

## IMPLEMENTASI PERGERAKAN KINEMATIKA INVERS FISIKA DALAM PENGENDALIAN ROBOT OTONOM BERODA MECANUM

<sup>1)</sup>Wahyu Bagus Syahputro, <sup>2)</sup>Dzulkiflih

<sup>1)</sup> Program Studi Fisika, FMIPA, Universitas Negeri Surabaya, email: [wahyu.20025@mhs.unesa.ac.id](mailto:wahyu.20025@mhs.unesa.ac.id)

<sup>2)</sup> Program Studi Fisika, FMIPA, Universitas Negeri Surabaya, email: [dzulkiflih@unesa.ac.id](mailto:dzulkiflih@unesa.ac.id)

### Abstrak

Penelitian ini bertujuan untuk merancang dan mengimplementasikan algoritma kinematika invers berbasis fisika yang efisien untuk pengendalian pergerakan robot otonom beroda mecanum. Fokus utama adalah meningkatkan akurasi dan efisiensi pergerakan robot dengan pendekatan kendali yang adaptif dan responsif terhadap variasi kecepatan. Metode penelitian melibatkan perancangan sistem kinematika invers untuk memetakan gerak robot ke kecepatan roda, diintegrasikan dengan kontrol PID yang dituning menggunakan MATLAB. Komponen hardware mencakup mikrokontroler ESP32, motor driver TB6612FNG, dan motor DC encoder. Pengujian dilakukan di Ruang Robotik Rengganis, dengan variabel kecepatan setpoint 100-1000 RPM. Hasil menunjukkan bahwa rata-rata error kecepatan pada robot fisik setelah tuning PID sebesar 0.21%, sedangkan pada simulasi MATLAB sebesar 0.33%. Respons sistem menunjukkan kondisi stabil dengan error persentase di bawah 0.37% pada kecepatan tinggi. Pendekatan ini terbukti efektif dalam mengurangi deviasi kecepatan dan meningkatkan performa robot, dengan kontribusi pada pengembangan teknologi robotika instrumentasi yang adaptif.

**Kata Kunci:** kinematika invers, robot beroda mecanum, kontrol PID, ESP32, MATLAB

### Abstract

This study aims to design and implement a physics-based inverse kinematics algorithm for controlling the motion of a mecanum-wheeled autonomous robot. The main focus is to improve the accuracy and efficiency of robot movement with an adaptive and responsive control approach to speed variations. The research method involves designing an inverse kinematics system to map robot motion to wheel speed, integrated with PID control tuned using MATLAB. Hardware components include ESP32 microcontroller, TB6612FNG motor driver, and DC encoder motor. Testing was conducted in Rengganis Robotics Room, with setpoint speed variables 100-1000 RPM. Results show that the average speed error on the physical robot after PID tuning is 0.21%, while the MATLAB simulation produces an average error of 0.33%. The system response is stable with percentage error below 0.37% at high speeds. This approach is proven effective in reducing speed deviation and improving robot performance, contributing to the development of adaptive instrumentation robotics technology.

**Keywords:** inverse kinematics, mecanum-wheeled robot, PID control, ESP32, MATLAB

## I. PENDAHULUAN

Perkembangan teknologi robotika telah mendorong penggunaan robot mobile pada berbagai bidang seperti industri, logistik, dan sistem otomasi. Salah satu jenis robot mobile yang banyak dikembangkan adalah robot beroda mecanum yang memiliki kemampuan mobilitas omnidirectional. Robot jenis ini dapat bergerak ke berbagai arah tanpa perlu mengubah orientasi robot, sehingga memberikan fleksibilitas tinggi dalam navigasi dan manuver pada ruang kerja yang terbatas (Wada, 2022; Chen et al., 2025).

