(120,0)	20 cm	6,87 cm	1,40 cm	4,75 cm
(0,0)	8,28 cm	10 cm	3,12 cm	3,87 cm

oranye yang menunjukkan setpoint dari sistem kontrol PID. Pada awal Gambar 7, output dari sistem kontrol PID berada di bawah setpoint. Hal ini menunjukkan bahwa terdapat kesalahan (error) antara output dan setpoint. Sistem kontrol PID kemudian akan memberikan sinyal kontrol untuk mengurangi kesalahan tersebut.

Gambar 8 menunjukkan respon output dari sistem kontrol PID dengan nilai parameter  $K_p = 0.5, K_i = 0.01, K_d = 0.001$ . Grafik tersebut memiliki dua garis, yaitu garis biru yang menunjukkan output dari sistem kontrol PID dan garis oranye yang menunjukkan setpoint dari sistem kontrol PID. Pada awal grafik, output dari sistem kontrol PID berada di bawah setpoint. Hal ini menunjukkan bahwa terdapat kesalahan (error) antara output dan setpoint. Sistem kontrol PID kemudian akan memberikan sinyal kontrol untuk mengurangi kesalahan tersebut.

Gambar 9 menunjukkan hasil tuning PID dengan nilai  $K_p = 1.3, K_i = 0.1, K_d = 0.001$ . Grafik tersebut menunjukkan bahwa output PID dapat mencapai nilai setpoint dengan cepat dan stabil. Pada awal grafik, output PID berada di bawah nilai setpoint. Hal ini menunjukkan bahwa sistem memiliki kesalahan positif. Kesalahan positif tersebut kemudian dihilangkan oleh PID dengan cara memberikan sinyal kontrol yang sesuai.

Dari ketiga grafik dapat disimpulkan bahwa Secara umum, Gambar 7 menunjukkan respon output dari sistem kontrol PID yang cukup baik. Rise time yang dibutuhkan cukup cepat, overshoot yang dihasilkan tidak terlalu besar, dan settling time yang dibutuhkan juga cukup singkat.

Sedangkan, Gambar 8 menunjukkan respon output dari sistem kontrol PID yang cukup baik. Rise time yang dibutuhkan cukup cepat, overshoot yang dihasilkan tidak terlalu besar, dan settling time yang dibutuhkan juga cukup singkat.

Dan Gambar 9 menunjukkan respon output dari sistem kontrol PID yang cukup baik. Rise time yang dibutuhkan cukup cepat, overshoot yang dihasilkan tidak terlalu besar, dan settling time yang dibutuhkan juga cukup singkat.

Maka dapat disimpulkan dari ketiga gambar, Gambar 8 mengalami overshoot paling kecil dibandingkan dengan percobaan lainnya. seta rise time juga cukup cepat dan settling cukup singkat. Oleh sebab itu untuk pengujian gerak robot akan menggunakan nilai parameter  $K_p = 1.5, K_i = 0$  dan  $K_d = 0.001$ .

F. Pengambilan Data dengan Kontrol PID

Pada tahap ini akan dilakukan pengujian pergerakan robot sepanjang jalur yang sudah disiapkan. Akan disiapkan sebuah jalur untuk robot, robot akan bergerak setelah sensor mendeteksi jalur yang ada, robot akan bergerak dan mengikuti jalur. Pada tahap ini akan diambil data berupa

koordinat posisi robot pada saat mulai hingga sampai di tujuan yang telah ditentukan. Selain itu data waktu dan untuk sketsa robot untuk pengambilan data bisa dilihat pada Gambar 10.

Kemudian akan di terapkan metode kontrol close loop. Sistem kontrol loop tertutup (closed-loop) memungkinkan self-correction terhadap output dengan menggunakan sinyal feedback untuk mengukur perbedaan antara nilai yang diinginkan (setpoint) dan nilai aktual (output). Sinyal error yang dihasilkan dari perbedaan ini digunakan sebagai input untuk kontroler, yang menghitung Actuating Signal atau sinyal kontrol. Sinyal kontrol ini dikirim ke plant (blok sistem) untuk menghasilkan output sistem dengan error yang semakin kecil dari sebelumnya. Proses koreksi ini berlanjut hingga error mencapai nilai minimum, sehingga output sesuai dengan setpoint [4]. Keuntungan utama dari sistem ini adalah mengurangi sensitivitas terhadap gangguan eksternal.

