

Implementation *Inverse Kinematics* for Precise Control of a Mecanum Wheels Robot

Senanjung Prayoga¹, Yossy Imelda Nathasia Simanjuntak²

¹²Politeknik Negeri Batam,
Jurusan Teknik Elektro, Program Studi Teknik Robotika,
Jl Ahmad Yani Batam Kota, Kota Batam, Kepulauan Riau 29461

Email: senanjung@polibatam.ac.id, yossyimeldaaa@gmail.com

Abstract— Perkembangan teknologi robotika telah membuka peluang baru dalam implementasi sistem navigasi yang lebih kompleks dan fleksibel. Salah satu inovasi yang muncul adalah penggunaan Robot mobile dengan roda mekanum pada robot, yang memungkinkan pergerakan omni-directional tanpa mengubah orientasi robot. Penelitian ini bertujuan untuk mengimplementasikan *Inverse Kinematics* (IK) untuk mencapai kontrol posisi yang presisi pada robot beroda mekanum. Metode IK diterapkan untuk menghitung nilai kecepatan linier (V_y) dan kecepatan lateral (V_x) dari roda mekanum, sehingga robot dapat mencapai posisi yang diinginkan dengan akurasi yang tinggi. Pengujian juga dilakukan dengan menerapkan kontroler PID Posisi pada robot untuk mengikuti lintasan yang telah ditentukan. Pengujian PID pada robot seluler menggunakan metode *trial and Error*. Penelitian ini bertujuan untuk meningkatkan akurasi pergerakan robot dalam lingkungan yang terstruktur, dengan mengikuti serangkaian *Waypoint* yang telah ditentukan. Hasil percobaan menunjukkan bahwa pengaturan PID dengan parameter $K_p = 0.25$, $K_i = 0.01$, $K_d = 0.1$ menghasilkan performa yang paling stabil dan halus, dengan rata-rata *Error* posisi 0.1% pada sumbu x dan y = 0.02%, serta *Error* orientasi (Theta) sebesar 0.003 %. percobaan lainnya memperlihatkan peningkatan *Error* dan overshoot ketika tuning parameter PID tidak optimal. Penggabungan antara IK dan Kendali PID pada robot seluler ini terbukti efektif dalam mengontrol pergerakan robot saat mengikuti lintasan.

Keyword: Invers Kinematika, Robot Mekanum, Kontrol PID

I. PENDAHULUAN

Robot adalah perangkat yang dirancang untuk membantu pekerjaan manusia, terdiri dari perangkat keras (sistem mekanik dan elektrik) dan perangkat lunak (program pengendali). Teknologi robot kini telah berkembang pesat dan digunakan di berbagai bidang kehidupan [1].

Perkembangan pasar dan teknologi robot seluler pada tahun 2018, nilai pasar robot mobile global mencapai USD 19 miliar dan diperkirakan hampir tiga kali lipat pada 2023 [2].

Dari semua jenis robot, mobile robot adalah yang paling umum dan sering digunakan sebagai subjek penelitian di

banyak pusat penelitian diseluruh dunia terutama karena kemampuannya bergerak dengan cepat di permukaan datar [3].

Selama ini banyak teknologi mobile robot menggunakan sistem pergerakan *differential drive*. *Differential drive* merupakan salah satu pergerakan robot yang memanfaatkan kecepatan pada roda kiri dan kanannya. Permasalahan utama dari sistem pergerakan *differential drive* adalah terbatasnya pergerakan robot karena hanya bisa bergerak maju, mundur dan belok tetapi robot tidak mampu bergerak kesegala arah (robot *non-holonomic*). Oleh karena itu dikembangkan sebuah robot yang dapat bermanuver kesegala arah pada bidang kartesian X - Y tanpa perlu memutar badan dari robot tersebut. Salah satu jenis robot yang memiliki kemampuan tersebut adalah robot dengan roda mekanum [4].

Roda mekanum, ditemukan oleh Bengt Ilon, adalah roda holonomic yang memungkinkan robot bergerak ke segala arah tanpa mengubah arah hadapnya. Dengan roller yang dipasang pada sudut 45°, roda ini memungkinkan gerakan maju, mundur, samping, atau diagonal. Meskipun sangat fleksibel untuk manuver di ruang terbatas, roda mekanum bisa mengalami slip pada permukaan tidak rata. Pengaturan gerakan memerlukan pemahaman invers kinematika untuk menghitung kecepatan atau sudut rotasi roda berdasarkan input gerakan yang diinginkan [5].

Dalam penelitian Fahmizal, kontrol PID digunakan untuk stabilitas robot karena kemudahan penggunaannya pada mikrokontroler. Namun, penentuan parameter K_p , K_i , dan K_d memerlukan *trial and error* yang memakan waktu. Metode Ziegler-Nichols mempercepat pencarian parameter dengan memulai *trial and error* hanya pada K_p . Proses tuning melibatkan peningkatan K_p hingga didapatkan grafik respon PID yang berkesinambungan, yang kemudian digunakan untuk menghitung parameter K_p , K_i , dan K_d sesuai metode Ziegler-Nichols [6].

Namun, dalam penelitian Tugas Akhir ini penulis menggunakan metode *Trial and Error* dalam aplikasi tuning PID. Keputusan ini diambil karena beberapa pertimbangan, seperti kebutuhan untuk menyesuaikan parameter yang spesifik. Dengan menggunakan metode *Trial and Error*, proses tuning melibatkan eksperimen langsung dengan berbagai kombinasi

nilai parameter K_p , K_i , dan K_d , serta pemantauan respon sistem untuk setiap kombinasi tersebut.

Dalam penelitian ini akan dirancang sebuah mobile robot roda mecanum yang dapat bergerak menuju titik koordinat tertentu. Selain itu, digunakan kontrol PID pada tiap motor mobile robot untuk mempertahankan *setpoint* kecepatan pada setiap roda robot. Dengan menggunakan rumus inverse kinematik untuk robot yang menggunakan empat buah roda mecanum diharapkan robot dapat memiliki sebuah sistem gerak yang baik, efisien dan tepat saat bermanuver. Kinematika robot sangat penting untuk mendefinisikan arah dan kecepatan robot [7].

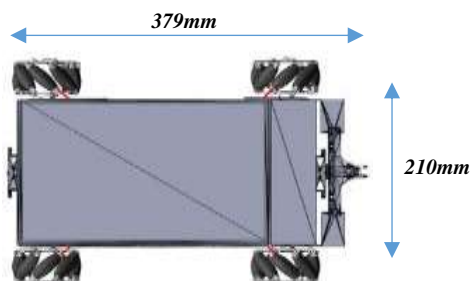
Susunan pada jurnal ini disusun menjadi empat bagian. Pada bab 2 metode penelitian. Di bab 3 menjelaskan hasil data dari implementasi invers kinematic pada setiap roda robot di bab 4 menjelaskan kesimpulan dari penelitian ini.

II. METODE PENELITIAN

Penelitian ini menerapkan metode *Inverse Kinematics* menggunakan platform Arduino Mega 2560 dengan bantuan software Robot *Operating System (ROS)* untuk memperoleh informasi mengenai arah dan kecepatan roda yang saling terhubung pada sebuah sistem robot. Metode *Inverse Kinematics* digunakan untuk menghitung nilai kecepatan masing-masing roda yang dibutuhkan agar robot dapat mencapai posisi dan orientasi yang diinginkan. Arduino sebagai otak utama sistem, bertanggung jawab atas pengolahan data dan pengiriman sinyal kendali ke motor-motor secara akurat.