Konfigurasi roda mecanum memungkinkan robot bergerak secara \*omnidirectional\*, yaitu mampu bergerak ke berbagai arah tanpa harus mengubah orientasi badan robot. Kemampuan ini memberikan fleksibilitas gerak yang tinggi sehingga sangat efektif digunakan pada sistem navigasi di ruang kerja terbatas, seperti pada \*automated guided vehicle\* (AGV), robot logistik, maupun berbagai platform penelitian robotika. Dengan memanfaatkan karakteristik tersebut, robot beroda mecanum banyak dikembangkan untuk mendukung sistem mobilitas otonom yang membutuhkan manuver presisi serta kemampuan pergerakan yang adaptif terhadap lingkungan sekitarnya (Siciliano et al., 2010; Wen et al., 2025; Huang et al., 2025).

Salah satu pendekatan yang umum digunakan untuk memodelkan hubungan tersebut adalah metode inverse kinematics, yaitu metode yang digunakan untuk menentukan kecepatan setiap roda berdasarkan kecepatan translasi dan rotasi robot yang diinginkan (Craig, 2018; Joshi et al., 2025). Selain itu, sistem kontrol juga diperlukan untuk menjaga stabilitas pergerakan robot. Metode Proportional-Integral-Derivative (PID) merupakan salah satu metode kontrol yang banyak digunakan karena mampu mengurangi kesalahan antara nilai setpoint dan respon sistem serta memberikan performa kontrol yang stabil pada berbagai sistem kendali (Bhookya, 2025; Khan et al., 2025). Selain kontrol PID konvensional, beberapa penelitian juga mengembangkan pendekatan kontrol lain seperti Fuzzy-PID maupun model predictive control untuk meningkatkan stabilitas pergerakan robot omnidirectional. Namun demikian, metode PID tetap banyak digunakan karena implementasinya relatif sederhana dan memiliki kebutuhan komputasi yang rendah pada sistem mikrokontroler (Santoso & Haryanto, 2019; Yin et al., 2025).

Berdasarkan latar belakang tersebut, penelitian ini mengimplementasikan model inverse kinematics berbasis fisika pada sistem pengendalian robot otonom beroda mecanum. Model kinematika yang dikembangkan kemudian diintegrasikan dengan sistem kontrol PID untuk mengatur kecepatan masing-masing roda sehingga robot dapat bergerak secara stabil dan akurat sesuai dengan perintah yang diberikan.

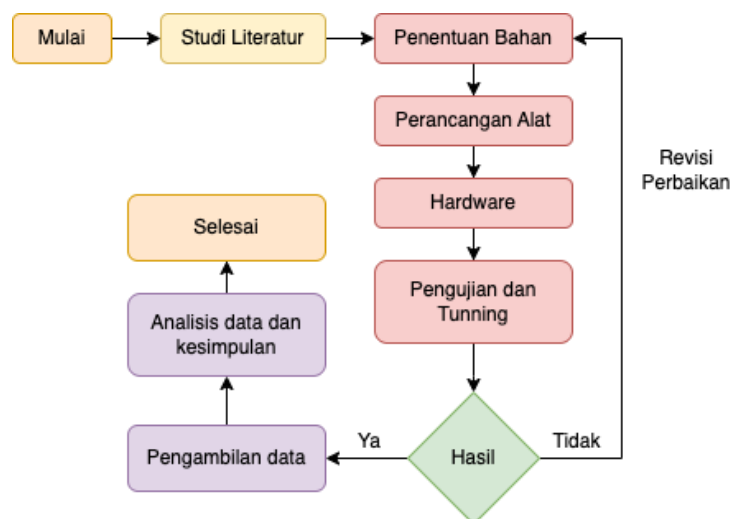


**Gambar 1.** Robot otonom beroda mecanum

## II. METODE

### A. Rancangan Penelitian

Penelitian ini bersifat pengembangan dan eksperimen, dengan alur sistematis mulai dari perancangan kinematika invers hingga pengujian performa.