Pada Gambar 11 ini menjelaskan bahwa plant terdiri dari driver motor yang kemudian akan menggerakkan DC Motor. DC motor dapat berputar karena di pengaruhi oleh sinyal PWM dan PID Controller dari driver motor. Dalam sistem ini setiap kali looping sistem error akan di perbarui dengan data yang diambil dari sensor IMU yang sebelumnya disimpan pada last error. Lalu dilakukan perhitungan matematis menggunakan rumus PID dan akan di dapatkan output yang akan digunakan untuk menggerakkan Motor DC.

$$\text{Error} = SP - PV \quad (2)$$

$$PWMMOTOR = BAS EPWM + PID \quad (3)$$

Setpoint (SP) adalah parameter acuan atau nilai yang diinginkan. present value (PV) adalah nilai pembacaan sensor atau variabel yang terukur yang di keluarkan oleh sensor. Setelah itu keluaran ini akan digunakan untuk pengujian pergerakan pada jalur yang sama seperti pada tahap pengujian sebelumnya [5].

#### G. Evaluasi

Data yang sudah didapatkan pada tahap sebelumnya nantinya akan di inputkan ke dalam perhitungan untuk mengoreksi hasil pengujian. Hasil evaluasi akhir didapat ketika robot sudah mencapai titik tujuan setelah kondisi terpenuhi semua nilai dari setiap tahapan akan dirata-rata untuk mendapatkan hasil akhir evaluasi. Tahap evaluasi ini menggunakan metode Mean Absolute Error (MAE).

Mean Absolute Error (MAE) atau Rata-Rata Kesalahan Mutlak (MAD) adalah sebuah metrik evaluasi yang digunakan untuk mengukur sejauh mana nilai-nilai prediksi berbeda dari nilai sebenarnya dalam suatu rangkaian data. Metrik ini memberikan gambaran rata-rata besarnya kesalahan prediksi, diukur dalam satuan yang sama dengan data yang dievaluasi.

Untuk data yang dipakai untuk mengevaluasi pergerakan adalah dengan menghitung rata - rata penyimpangan mutlak. untuk mendapatkan hasil rata - rata penyimpangan mutlak digunakan rumus matematis sebagai berikut:

$$\text{Rata-rata Penyimpangan} = \frac{1}{n} \sum_{i=1}^n |X_i - \bar{X}| \quad (4)$$

Setelah mendapatkan nilai rata - rata penyimpangan tiap sumbu X dan Y maka selanjutnya akan dilakukan perbandingan antara pergerakan robot dengan metode open loop dengan pergerakan robot berbasis PID.

## IV. HASIL

Pada penelitian ini dipaparkan hasil pengujian serta analisis dari desain sistem dan implementasi. Pengujian ini dilakukan untuk mengetahui penerapan penelitian ini, sehingga nantinya dapat ditarik kesimpulan. Skenario yang nantinya akan diterapkan pada pengujian meliputi beberapa poin sebagai berikut: (1) Pengujian pergerakan robot dengan menggunakan metode open loop. (2) Pengujian dan pengambilan data sensor IMU dari robot. (3) Pengujian pergerakan dengan Control PID dan IMU. (4) Evaluasi pengujian

### A. Pengujian Gerakan Robot Pada Jalur Lurus

Delapan eksperimen telah dilakukan untuk menilai pergerakan robot sepanjang jalur lurus menggunakan metode open-loop dan metode PID. Dalam eksperimen ini, data mengenai penyimpangan robot selama pergerakannya menuju titik-titik tertentu dicatat. Data mengenai penyimpangan gerakan robot dapat dilihat pada Tabel 1.

Dari Tabel 1 dapat dilihat beberapa perbedaan antara deviasi kedua metode dimana pada metode pertama pada koordinat sumbu X Memiliki perbedaan dengan nilai metode kedua.