A. Desain Mekanik Mecanum Wheels Robot

Pada penelitian ini, robot dilengkapi dengan empat buah roda mecanum yang memungkinkan pergerakan omnidirectional. Chassis robot ini dibuat menggunakan bahan plat aluminium, dengan dimensi ukuran robot panjang 379 mm, lebar 210 mm, tinggi 90 mm dan bobot keseluruhan robot mencapai 3.2 kg. Diameter roda mecanum adalah 97 mm.



Gambar 1. Desain Mekanik Robot Trainer Kit.

Desain robot (Gambar 1) menggunakan perangkat lunak 3D seperti SolidWorks sangat penting dalam pembuatan mekanik robot dengan roda mecanum. Jarak sumbu antar roda penting untuk stabilitas, manuver, dan distribusi pusat massa robot. Jarak yang tidak tepat dapat mengganggu kemampuan manuver dan akurasi perhitungan kinematika. Oleh karena itu, desain mekanik yang baik diperlukan agar perhitungan invers kinematika dan sistem kontrol PID dapat bekerja dengan akurat, memastikan gerakan robot sesuai yang diinginkan [8].

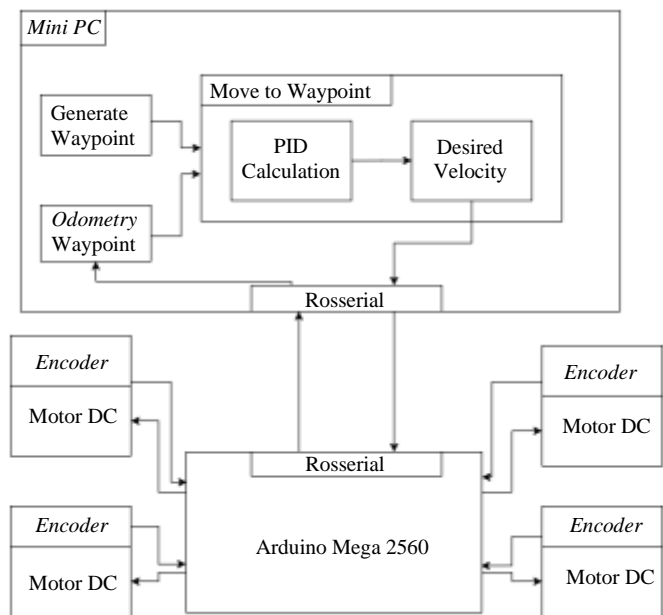
Selain itu, saat memasang roda mecanum, penting untuk memastikan bahwa semua roda mengarah ke tengah sasis robot

(Gambar 2). Ini bertujuan untuk menentukan pusat massa robot yang harus berada di tengah sasis agar performa dan stabilitas robot tetap optimal, terutama ketika roda mecanum bergerak dalam berbagai arah. Penentuan pusat massa yang tepat akan sangat berpengaruh terhadap kestabilan robot dan kemampuannya untuk bermanuver dengan baik.



Gambar 2. Instalasi Roda Mecanum.

B. Diagram Blok Sistem



Gambar 3. Blok Diagram Sistem.

Pada Gambar 3 blok diagram sistem pengendalian robot menggunakan Arduino mega 2560 dan *Mini PC*, yang terhubung dengan motor DC melalui *encoder*. *Mini PC* digunakan untuk menentukan *Waypoint* dan mengendalikan pergerakan robot dengan menghitung kontrol PID untuk menentukan kecepatan. Data *odometry* dikirim ke *Mini PC* untuk memantau posisi robot secara real-time. *Mini PC* dan Arduino berkomunikasi melalui *rosserial* untuk transfer data seperti perintah *Waypoint* dan *odometry*. *Odometry* memperkirakan posisi relatif robot berdasarkan pulsa dari sensor *rotary encoder* yang dikonversi menjadi milimeter [9].

Arduino berfungsi sebagai pusat kontrol yang menerima perintah dari *Mini PC* melalui protokol komunikasi *Rosserial*, yang menghubungkan perangkat keras seperti Arduino dengan perangkat lunak seperti *Mini PC*. Protokol ini memungkinkan pengiriman data yang efisien, sehingga Arduino dapat mengontrol motor dan perangkat lain serta mengirim umpan

balik ke *Mini PC*. Komunikasi ini penting untuk memastikan sistem robot bekerja secara sinkron dan menjalankan tugas dengan benar [10].

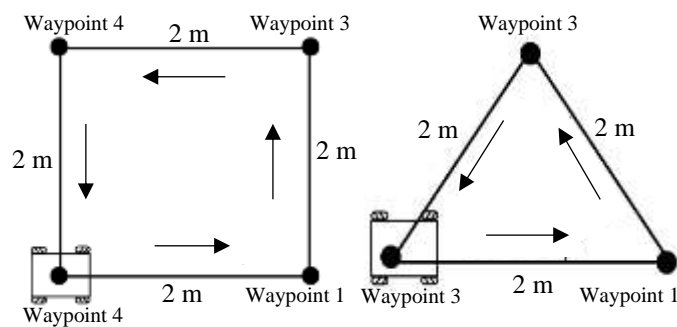
Arduino terhubung ke *encoder* pada empat motor DC, yang mengukur putaran dan posisi motor untuk memperbarui kecepatan dan posisi robot. Data PID dihitung di *Mini PC*, lalu kecepatan yang diinginkan dikirim ke Arduino untuk mengontrol motor berdasarkan umpan balik dari *encoder*. Sistem ini menggunakan kontrol loop tertutup, di mana PID terus-menerus mengoreksi pergerakan robot berdasarkan umpan balik odometri untuk memastikan pergerakan sesuai jalur yang diinginkan [11].

Penelitian ini menggunakan metode kuantitatif untuk mengukur akurasi dan performa sistem kontrol dengan mengumpulkan data posisi dan kecepatan robot dari setiap percobaan. Metode ini melibatkan pengukuran numerik dan analisis statistik untuk menguji teori melalui proses yang meliputi teori, hipotesis, desain penelitian, pengumpulan dan analisis data, serta kesimpulan. Fokus penelitian adalah pada data yang dapat dihitung dan dianalisis empiris untuk menguji keabsahan teori [12].

Data ini kemudian dianalisis untuk mengevaluasi keberhasilan sistem dalam mencapai *Waypoint* dengan tingkat akurasi yang diharapkan.

1. *Waypoint* ditentukan sebagai target posisi.
2. *Inverse Kinematics* menghitung kecepatan angular yang diperlukan untuk mencapai *Waypoint*.
3. Kontrol PID mengoreksi pergerakan berdasarkan *Error* antara posisi yang diinginkan dan posisi aktual.
4. Odometri memberikan *Feedback* posisi dan orientasi aktual robot selama pergerakan.
5. PID terus mengontrol pergerakan hingga robot mencapai *Waypoint* dengan akurasi tinggi.

C. Perancangan Pergerakan Robot Mecanum



Gambar 4. Pergerakan Robot Mecanum

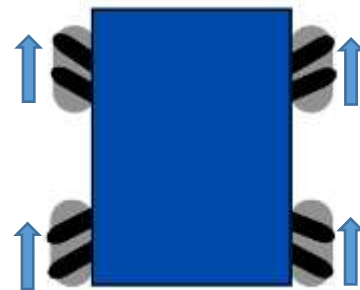
Penelitian ini melibatkan dua percobaan lintasan untuk robot mecanum. Lintasan 1 adalah persegi 2 m x 2 m dan Lintasan 2 adalah segitiga sama sisi dengan sisi 2 m. Pada lintasan persegi, robot bergerak dari Start/Waypoint 4 ke Waypoint 1, melewati Waypoint 2 dan Waypoint 3, dan kembali ke Waypoint 4. Pada lintasan segitiga, robot bergerak dari Start/Waypoint 3 ke Waypoint 1, melewati Waypoint 2, dan berhenti di Waypoint 3. Dalam pengujian Invers Kinematik (IK), input berupa kecepatan linier (V_x , V_y) dan kecepatan

sudut (ω) digunakan untuk menghitung kecepatan rotasi roda (ω_1 - ω_4), memungkinkan robot bergerak sesuai input tersebut [13].