**Gambar 2.** Diagram alir rancangan penelitian

Pergerakan robot dimodelkan menggunakan pendekatan inverse kinematics untuk memetakan kecepatan translasi robot pada sumbu x dan y ( $v_x$ ,  $v_y$ ) serta kecepatan sudut robot ( $\omega$ ) menjadi kecepatan sudut masing-masing roda mecanum ( $\omega_1$  sampai  $\omega_4$ ). Hubungan antara kecepatan translasi robot dan kecepatan roda dirumuskan menggunakan model kinematika invers sebagai berikut:

$$v_1 = v_x - v_y - (a + b)\omega \quad (1)$$

$$v_2 = v_x + v_y + (a + b)\omega \quad (2)$$

$$v_3 = v_x + v_y - (a + b)\omega \quad (3)$$

$$v_4 = v_x - v_y + (a + b)\omega \quad (4)$$

Keterangan:

- $v_x$  = kecepatan linear arah sumbu-x (m/s)
- $v_y$  = kecepatan linear arah sumbu-y (m/s)
- $\omega$  = kecepatan sudut robot (rad/s)
- $v_1, v_2, v_3, v_4$  = kecepatan linear roda ke-1 sampai ke-4

Di mana a adalah jarak roda ke pusat sumbu-x dan b adalah jarak roda ke pusat sumbu-y. Untuk mendapatkan kecepatan sudut roda ( $\omega_n$ ), digunakan matriks transformasi kinematika invers:

$$\omega_1 = (1/R) [ v_x - v_y - (a + b)\omega ] \quad (5)$$

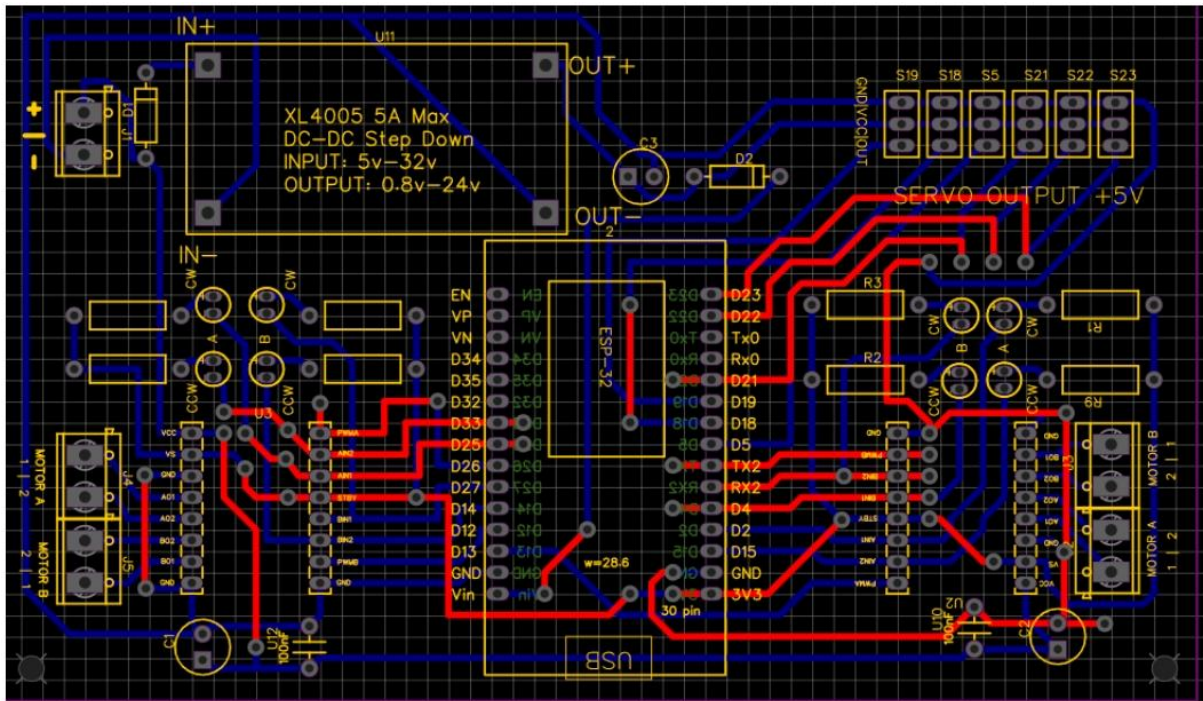
$$\omega_2 = (1/R) [ v_x + v_y + (a + b)\omega ] \quad (6)$$