#### 1) X axis

Metode PID menunjukkan kinerja yang lebih baik pada sumbu X dengan rata-rata penyimpangan mutlak yang lebih rendah dibandingkan dengan metode Open Loop.

#### 2) Y axis

Tidak ada perbedaan yang signifikan antara metode Open Loop dan PID pada sumbu Y, karena keduanya memiliki rata-rata penyimpangan mutlak yang hampir sama.

Metode PID cenderung memberikan performa yang lebih baik daripada metode Open Loop, terutama pada sumbu X. Perlu diperhatikan bahwa pada sumbu Y, keduanya memiliki rata-rata penyimpangan mutlak yang rendah, sehingga perbedaan antara keduanya mungkin tidak signifikan secara praktis.

### B. Pengujian Gerakan Robot Belok Kanan

Delapan eksperimen telah dilakukan untuk menilai pergerakan robot sepanjang jalur lurus menggunakan metode open-loop dan metode PID. Dalam eksperimen ini, data mengenai penyimpangan robot selama pergerakannya menuju titik-titik tertentu dicatat. Data mengenai penyimpangan gerakan robot dapat dilihat pada Tabel 2. Dari Tabel 2 di dapat kesimpulan sebagai berikut:

#### 1) Koordinat (0,160)

Metode 1 memiliki perbedaan yang lebih besar pada sumbu Y dibandingkan dengan Metode 2. Secara umum, Metode 2 menunjukkan kinerja yang lebih baik dalam meminimalkan penyimpangan pada sumbu Y pada koordinat ini.

#### 2) Koordinat (120,160)

Metode 1 memiliki perbedaan yang lebih besar pada kedua sumbu, baik sumbu X maupun sumbu Y. Metode 2, meskipun memiliki perbedaan yang signifikan, masih menunjukkan penyimpangan yang lebih rendah dibandingkan dengan Metode 1 pada koordinat ini.

Dari analisa di atas, dapat disimpulkan bahwa penyesuaian nilai-nilai PID yang dilakukan oleh pengontrol berhasil membuat robot bergerak mengikuti garis hitam dengan lebih akurat. Dengan nilai-nilai PID yang optimal, robot dapat bergerak mengikuti garis hitam dengan penyimpangan yang sangat kecil. Hal ini menunjukkan bahwa algoritma PID sudah mulai bekerja dengan baik. Namun masih terjadi penyimpangan yang cukup kecil pada Metode PID.

### C. Pengujian Gerakan Robot Belok Kiri

Delapan eksperimen telah dilakukan untuk menilai pergerakan robot sepanjang jalur lurus menggunakan metode open-loop dan metode PID. Dalam eksperimen ini, data mengenai penyimpangan robot selama pergerakannya menuju titik-titik tertentu dicatat. Data mengenai penyimpangan gerakan robot dapat dilihat pada Tabel 3. Dari Tabel 3 di dapat kesimpulan sebagai berikut:

#### 1) Pergerakan pada Koordinat (0,160)

Metode 1 memiliki nilai penyimpangan mutlak yang sama dengan nol pada sumbu X dan sumbu Y. Metode 2 memiliki nilai penyimpangan mutlak pada sumbu Y sebesar 0,656 cm pada koordinat ini. Metode 1 menunjukkan pergerakan yang lebih baik dan presisi pada jalur lurus pada koordinat (0,160), sementara Metode 2 menunjukkan sedikit penyimpangan pada sumbu Y.

#### 2) Pergerakan pada Koordinat (120,160)

Metode 1 memiliki nilai penyimpangan mutlak pada sumbu X sebesar 1 cm dan pada sumbu Y sebesar 3,43 cm. Metode 2 memiliki nilai penyimpangan mutlak pada sumbu X sebesar 0,5 cm dan pada sumbu Y sebesar 1,25 cm. Secara umum, Metode 2 menunjukkan performa yang lebih baik dalam mengurangi penyimpangan mutlak pada koordinat (120,160) untuk pergerakan belok kiri, baik pada sumbu X maupun sumbu Y.