Proses ini penting karena untuk menggerakkan robot dengan benar, sistem harus menghitung bagaimana setiap roda berputar secara independen agar robot bergerak ke arah dan kecepatan yang diinginkan.

D. Inverse Kinematics

Robot mecanum ini akan dimulai dari penyelesaian rumus *Inverse Kinematics* yang digunakan sebagai dasar perhitungan arah gerak robot (Gambar 5) dari hasil perhitungan *Inverse Kinematics* ini, arah gerak robot dapat bergerak ke segala arah dari mulai arah ke depan, ke samping, mundur bahkan perputaran.



Gambar 5. Arah putar roda penentu arah.

Persamaan *Inverse Kinematics* pada robot dengan roda mecanum merujuk pada persamaan matematis yang dapat menghitung kecepatan roda (θ_{FL} , θ_{FR} , θ_{BL} , θ_{BR}). Fungsi persamaan ini adalah untuk mencapai pergerakan robot dengan kecepatan linier (V_x dan V_y) dan kecepatan angular (ω) sesuai dengan tujuan yang diinginkan [1].

Berikut persamaan *Inverse Kinematics Mecanum Wheels* Robot yang digunakan pada penelitian ini. Persamaan *Inverse Kinematics* dari konfigurasi robot *Trainer Kit* beroda mecanum disajikan pada persamaan (1).

$$\begin{bmatrix} \theta_{FL} \\ \theta_{FR} \\ \theta_{BL} \\ \theta_{BR} \end{bmatrix} = \frac{1}{2\pi r} \begin{bmatrix} 1 & -1 & -(d+e)/2 \\ 1 & 1 & (d+e)/2 \\ 1 & 1 & -(d+e)/2 \\ 1 & -1 & (d+e)/2 \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} \quad (1)$$

Keterangan:

- θ_{FL} : Kecepatan roda depan-kiri (rps)
- θ_{FR} : Kecepatan roda depan-kanan (rps)
- θ_{BL} : Kecepatan roda belakang-kiri (rps)
- θ_{BR} : Kecepatan roda belakang-kanan (rps)
- r : Radius roda (mm)
- d : Jarak roda kiri ke roda kanan (mm)
- e : Jarak roda depan ke roda belakang (mm)
- V_x : Kecepatan robot dalam arah maju (m/s)
- V_y : Kecepatan robot dalam arah samping (m/s)
- ω : Kecepatan rotasional robot (rad/s)

TABEL 1
DATA ROBOT DAN PARAMETER RODA ROBOT

Parameter	Panjang (mm)
Diameter Roda	97
Jarak d	270
Jarak e	300

Dalam penyelesaian Invers Kinematik (IK) untuk roda mecanum, nilai V_x (kecepatan translasi sumbu X), V_y (kecepatan translasi sumbu Y), dan ω (kecepatan rotasi) diterapkan. Ketika robot diberikan target kecepatan kombinasi translasi dan rotasi, persamaan kinematik invers digunakan untuk menentukan kecepatan masing-masing roda, sehingga robot bergerak sesuai target tersebut. Tabel 1 menyajikan data dan parameter robot yang digunakan dalam persamaan untuk mencari kecepatan roda.

E. Kontrol PID

Kontrol PID (Proportional-Integral-Derivative) adalah metode pengendalian yang menggabungkan tiga komponen utama—Proportional, Integral, dan Derivative—untuk memastikan keluaran sistem sesuai dengan masukan yang diinginkan. Digunakan dalam berbagai bidang seperti industri, elektronika, dan robotika, PID membantu mengatasi gangguan, mengurangi kesalahan, dan meningkatkan performa sistem melalui rumus yang mengoptimalkan keseimbangan pengendalian [15]:

1. *Proportional* (P): Kontrol proportional memperkuat sinyal kesalahan, yaitu selisih antara posisi robot saat ini dan posisi yang diinginkan. Pada roda mecanum, kontrol ini membantu mengatur posisi robot dalam gerak translasi (V_x , V_y) dan rotasi (ω) sesuai target. Kontrol proportional merespons kesalahan posisi dengan cepat, menyesuaikan pergerakan robot untuk mengurangi perbedaan antara posisi aktual dan yang diinginkan.
2. *Integral* (I): Kontrol integral mengintegrasikan kesalahan dari waktu ke waktu untuk mengatasi kesalahan steady-state, memastikan gerakan lebih akurat. Ini juga memperbaiki respons terhadap gangguan dan faktor eksternal. Namun, jika gain integral terlalu besar, bisa menyebabkan osilasi, sehingga perlu keseimbangan yang tepat dengan kontrol proportional dan derivatif.
3. *Derivative* (D): berfungsi untuk mengurangi overshoot, mempercepat respons terhadap perubahan mendadak, dan menekan osilasi. Ini dilakukan dengan menghitung laju perubahan kesalahan. Dengan demikian, kontrol derivatif membantu menjaga stabilitas dan akurasi pergerakan robot. Namun, sensitivitas yang terlalu tinggi dapat memperkuat noise, sehingga diperlukan pengaturan yang tepat dengan kontrol proportional dan integral.

Metode PID ini bertujuan untuk mengatur dan mengontrol posisi robot agar bergerak secara presisi sehingga mencapai nilai target (*Set point*) dengan mengurangi *Error* posisi ke tingkat yang minimal [16]. Penelitian ini menggunakan metode *Trial and Error* sebagai pendekatan untuk menentukan

parameter kontroler PID. Teknik *Trial and Error* merupakan metode coba-coba yang sering digunakan dalam penentuan parameter PID, di mana peneliti mencoba berbagai kombinasi parameter PID hingga mendapatkan respons sistem yang sesuai dengan kebutuhan dan mengetahui perbandingan parameter PID yang didapatkan [17].

Pada penelitian ini, peneliti melakukan percobaan dengan beberapa set parameter PID untuk menguji performa sistem pada dua jenis lintasan, yaitu lintasan 1 dan lintasan 2. Setiap kombinasi parameter diuji untuk melihat bagaimana pengaruhnya terhadap pergerakan dan kestabilan robot. Hasil dari percobaan ini digunakan untuk menentukan kombinasi parameter yang memberikan respons terbaik pada masing-masing lintasan.

Berikut adalah tabel yang menunjukkan parameter percobaan *Trial and Error* yang digunakan pada lintasan 1 dan lintasan 2:

TABEL 2
PARAMETER PADA LINTASAN 1 DAN LINTASAN 2

Parameter Pengujian pada Robot Mecanum				
Pengujian	Target posisi sudut (m)	Kp	Ki	Kd
1	2	0.3	0.01	0.1
2	2	0.25	0.01	0.1
3	2	0.35	0	0

Tabel 2 menunjukkan parameter pengujian PID pada lintasan 1 dan 2 dengan target posisi sudut sejauh 2 meter. Pengujian dilakukan dengan tiga set parameter PID berbeda untuk mengevaluasi pengaruhnya terhadap kemampuan robot mecanum mencapai dan mempertahankan posisi target. Persamaan PID yang digunakan pada robot mecanum disajikan dalam penelitian ini (2).