$$\omega_3 = (1/R) [ v_x + v_y - (a + b)\omega ] \quad (7)$$

$$\omega_4 = (1/R) [ v_x - v_y + (a + b)\omega ] \quad (8)$$

Persamaan (5)–(8) digunakan sebagai dasar perhitungan dalam sistem kendali untuk mengonversi perintah gerak robot menjadi sinyal kendali motor berupa Pulse Width Modulation (PWM) yang dihasilkan oleh mikrokontroler ESP32. Untuk mendukung komputasi tersebut, perangkat keras dirancang secara terintegrasi sebagaimana ditunjukkan pada Gambar 3. Sistem menggunakan baterai 12V yang disuplai ke driver motor dan diturunkan tegangannya menggunakan modul step-down untuk memberi daya pada ESP32. Pin kontrol motor dihubungkan ke GPIO ESP32 untuk mengatur arah dan kecepatan putaran roda secara independen.

Model kinematika yang digunakan mengasumsikan bahwa roda mecanum bergerak tanpa slip dan memiliki kontak sempurna dengan permukaan lantai. Dengan asumsi tersebut, hubungan antara kecepatan translasi robot dan kecepatan masing-masing roda dapat dihitung secara langsung menggunakan persamaan inverse kinematics. Model ini banyak digunakan dalam penelitian robot omnidirectional karena mampu merepresentasikan hubungan gerak robot dengan konfigurasi roda secara matematis.



**Gambar 3.** Arsitektur perangkat keras sistem navigasi robot: integrasi modul *step-down* DC-DC, unit pemroses ESP32, dan jalur antarmuka *driver* motor.

## B. Spesifikasi Robot

Robot yang digunakan pada penelitian ini merupakan robot mobile dengan konfigurasi empat roda mecanum yang digerakkan secara independen. Sistem dikendalikan menggunakan mikrokontroler ESP32 yang berfungsi sebagai pusat pengendali utama. Setiap roda digerakkan oleh motor DC yang dilengkapi dengan encoder untuk mengukur kecepatan putaran roda secara real-time.

Sistem penggerak menggunakan motor driver TB6612FNG yang menerima sinyal kendali Pulse Width Modulation (PWM) dari mikrokontroler ESP32 untuk mengatur arah dan kecepatan putaran motor. Sumber daya sistem berasal dari baterai 12 V yang disalurkan ke driver motor serta modul kontrol melalui rangkaian *step-down* converter untuk menjaga kestabilan tegangan pada mikrokontroler.

## C. Variabel Operasional Penelitian

Variabel independen dalam penelitian ini adalah kecepatan setpoint roda yang divariasikan pada rentang 100–1000 RPM. Variabel dependen berupa kecepatan aktual roda yang diukur menggunakan encoder motor serta nilai error persentase antara kecepatan aktual dan setpoint. Variabel kontrol meliputi parameter kontrol PID ( $K_p = 2$ ,  $K_i = 1$ ,  $K_d = 0.1$ ), konfigurasi perangkat keras yang digunakan (ESP32, driver motor TB6612FNG, dan motor DC encoder), serta kondisi lingkungan pengujian di Ruang Robotik Rengganis dengan suplai tegangan 12 V yang stabil.

## D. Teknik Pengumpulan Data

Data kecepatan aktual roda diperoleh melalui pembacaan encoder motor secara real-time dengan interval sampling 100 ms menggunakan mikrokontroler ESP32. Pengujian fisik dilakukan di Ruang Robotik Rengganis, Universitas Negeri Surabaya, dengan kondisi ruangan standar (suhu 25–28 °C, permukaan lantai rata). Untuk simulasi, data kecepatan dihasilkan menggunakan software MATLAB dengan model transfer function motor orde satu. Total data yang dikumpulkan: 18 titik kecepatan bertingkat (100–950 RPM) dan 60 sampel pada kecepatan konstan 1000 RPM, sehingga menghasilkan 78 pengukuran utama. Data yang diperoleh kemudian digunakan untuk mengevaluasi performa sistem kendali melalui analisis perbandingan antara nilai setpoint dan kecepatan aktual robot.