Dari analisa di atas, dapat disimpulkan bahwa penyesuaian nilai-nilai PID yang dilakukan oleh pengontrol berhasil membuat robot bergerak mengikuti garis hitam dengan lebih akurat. Dengan nilai-nilai PID yang optimal, robot dapat bergerak mengikuti garis hitam dengan penyimpangan yang sangat kecil.

### D. Pengujian Gerakan Robot Pada Jalur Persegi

Delapan eksperimen telah dilakukan untuk menilai pergerakan robot sepanjang jalur lurus menggunakan metode open-loop dan metode PID. Dalam eksperimen ini, data mengenai penyimpangan robot selama pergerakannya menuju titik-titik tertentu dicatat. Data mengenai penyimpangan gerakan robot dapat dilihat pada tabel 4. Dari tabel 4 di dapat kesimpulan sebagai berikut:

#### 1) Pergerakan pada Koordinat (0,160)

Metode 1 dan Metode 2 sama-sama memiliki nilai penyimpangan mutlak yang nol pada koordinat ini. Kedua metode menunjukkan pergerakan yang sangat baik dan akurat pada titik awal (0,160).

#### 2) Pergerakan pada Koordinat (120,160)

Metode 1 memiliki nilai penyimpangan mutlak yang lebih tinggi pada sumbu X dan sumbu Y dibandingkan dengan Metode 2. Metode 2 menunjukkan performa yang lebih baik dengan nilai penyimpangan mutlak yang lebih rendah pada koordinat ini.

#### 3) Pergerakan pada Koordinat (120,0)

Metode 1 memiliki nilai penyimpangan mutlak yang lebih tinggi pada sumbu X dan sumbu Y dibandingkan dengan Metode 2. Metode 2 kembali menunjukkan performa yang lebih baik dengan nilai penyimpangan mutlak yang lebih rendah pada koordinat ini.

#### 4) Pergerakan pada Koordinat (0,0)

Metode 1 memiliki nilai penyimpangan mutlak yang lebih tinggi pada sumbu X dan sumbu Y dibandingkan dengan Metode 2. Metode 2 masih menunjukkan performa yang lebih baik dengan nilai penyimpangan mutlak yang lebih rendah pada koordinat ini.

### E. Evaluasi

Dapat terlihat dari beberapa percobaan yang telah dilakukan, pada percobaan gerakan robot pada garis lurus metode PID dapat mengurangi nilai penyimpangan yang dialami robot selama bergerak. Hal tersebut juga terjadi pada pergerakan robot lurus dan belok kanan dan kiri. Namun, masih terjadi penyimpangan yang cukup kecil ketika robot mencapai titik akhir tujuan. Lain halnya dengan percobaan pergerakan robot pada jalur persegi, meskipun penyimpangan dapat dikurangi, namun penyimpangan akhir ketika robot mencapai titik tujuan masih bisa dibilang cukup besar.

Dari analisa tersebut, dapat disimpulkan bahwa penyimpangan terhadap garis hitam pergerakan robot dengan PID dapat mengurangi penyimpangan secara signifikan, namun masih terjadi penyimpangan yang masih kecil. Hal ini mungkin disebabkan oleh beberapa faktor diantaranya : (1) Penggunaan komponen robot yang lebih baik sehingga pergerakan yang dihasilkan lebih akurat dan presisi. (2) Melakukan tuning PID dengan metode yang lebih baik, agar sistem PID dapat bekerjadengan maksimal dan akurat.