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

Keterangan:

1. $U(t)$ = Output kendali
2. $e(t)$ = Kesalahan atau *Error* posisi
3. \int_0^t = Akumulasi nilai fungsi kesalahan dari sebelumnya.
4. $d\tau$ = Akumulasi kesalahan posisi pada waktu τ .
5. $de(t)$ = Determinan t
6. $e(\tau)$ = Kesalahan posisi aktual dari posisi set point.
7. $\frac{de(t)}{dt}$ = Turunan kesalahan terhadap waktu.

Dalam desain kontroler PID, tuning parameter Proportional (P), Integral (I), dan Derivative (D) sangat penting untuk memenuhi kriteria performansi sistem loop tertutup. Tuning yang optimal bertujuan mencapai rise time dan settling time yang cepat, bebas dari steady-state error, dan overshoot minimal. Dengan tuning yang tepat, kontroler PID meningkatkan kecepatan respons, akurasi, dan stabilitas sistem. [18].

III. HASIL DAN PEMBAHASAN

Pengujian *Inverse Kinematics* robot dilakukan dengan memberikan *input* berupa (V_x , V_y , ω) kemudian nilai-nilai tersebut dimasukkan ke dalam persamaan kinematik robot untuk menghasilkan ω_1 , ω_2 , ω_3 dan ω_4 .

A. Hasil Pengujian *Inverse Kinematics* pada Roda Mekanum

Pengujian pengendalian pergerakan robot bertujuan menentukan kecepatan masing-masing motor dengan menggunakan metode *Inverse Kinematics* (IK). Output IK berupa nilai rps yang digunakan untuk mengatur kecepatan motor. Tabel 3 menunjukkan hasil pengujian kecepatan roda robot Trainer Kit berdasarkan formula IK. Pengujian dilakukan tanpa hambatan, dengan nilai positif pada output mengindikasikan perputaran roda searah jarum jam, dan nilai negatif menunjukkan perputaran berlawanan arah jarum jam. Sebagai contoh, dengan input $V_x=0.5$ m/s, $V_y=0$ m/s, dan $\omega=0$ rad/s, kecepatan roda adalah $\omega_1=\omega_2=\omega_3=\omega_4=1.64$ rad/s.

TABEL 3
INPUT PERCOBAAN

No.	V_X (m/s)	V_Y (m/s)	θ (rad/s)
1	0.5	0	0
2	-0.5	0	0
3	0	0.5	0
4	0	-0.5	0
5	0.5	0.5	0
6	0.5	-0.5	0
7	0	0	1.5
8	0	0	-1.5

TABEL 4
RESULT PERCOBAAN ω_1 , ω_2 , ω_3 , ω_4 .

Output Percobaan					
No.	ω_1 (rad/s)	ω_2 (rad/s)	ω_3 (rad/s)	ω_4 (rad/s)	Pergerakan Robot
1	1.64	1.64	1.64	1.64	Maju
2	-1.64	-1.64	-1.64	-1.64	Mundur
3	-1.64	1.64	-1.64	1.64	Kanan
4	1.64	-1.64	1.64	-1.64	Kiri
5	0	3.28	0	3.28	Serong Kanan
6	3.28	0	3.28	0	Serong Kiri
7	1.4	-1.4	-1.4	1.4	Putar Kanan
8	-1.4	1.4	1.4	-1.4	Putar Kiri

Dapat disimpulkan arah pergerakan roda pada Tabel 4 menunjukkan, jika *input* $V_x=0.5$ m/s, $V_y=0$ m/s, dan $\omega=0$ rad/s menghasilkan *output* Kecepatan roda $\omega_1=1.64$ rad/s, kecepatan roda $\omega_2=1.64$ rad/s, kecepatan roda $\omega_3=1.64$ rad/s, Kecepatan roda $\omega_4=1.64$ rad/s, Maka arah pergerakan robot maju ke depan.

B. Hasil Pengujian Lintasan 1

1) Hasil Pengujian Lintasan 1 tanpa Menggunakan PID

Pengujian posisi pergerakan robot yang dilengkapi dengan roda mekanum dilakukan dengan tujuan untuk mengevaluasi sejauh mana *Error* pada posisi robot terhadap target posisi. Penentuan arah pergerakan robot ini dihitung menggunakan persamaan *Inverse Kinematics* dengan bantuan *Forward Kinematics* yang di implementasikan.

Dalam pengujian ini, mikrokontroler diberi *input* berupa x , y dan θ lalu robot bergerak melintasi lintasan 1. Robot bergerak

secara otonom mengikuti *Waypoint* dimulai dari *Waypoint* 1 sampai ke *Waypoint* 4 selanjutnya robot akan berhenti ketika mencapai tujuannya dapat dilihat dari tabel 5.

Pada Tabel 5, terlihat bahwa dari *Start* ke *Waypoint* 1, robot diberi *input* sebesar $x=2$ m, $y=0$ m, dan $\theta=0$ rad/s, menghasilkan posisi pada Tabel 5. *Error* dihitung sebagai perbedaan antara posisi yang diinginkan (*Waypoint*) dan posisi aktual dapat dilihat dari persamaan 3.

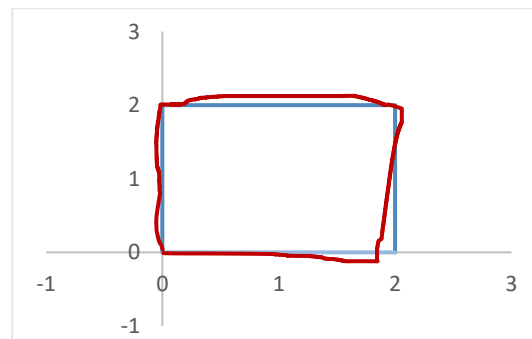
$$\text{Error} = \text{Nilai Target} - \text{Nilai Posisi} \times 100\% \quad (3)$$

TABEL 5
ERROR PERGERAKAN ROBOT

		Waypoint			
		1	2	3	4
Target	x (m)	2	2	0	0
	y (m)	0	2	2	0
	Theta (rad/s)	0	0	0	0
Posisi	x (m)	1.923	2.147	0.048	-0.007
	y (m)	-0.107	1.950	2.005	0.048
	Theta (rad/s)	-0.030	0.048	-0.024	0.060
Error	x (m)	0.077	-0.147	-0.048	0.007
	y (m)	0.107	0.05	-0.005	-0.048
	Theta (rad/s)	0.030	-0.048	0.024	-0.060

Data *Error* dicatat selama pergerakan robot dari satu *Waypoint* ke *Waypoint* berikutnya. Pengujian ini dilakukan dengan mengambil posisi robot setiap waktu dari *odometry* menggunakan *encoder* pada masing-masing roda. Rata-rata *Error* pada sumbu X yaitu 0.02775 % , Y yaitu 0.026 % dan theta yaitu 0.0135 %.

Proses perhitungan *Feedback system* digunakan untuk menentukan posisi aktual robot. Robot berhenti begitu mencapai tujuannya, dengan perhitungan *Feedback system* menunjukkan adanya perubahan koordinat pada saat pemberhentian robot.



Gambar 6. Target posisi dan Posisi Aktual.