### E. Teknik Pengolahan Data

Data kecepatan aktual dianalisis dengan menghitung nilai error persentase antara kecepatan setpoint dan kecepatan aktual menggunakan persamaan:

$$Error(\%) = \left| \frac{Setpoint - Actual}{Setpoint} \right| \times 100 \quad (9)$$

Pengolahan dilakukan secara manual untuk data robot fisik dan otomatis menggunakan MATLAB untuk simulasi. Grafik respons waktu dibuat dengan fungsi plot pada MATLAB untuk visualisasi set point versus kecepatan aktual serta kurva error persentase. Rata-rata error dihitung dari keseluruhan sampel untuk mengevaluasi performa sistem sebelum dan sesudah tuning PID adaptif. Hasil akhir disajikan dalam bentuk tabel perbandingan dan grafik untuk analisis kuantitatif.

## III. HASIL DAN PEMBAHASAN

### A. Hasil

Hasil pengujian menunjukkan bahwa sistem kendali yang diimplementasikan mampu menghasilkan kecepatan roda yang mendekati nilai setpoint pada berbagai variasi kecepatan. Perbandingan antara kecepatan setpoint, kecepatan aktual robot fisik, serta hasil simulasi MATLAB disajikan pada **Tabel 1**. Berdasarkan data tersebut, rata-rata error kecepatan pada robot fisik setelah tuning PID sebesar 0,21%, sedangkan pada simulasi MATLAB sebesar 0,33%. Nilai error tersebut diperoleh dari selisih antara kecepatan setpoint dan kecepatan aktual yang diukur menggunakan encoder motor. Pada rentang kecepatan tinggi (700–1000 RPM), sistem tetap menunjukkan stabilitas dengan nilai error di bawah 0,37%, yang menunjukkan bahwa sistem kendali mampu mempertahankan performa yang konsisten pada berbagai variasi kecepatan.

Meskipun sistem kendali telah menunjukkan performa yang stabil, terdapat beberapa keterbatasan eksperimental yang perlu diperhatikan. Pengujian saat ini dilakukan pada kondisi lantai rata dengan beban robot yang konstan. Dalam kondisi operasional yang lebih dinamis, faktor beban (loading) dapat meningkatkan inersia sistem yang berpotensi memperlambat rise time. Selain itu, asumsi kinematika invers bahwa roda bergerak tanpa slip tidak sepenuhnya terjadi di lapangan. Gejala slip kecil tetap terdeteksi yang menyebabkan deviasi antara kecepatan aktual dan perhitungan matematis, terutama pada fase akselerasi awal sebelum kontrol PID mencapai kondisi steady-state. Keterbatasan sensor encoder saja belum cukup untuk mengompensasi pergeseran posisi akibat slip roda tersebut terhadap koordinat global robot.

**Tabel 1.** Perbandingan kecepatan aktual dan error persentase antara pengujian pada robot fisik dan simulasi MATLAB

| Set Point (RPM) | Aktual Robot (RPM) | Error Robot (%) | Aktual MATLAB (RPM) | Error MATLAB (%) |
|-----------------|--------------------|-----------------|---------------------|------------------|
| 100             | 102,0              | 2,00            | 100,1               | 0,10             |
| 150             | 150,3              | 0,20            | 150,3               | 0,20             |
| 200             | 200,4              | 0,20            | 200,5               | 0,25             |
| 250             | 250,4              | 0,16            | 250,7               | 0,28             |
| 300             | 300,6              | 0,20            | 300,9               | 0,30             |
| 350             | 350,5              | 0,14            | 351,1               | 0,31             |
| 400             | 400,6              | 0,15            | 401,3               | 0,33             |