## V. KESIMPULAN

Berdasarkan hasil pengujian dan analisa yang telah dilakukan di dapatkan kesimpulan sebagai berikut: (1) Hasil perbandingan gerak lurus dengan metode PID dapat mengurangi nilai rata - rata penyimpangan yang didapat dengan metode open loop sebesar 2,08 cm pada sumbu X dan 0,22 cm pada sumbu Y menuju titik (0,160) cm. (2) Hasil Perbandingan gerak lurus dan belok kanan dengan metode PID dapat mengurangi nilai rata - rata penyimpangan yang didapat dengan metode open loop sebesar 1,41 cm pada sumbu X dan 0 cm pada sumbu Y menuju titik (0,160) cm, dan 0,94 cm pada sumbu X dan 4,16 cm pada sumbu Y menuju titik (120,160) cm. (3) Hasil Perbandingan gerak lurus dan belok kiri dengan metode PID dapat mengurangi nilai rata - rata penyimpangan yang didapat dengan metode open loop sebesar 0,44 cm pada sumbu X dan 3,81 cm pada

sumbu Y menuju titik (0,160) cm, dan 2,44 cm pada sumbu X dan 7,62 cm pada sumbu Y menuju titik (120,160) cm. (4) Hasil Perbandingan gerak robot kotak dengan metode PID dapat mengurangi nilai rata - rata penyimpangan yang didapat dengan metode open loop sebesar 0,72 cm pada sumbu X dan 0,62 cm pada sumbu Y menuju titik (0,160) cm, 0,6 cm pada sumbu X dan 8,50 cm pada sumbu Y menuju titik (120,160) cm, 18,6 cm pada sumbu X dan 2,12 cm pada sumbu Y menuju titik (120,0) cm dan 5,16 cm pada sumbu X dan 6,13 cm pada sumbu Y menuju titik (0,0) cm. (5) Pembuatan model kontrol gerak robot dengan PID dapat mengurangi nilai penyimpangan yang terjadi pada pergerakan robot dengan metode open loop. Namun masih terjadi penyimpangan pada gerakan robot terutama pada gerakan robot mengikuti jalur persegi, Hal ini mungkin disebabkan oleh kurang maksimalnya parameter PID yang digunakan saat proses tuning PID dilakukan. Selain itu, Komponen dan rangkaian robot yang digunakan masih bersifat low cost, sehingga performa yang dihasilkan dirasa masih kurang maksimal.

Untuk pengembangan lebih lanjut pada penelitian ini antara lain: (1) Memperbaiki dan mengevaluasi sistem kontrol PID dengan melakukan tuning dan kali brasi variabel nya agar mendapatkan hasil yang akurat. (2) Membuat standar maksimal penyimpangan agar sistem kontrol dapat

mereset kembali output sehingga bisa didapatkan pergerakan yang maksimal. (3) Dalam perancangan dan pembuatan alat ini masih terdapat kelemahan. Untuk memperbaiki kinerja alat dan pengembangan lebih lanjut di sarankan: Pembuatan chasis robot dan komponen yang lebih baik dan lebih kuat, agar pergerakan robot lebih stabil dan presisi.

## REFERENSI

- [1] D. Pratama, E. H. Binugroho, and F. Ardilla, "Movement Control Of Two Wheels Balancing Robot Using Cascaded PID Controller," in *2015 International Electronics Symposium (IES)*, Surabaya, Indonesia: IEEE, 2015, pp. 94–99. doi: 10.1109/ELECSYM.2015.7380821.
- [2] A. G. Wibisono, "Sistem pengendali robot bergerak lurus dengan kontrol Proporsional Integral Derivatif (PID) berbasis LabView," *TELEKONTRAN*, vol. 3, no. 2, pp. 13–24, 2015.
- [3] E. W. Suseno, A. Ma'arif, and R. D. Puriyanto, "Tuning parameter pengendali PID dengan metode algoritma genetik pada Motor DC," *TELKA - Telekomunikasi Elektronika Komputasi dan Kontrol*, vol. 8, no. 1, pp. 1–13, 2022, doi: 10.15575/telka.v8n1.1-13.
- [4] P. S. Gilapasa, S. Priyambodo, and S. Subandi, "Sistem pembelajaran kontrol PID (Proporsional Integral Derivatif) pada pengatur kecepatan motor DC," *Jurnal Elektrikal*, vol. 3, no. 1, pp. 72–77, 2016.
- [5] R. Arindya, "Penalaan Kendali PID untuk pengendali proses," *Jurnal Teknologi Elektro*, vol. 8, no. 2, p. 109, 2017.