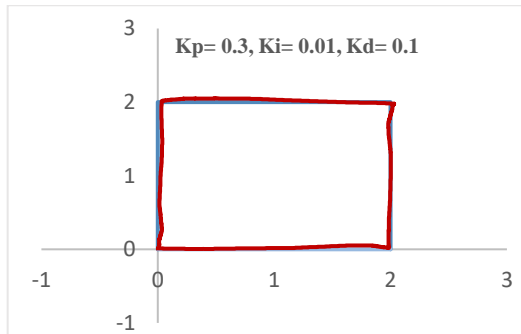
Untuk mengilustrasikan *Waypoint* dari jalannya robot pada lintasan 1, kita bisa melihat Gambar 6 yang tersedia dibawah ini. Gambar tersebut menjelaskan tentang jalannya robot dari titik koordinat ($x=0$, $y=0$) *Waypoint* 4/*Start* lalu melintasi koordinat ($x=2$, $y=0$) *Waypoint* 1 setelah itu menuju titik koordinat ($x=2$, $y=2$) *Waypoint* 2, lalu menuju ke koordinat ($x=0$, $y=2$) *Waypoint* 3 dan kembali ke posisi awal robot dengan koordinat ($x=0$, $y=0$) *Waypoint* 4, lalu robot berhenti sesuai dengan titik koordinat yang telah ditentukan. Terlihat dalam Gambar 1 adalah perbandingan antara lintasan yang

diharapkan garis berwarna biru dan Lintasan aktual yang dilalui robot berwarna merah.

2) Hasil Percobaan PID pada Lintasan 1

Pada tahap ini, penelitian menguji sistem kontrol PID pada robot seluler dengan roda mekanum saat mengikuti Lintasan 1. Percobaan dilakukan untuk menganalisis kemampuan robot mencapai setiap *Waypoint* dengan akurasi tinggi menggunakan kontrol PID. Percobaan dilakukan pada lintasan berbentuk persegi dengan 4 *Waypoint* yang berjarak sama, dan setiap *Waypoint* menjadi target pergerakan robot secara bertahap.

- *Percobaan Lintasan 1 dengan Parameter $K_p=0.3$, $K_i=0.01$ dan $K_d=0.1$*



Gambar 7. Hasil Percobaan 1.

Gambar 7 menunjukkan bahwa robot berhasil mengikuti lintasan yang cukup mendekati target *Waypoint*. Namun, terdapat sedikit *overshoot* di setiap sudut pola, di mana robot cenderung bergerak melewati titik belok sebelum kembali ke jalur yang diinginkan. Meskipun stabil, gerakan robot masih menunjukkan sedikit osilasi, terutama saat mendekati dan mengubah arah di sudut-sudut pola. Robot masih memerlukan beberapa koreksi untuk mencapai posisi akhir yang akurat. Secara keseluruhan, lintasan robot cukup stabil dan tidak terlalu banyak mengalami *overshoot* atau *undershoot* besar.

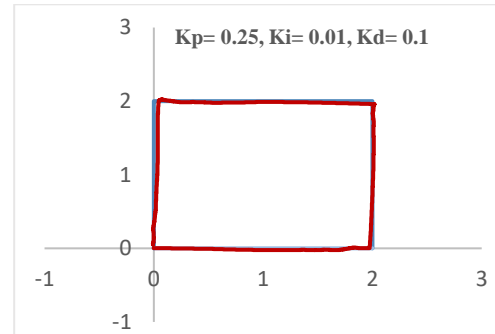
TABEL 6
PENGUJIAN PERGERAKAN ROBOT DENGAN LINTASAN 1

		Waypoint			
		1	2	3	4
Target	x (m)	2	2	0	0
	y (m)	0	2	2	0
	Theta (rad/s)	0	0	0	0
Posisi	x (m)	1.950	1.996	0.049	0.021
	y (m)	0.007	1.950	1.995	0.044
	Theta (rad/s)	0.018	0.003	-0.006	0.008
Error	x (m)	0.05	0.004	-0.049	-0.021
	y (m)	-0.007	0.05	0.005	-0.044
	Theta (rad/s)	-0.018	-0.003	0.006	-0.008

Error dihitung sebagai perbedaan antara posisi yang diinginkan dan posisi aktual, sesuai persamaan 3. Dengan parameter $K_p=0.3$, $K_i=0.01$, dan $K_d=0.1$, pengujian lintasan robot menunjukkan hasil pada Gambar 7 dan Tabel 6. Robot cukup dekat dengan target, dengan rata-rata error $x=0.004\%$, $y=0.001\%$, dan $\text{Theta}=0.00575\%$. Meskipun akurasi posisi dan

orientasi baik, penyesuaian lebih lanjut pada parameter mungkin diperlukan untuk meminimalkan kesalahan pada sumbu x dan orientasi.

- *Percobaan Lintasan 1 dengan Parameter $K_p=0.25$, $K_i=0.01$ dan $K_d=0.1$*



Gambar 8. Hasil Percobaan 2.

Gambar 8 menunjukkan bahwa robot mampu mengikuti lintasan dengan cukup stabil dan minim osilasi. Kurva lintasan terlihat lebih halus dibandingkan dengan percobaan sebelumnya, yang menunjukkan kontrol yang lebih baik dalam menjaga posisi robot di setiap titik *Waypoint*. Stabilitas yang ditingkatkan ini menunjukkan bahwa pengaturan parameter PID ini lebih sesuai untuk gerakan yang presisi dan terkendali.

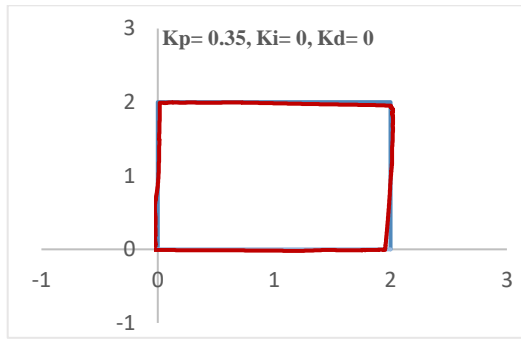
TABEL 7
PENGUJIAN PERGERAKAN ROBOT DENGAN LINTASAN 1

		Waypoint			
		1	2	3	4
Target	x (m)	2	2	0	0
	y (m)	0	2	2	0
	Theta (rad/s)	0	0	0	0
Posisi	x (m)	1.9501	1.9956	0.0488	-0.0076
	y (m)	0.0004	1.9501	2.0095	0.0492
	Theta (rad/s)	-0.0097	0.0084	-0.0036	-0.0099
Error	x (m)	0.0499	0.0044	-0.0488	0.0076
	y (m)	-0.0004	0.0499	-0.0095	-0.0492
	Theta (rad/s)	0.0097	-0.0084	0.0036	0.0099

Dari tabel 7, didapati pada waypoint 1 posisi aktual $x=1.9501$ m robot hampir mendekati target (2 m) dengan error hanya 0.0499 m dan posisi aktual $y=0.0004$ m sangat dekat dengan target (2 m) dengan error yang sangat kecil -0.0004 m. ini menunjukkan bahwa robot berada cukup dekat dengan target, tetapi masih terdapat kekurangan dalam posisi target pada sumbu x dan robot telah hampir mencapai target pada sumbu y.

Rata-rata error untuk posisi $x=0.003275\%$, $y=0.0023\%$ dan $\text{Theta}=0.0037\%$. Parameter PID yang digunakan dalam percobaan ini telah memberikan hasil yang memuaskan dengan kesalahan rata-rata yang rendah. Parameter $K_p=0.25$, $K_i=0.01$ dan $K_d=0.1$ yang digunakan sebelumnya terbukti efektif dalam mengendalikan posisi robot, dengan performa yang baik dalam mencapai dan mempertahankan posisi target.

- Percobaan Lintasan 1 dengan Parameter $K_p=0.35$, $K_i=0$ dan $K_d=0$



Gambar 9. Hasil Percobaan 3.

Dari hasil percobaan dengan parameter $K_p=0.35$, $K_i=0$ dan $K_d=0$, Dilakukan pengujian lintasan robot untuk melihat akurasi posisi dan orientasi. Hasil percobaan ini dapat dilihat pada Gambar 9 dengan data uji posisi lintasan yang disajikan pada Tabel 8.

TABEL 8
PENGUJIAN PERGERAKAN ROBOT DENGAN LINTASAN 1

		Waypoint			
		1	2	3	4
Target	x (m)	2	2	0	0
	y (m)	0	2	2	0
	Theta (rad/s)	0	0	0	0
Posisi	x (m)	1.950	2.001	0.049	0.007
	y (m)	-0.001	1.950	1.992	0.049
	Theta (rad/s)	-0.009	0.013	0.007	-0.016
Error	x (m)	0.05	-0.001	-0.049	-0.007
	y (m)	0.001	0.05	0.008	-0.049
	Theta (rad/s)	0.009	-0.013	-0.007	0.016

Dari Tabel 8, pada Waypoint 1, posisi aktual robot hampir mencapai target dengan error $x=0.05$ m dan $y=-0.009$ m. Meskipun ada sedikit kekurangan pada sumbu x dan penyimpangan kecil pada sudut theta, robot cukup dekat dengan target. Pengaturan nilai $K_p=0.35$ tanpa K_i dan K_d memberikan kinerja stabil, dengan sedikit overshoot pada orientasi. Error rata-rata adalah $x=0.00175\%$, $y=0.0025\%$, dan orientasi 0.00125% , menunjukkan robot dapat mempertahankan arah dengan baik.

TABEL 9
PERBANDINGAN PARAMETER PID PADA LINTASAN 1

Trial	Error (x)	Error (y)	Error (Theta)	Kesimpulan
1	-0.004 %	-0.0092 %	-0.0017%	Robot berada dibawah target, kendali orientasi perlu ditingkatkan.
2	0.0033 %	-0.0023 %	0.0025 %	Robot menunjukkan perbaikan dalam kontrol arah robot
3	-0.0058%	0.01%	0.00125%	Robot sedikit bergeser dari arah yang diinginkan.

Pada ketiga percobaan, *Error* posisi rata-rata untuk sumbu $x = 0.00215\%$ dan $y = 0.0005\%$ dan $\theta = 0.0007\%$ dapat dilihat dari Tabel 9. Hal ini menunjukkan bahwa pengendalian posisi robot secara keseluruhan baik pada ketiga pengujian.

3) Perbandingan Pengujian Lintasan 1 Menggunakan PID dan Tanpa Menggunakan PID

TABEL 10
PERBANDINGAN PENGGUNAAN PID DAN TANPA PID

Error	Tanpa PID	PID
X (%)	-0.028 %	-0.0022 %
Y (%)	0.026 %	-0.0005 %
Theta	-0.014 %	0.0007 %

Tabel 10 menunjukkan perbandingan error posisi dan orientasi (theta) pada robot dengan dan tanpa penggunaan kontrol PID. Tabel ini mengukur tiga parameter error, yaitu pada sumbu X, sumbu Y, dan orientasi (Theta). Pada sumbu X, tanpa PID error sebesar 0,028%, sedangkan dengan PID, error berkurang drastis menjadi 0,0022%. Di sumbu Y, error tanpa PID adalah 0,026%, namun dengan PID menurun menjadi 0,0005%, menunjukkan peningkatan akurasi. Untuk orientasi (Theta), error tanpa PID sebesar -0,014%, sementara dengan PID berkurang menjadi 0,0007%. Hasil ini menunjukkan bahwa penggunaan PID secara signifikan mengurangi error pada sumbu X, Y, dan orientasi, menghasilkan pergerakan robot yang lebih akurat dan stabil.

C. Hasil Pengujian Lintasan 2

1) Hasil Percobaan Lintasan 2 tanpa Menggunakan PID

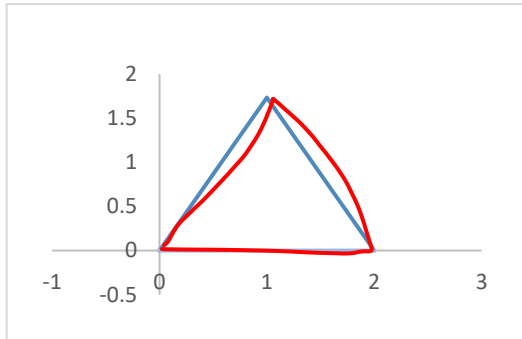
Pengujian ini menggunakan lintasan dari Gambar 8, dengan mikrokontroler menerima input x , y , dan θ . Robot bergerak otonom mengikuti Waypoint dari Waypoint 1 ke Waypoint 3 dan berhenti saat mencapai tujuan, seperti yang terlihat di Tabel 11. Dari Start ke Waypoint 1, input adalah $x=2$ m, $y=0$ m, dan $\theta=0$ rad/s. Error dihitung sebagai perbedaan antara posisi yang diinginkan dan posisi aktual, seperti di persamaan 2. Data Error dicatat selama pergerakan antar Waypoint, berdasarkan posisi robot yang diambil dari odometry menggunakan encoder roda, seperti tertera di Tabel 10.

TABEL 11
PERBANDINGAN PARAMETER TANPA PID PADA LINTASAN 1

		Waypoint		
		1	2	3
Target	x (m)	2	2	0
	y (m)	0	2	0
	Theta (rad/s)	0	0	0
Posisi	x (m)	1.9531	1.0375	0.0222
	y (m)	-0.0158	1.6993	0.0447
	Theta (rad/s)	-0.0294	-0.0008	0.0091
Error	x (m)	0.0469	0.9625	-0.0222
	y (m)	0.0158	0.3007	-0.0447
	Theta (rad/s)	0.0294	0.0008	-0.0091

Pengujian ini mengevaluasi kemampuan robot mengikuti pergerakan sesuai rencana dalam Gambar 10. Tabel 11 menunjukkan hasil pengukuran Error di setiap Waypoint, dengan robot mulai dari $(x=2, y=0)$ ke Waypoint 1, melintasi $(x=2, y=2)$ ke Waypoint 2, dan kembali ke $(x=0, y=0)$ Waypoint

3. Rata-rata Error adalah 0.33% pada sumbu x, 0.09% pada sumbu y, dan 0.007% pada theta, berdasarkan perhitungan Feedback system.



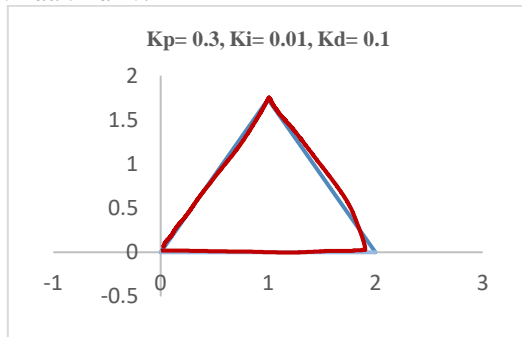
Gambar 10. Target posisi dan Posisi aktual robot.

Gambar 10 menunjukkan perbandingan antara target posisi (ditunjukkan dengan garis biru) dan posisi aktual robot (ditunjukkan dengan garis merah). Target posisi mewakili lintasan yang direncanakan atau diharapkan untuk diikuti oleh robot, sedangkan posisi aktual menunjukkan lintasan yang sebenarnya dilalui oleh robot selama pengujian. Pada gambar ini, perbedaan antara lintasan yang diharapkan dan lintasan aktual dapat terlihat, di mana robot tidak sepenuhnya mengikuti jalur yang diinginkan. Perbedaan tersebut menunjukkan adanya deviasi atau error dalam pergerakan robot dibandingkan dengan target posisi yang telah ditentukan.

2) Hasil Percobaan Lintasan 2 Menggunakan PID

Pada tahap ini, penelitian menguji sistem kontrol PID pada robot seluler dengan roda mekanum saat mengikuti Lintasan 2. Percobaan dilakukan untuk menganalisis kemampuan robot mencapai setiap Waypoint dengan akurasi tinggi menggunakan kontrol PID. Percobaan dilakukan pada lintasan berbentuk segitiga dengan 3 Waypoint yang berjarak sama, dan setiap Waypoint menjadi target pergerakan robot secara bertahap.

- Percobaan PID Lintasan dengan Parameter $K_p=0.3$, $K_i=0.01$ dan $K_d=0.1$



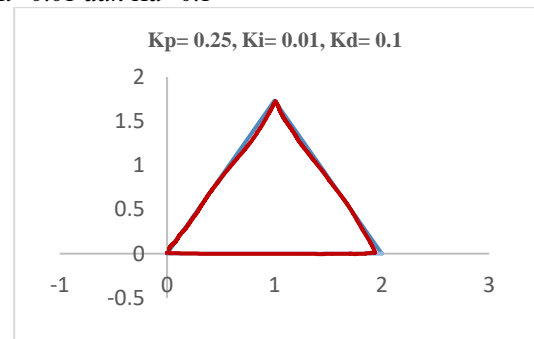
Gambar 11. Hasil Percobaan 1.

Gambar 11 menunjukkan bahwa gerakan robot stabil dengan sedikit penyimpangan pada lintasan, menandakan perlunya perbaikan parameter PID. Data uji posisi lintasan pada Tabel 12 menunjukkan Error kecil: rata-rata pada sumbu x adalah 0.33%, y adalah 0.084%, dan theta adalah 0.013%. Meskipun ada Error orientasi, nilainya rendah dan kontrol arah robot relatif stabil sepanjang lintasan.

TABEL 12
PERCOBAAN I PADA LINTASAN 2

		Waypoint		
		1	2	3
Target	x (m)	2	2	0
	y (m)	0	2	0
	Theta (rad/s)	0	0	0
Posisi	x (m)	1.951	1.033	0.017
	y (m)	0.007	1.695	0.046
	Theta (rad/s)	-0.017	-0.023	-0.001
Error	x (m)	0.049	0.967	-0.017
	y (m)	-0.007	0.305	-0.046
	Theta (rad/s)	0.017	0.023	0.001

- Percobaan PID Lintasan dengan Parameter $K_p=0.25$, $K_i=0.01$ dan $K_d=0.1$



Gambar 12. Hasil Percobaan 2.

Gambar 12 menunjukkan lintasan robot dengan parameter PID $K_p=0.25$, $K_i=0.01$, & $K_d=0.1$. Lintasan robot terlihat cukup stabil, dengan pergerakan yang mendekati bentuk ideal. Robot juga berhasil mengikuti lintasan tanpa *overshoot* atau *undershoot* besar, namun sedikit penyimpangan terlihat pada beberapa titik, terutama pada transisi dari Waypoint 4/ Start ke Waypoint 1.

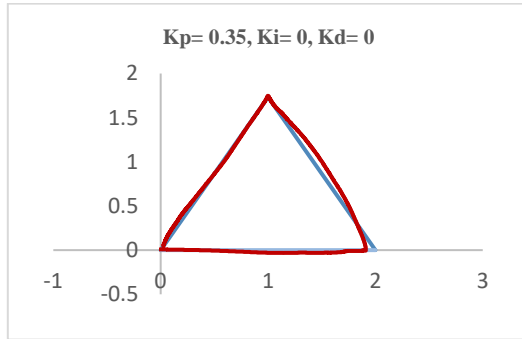
TABEL 13
PERCOBAAN II PADA LINTASAN 2

		Waypoint		
		1	2	3
Target	x (m)	2	2	0
	y (m)	0	2	0
	Theta (rad/s)	0	0	0
Posisi	x (m)	1.950	1.024	0.028
	y (m)	0.004	1.688	0.040
	Theta (rad/s)	0.006	-0.012	-0.003
Error	x (m)	0.05	0.976	-0.028
	y (m)	-0.004	0.312	-0.04
	Theta (rad/s)	-0.006	0.012	0.003

Berdasarkan Tabel 13, Error pada sumbu x dan y menunjukkan hasil yang cukup baik, dengan Error rata-rata untuk sumbu x = 0.33% untuk sumbu y = 0.089% dan theta = 0.003 % . Hal ini mengindikasikan bahwa robot mendekati posisi yang diharapkan dengan akurasi yang cukup tinggi,

meskipun ada peningkatan *Error* pada sumbu x dibandingkan sumbu y.

- *Percobaan PID Lintasan dengan Parameter $K_p= 0.35$, $K_i= 0$ dan $K_d=0$*



Gambar 13. Hasil Percobaan 3.

Gambar 13 menampilkan lintasan robot dengan parameter PI $K_p= 0.35$, $K_i=0$, $K_d= 0$. Terlihat bahwa pergerakan robot cenderung stabil, namun masih ada penyimpangan pada transisi antara *Waypoint 3/Start* ke *Waypoint 1*.

TABEL 14
PERCOBAAN III PADA LINTASAN 2

	Waypoint		
	1	2	3
Target	x (m)	2	0
	y (m)	0	0
	Theta (rad/s)	0	0
Posisi	x (m)	1.950	0.007
	y (m)	-0.001	0.049
	Theta (rad/s)	-0.009	-0.016
Error	x (m)	0.05	-0.007
	y (m)	0.001	-0.049
	Theta (rad/s)	0.009	0.016

Berdasarkan Tabel 14, *Error* pada sumbu x dan y menunjukkan hasil yang cukup baik, dengan *Error* rata-rata sumbu x = 0.014% , sumbu y = 0.00067 % dan theta = 0.004 % . Perbedaan ini dapat disebabkan oleh tuning parameter yang belum optimal tanpa bantuan K_i dan K_d .

TABEL 15
PERBANDINGAN KESELURUHAN PERCOBAAN LINTASAN 2

Trial	Error (x)	Error (y)	Error (Theta)	Kesimpulan
1	0.33%	0.084%	0.013 %	Kendali robot belum optimal.
2	0.33%	0.089%	0.003 %	Hasil yang paling signifikan dari percobaan sebelumnya.
3	0.014%	0.00067%	0.004%	Kontrol PID mulai memperbaiki stabilitas arah robot.

Dari ketiga percobaan pengujian PID posisi pada robot *Trainer kit*, Tabel 15 merupakan perbandingan keseluruhan percobaan pada lintasan 2. Ditemukan percobaan 3 dengan

parameter $K_p= 0.25$, $K_i= 0.01$, $K_d= 0.1$ terlihat bahwasannya Robot bergerak lebih stabil sehingga menghasilkan gerakan yang lebih stabil dan presisi.

Meskipun *Error* pada sumbu x dan y sedikit lebih tinggi, kemampuan robot dalam menjaga orientasi lebih baik, membuat percobaan ini menjadi pilihan yang lebih stabil.

3) Perbandingan Pengujian Lintasan 2 Menggunakan PID dan Tanpa Menggunakan PID

TABEL 16
PERBANDINGAN PENGGUNAAN PID DAN TANPA PID

Error	Tanpa PID	PID
X (%)	0.33%	0.22%
Y (%)	0.09%	0.05%
Theta	0.007%	0.006%

Tabel 16 menunjukkan perbandingan error pada posisi dan orientasi (theta) robot dengan dan tanpa penggunaan kontrol PID. Tabel ini mengukur tiga parameter: error pada sumbu X, sumbu Y, dan orientasi (Theta). Pada sumbu X, tanpa PID, error tercatat sebesar 0,33%, sedangkan dengan PID, error menurun menjadi 0,22%. Untuk sumbu Y, error tanpa PID sebesar 0,09% dan menurun menjadi 0,05% dengan PID. Pada orientasi (Theta), error tanpa PID sebesar 0,007%, dan sedikit berkurang menjadi 0,006% dengan PID. Dari tabel ini, terlihat bahwa penggunaan kontrol PID secara signifikan mengurangi error pada sumbu X dan Y, serta sedikit mengurangi error pada orientasi, sehingga meningkatkan akurasi pergerakan robot.

IV. KESIMPULAN

Penerapan invers kinematik pada robot mecanum untuk mengontrol kecepatan masing-masing roda lebih spesifik dengan menambahkan kontrol PID efektif digunakan untuk mengontrol posisi robot mecanum dengan mengoreksi pergerakan secara berulang berdasarkan error yang dihitung melalui odometri. PID berhasil membantu robot mencapai waypoint dengan akurasi tinggi, mengurangi kesalahan posisi, serta menjaga pergerakan yang stabil dan responsive dengan kecepatan masing-masing roda yang ditentukan. Penelitian ini juga memaparkan penerapan inverse kinematics (IK) pada roda mecanum, dengan kontrol PID yang signifikan mempengaruhi kestabilan pergerakan robot hingga mencapai target yang diinginkan pada setiap percobaan yang dilakukan dalam penelitian ini.

Hasil pengujian menunjukkan adanya error dalam lintasan robot. Pada pengujian dengan kontrol PID, error untuk sumbu X dan Y masing-masing sebesar 0,1% dan 0,02%, sedangkan untuk orientasi (theta) sebesar 0,003%. Tanpa PID, error rata-rata meningkat menjadi 0,2% untuk sumbu X, 0,06% untuk sumbu Y, dan 0,01% untuk theta. Dari berbagai parameter yang diuji, percobaan kedua dengan parameter PID $K_p = 0,25$, $K_i = 0,01$, dan $K_d = 0,1$ menunjukkan hasil yang paling stabil dan efektif dalam mengontrol pergerakan robot dengan error orientasi dan posisi yang lebih kecil dibandingkan percobaan lainnya. Hal ini menunjukkan bahwa kombinasi parameter tersebut paling optimal untuk mencapai kontrol pergerakan yang akurat dan stabil pada robot mecanum.

REFERENCES

- [1] F. R. Iskandar, I. Sucahyo, and M. Yantidewi, "Penerapan Metode Invers kinematik Pada Kontrol Gerak Robot Lengan Tiga Derajat Bebas," *Inov. Fis. Indones.*, vol. 9, no. 2, pp. 64–71, 2020, doi: 10.26740/ifi.v9n2.p64-71.
- [2] "applied sciences A Mobile Robot with Omnidirectional Tracks — Design and Experimental Research," 2021.
- [3] P. Sperzyński, J. Bałchanowski, and A. Gronowicz, "Simulation of motion of a mobile robot on uneven terrain," *J. Theor. Appl. Mech.*, vol. 58, no. 2, pp. 541–552, 2020, doi: 10.15632/jtam-pl/119062.
- [4] R. A. Maulana, B. Herawan, M. E. Hidayat, and Fauziah, "Robot Entertainment Berbasis Mecanum Wheel," pp. 7–13, 2023.
- [5] F. Fahmizal, A. Priyatmoko, and A. Mayub, "Implementasi Kinematika Trajectory Lingkaran pada Robot Roda Mecanum," *J. List. Instrumentasi dan Elektron. Terap.*, vol. 3, no. 1, pp. 25–30, 2022, doi: 10.22146/juliet.v3i1.74760.
- [6] A. A. Wardani and Z. Iklima, "Rancang Bangun Automated Guided Vehicle Menggunakan Metode PID Zeigler Nichols," *J. Teknol. Elektro*, vol. 13, no. 1, p. 6, 2022, doi: 10.22441/jte.2022.v13i1.002.
- [7] M. Royhan Darmawan, F. Fitri, and I. Siradjuddin, "Implementasi Persamaan Forward dan Inverse Kinematic pada Mobile Robot Asimetris Beroda 4 Omni Wheels," *J. Elektron. dan Otomasi Ind.*, vol. 10, no. 2, pp. 209–217, 2023, doi: 10.33795/elkolind.v10i2.3641.
- [8] A. N. Aliansyah, M. Safarudin, L. Pagiling, and ..., "Rancang Bangun Kendali Robot Mekanum Menggunakan Metode Face Kontrol," ... *Tek. Elektro dan ...*, 2023, [Online]. Available: <http://jurnal.poliupg.ac.id/index.php/sntei/article/view/3535%0Ahttp://jurnal.poliupg.ac.id/index.php/sntei/article/download/3535/3596>
- [9] S. Rifandi, "Implementasi Metode Behavior-Based, Odometry Dan Pid Pada Mobile Robot Dalam Proses Autodocking Untuk Mengisi Baterai," *J. Tek. Elektro dan Komput. TRIAC*, vol. 9, no. 2, 2022.
- [10] G. Al Azhar, T. Winarno, and S. Izza, "Sistem Distribusi Data Kontrol Pada Differential Drive Mobile Robot Menggunakan Robot Operating System," *Metrotech (Journal Mech. Electr. Technol.)*, vol. 1, no. 3, pp. 19–27, 2022, doi: 10.33379/metrotech.v1i3.1803.
- [11] S. Parulian, T. Pangaribuan, and A. Simamora, "Implementasi Kontrol Lup Tertutup Multi Point Pada Pengatur Temperatur Oven Panggang Roti," *J. ELPOTECS*, vol. 4, no. 1, pp. 38–45, 2021, doi: 10.51622/elpotecs.v4i1.450.
- [12] M. M. Ali, T. Hariyati, M. Y. Pratiwi, and S. Afifah, "Metodologi Penelitian Kuantitatif dan Penerapannya dalam Penelitian," *Educ. Journal.2022*, vol. 2, no. 2, pp. 1–6, 2022.
- [13] Y. Li *et al.*, "Kinematic modeling of a combined system of multiple mecanum-wheeled robots with velocity compensation," *Sensors (Switzerland)*, vol. 20, no. 1, 2020, doi: 10.3390/s20010075.
- [14] Y. Xu, Y. Zhao, and S. L. Zhao, "Grid Line Tracking Omnidirectional Robot Based on Visible Light Sensor and Mecanum Wheel PID Control," *Proc. - 2021 7th Int. Symp. Mechatronics Ind. Informatics, ISMII 2021*, pp. 57–60, 2021, doi: 10.1109/ISMII52409.2021.00019.
- [15] Y. Pratama *et al.*, "Implementasi Kontrol PID untuk Percepatan Rotasi pada Robot Hexapod," *J. Media Inform. Budidarma*, vol. 7, no. 4, pp. 1869–1876, 2023, doi: 10.30865/mib.v7i4.6433.
- [16] M. N. F. Astoro, T. Winarno, and A. Komarudin, "Implementasi Kontrol PID Positioning Robot Pada Titik Pengambilan Objek Shagai Menggunakan Sensor Kamera," *J. Elektron. dan Otomasi Ind.*, vol. 7, no. 2, p. 57, 2021, doi: 10.33795/elkolind.v7i2.195.
- [17] E. W. Suseno, A. Ma'arif, and R. D. Puriyanto, "Tuning Parameter Pengendali PID dengan Metode Algoritma Genetik pada Motor DC," *TELKA - Telekomun. Elektron. Komputasi dan Kontrol*, vol. 8, no. 1, pp. 1–13, 2022, doi: 10.15575/telka.v8n1.1-13.
- [18] A. Rahmadiansyah and M. S. Zuhri, "Maze Solver Robot Dengan Kendali PID Berbasis Arduino Uno Dengan Tuning Melalui Smartphone Android," 2019, [Online]. Available: <https://www.digikey.co.uk/product->