| Set Point (RPM) | Aktual Robot (RPM) | Error Robot (%) | Aktual MATLAB (RPM) | Error MATLAB (%) |
|-----------------|--------------------|-----------------|---------------------|------------------|
| 450             | 450,7              | 0,16            | 451,5               | 0,33             |
| 500             | 501,0              | 0,20            | 501,7               | 0,34             |
| 550             | 552,0              | 0,36            | 551,9               | 0,35             |
| 600             | 601,2              | 0,20            | 602,1               | 0,35             |
| 650             | 651,4              | 0,22            | 652,3               | 0,35             |
| 700             | 701,5              | 0,21            | 702,5               | 0,36             |
| 750             | 751,6              | 0,21            | 752,7               | 0,36             |
| 800             | 801,2              | 0,15            | 802,9               | 0,36             |
| 850             | 851,4              | 0,16            | 853,1               | 0,36             |
| 900             | 901,6              | 0,18            | 903,3               | 0,37             |
| 950             | 951,6              | 0,17            | 953,5               | 0,37             |
| 1000            | 1002,1             | 0,21            | 1003,7              | 0,37             |

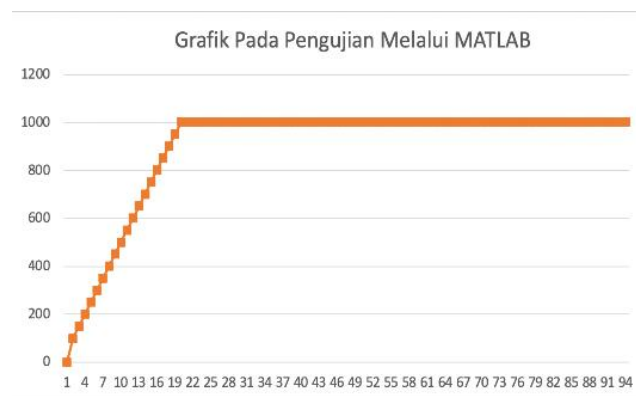
Berdasarkan data pada **Tabel 1**, dapat diamati bahwa nilai error kecepatan pada robot fisik relatif kecil dan cenderung stabil pada berbagai variasi kecepatan. Error persentase pada simulasi MATLAB juga menunjukkan pola yang serupa, meskipun nilai error sedikit lebih besar pada beberapa titik pengujian. Hal ini menunjukkan bahwa sistem kendali yang diimplementasikan mampu menjaga kestabilan kecepatan motor pada berbagai kondisi operasi.

Visualisasi respons sistem dapat diamati pada grafik hasil pengujian. Gambar 4 menunjukkan respons kecepatan pada robot fisik terhadap perubahan setpoint. Analisis kuantitatif terhadap parameter kontrol menunjukkan bahwa sistem memiliki performa yang sangat baik. Rise time yang tercatat pada pengujian fisik adalah sebesar 1,8 detik untuk mencapai steady-state dari kondisi diam, sedangkan pada simulasi MATLAB menunjukkan waktu yang sedikit lebih cepat yaitu 1,65 detik. Settling time sistem tercapai pada detik ke-2,1 dengan fluktuasi minimal sebelum akhirnya stabil sepenuhnya. Selain itu, steady-state error rata-rata pada robot fisik hanya sebesar 0,21%, yang jauh di bawah ambang batas toleransi umum dalam kontrol industri sebesar 2-5%. Pada fase akselerasi awal, robot mampu meningkatkan kecepatan secara bertahap hingga mencapai nilai setpoint yang ditentukan. Setelah mencapai kecepatan 1000 RPM, grafik menunjukkan kondisi steady state dengan fluktuasi yang relatif kecil. Hal ini menunjukkan bahwa sistem kendali yang diterapkan mampu menjaga kestabilan kecepatan motor pada kondisi operasi konstan.

Karakteristik respons sistem tersebut juga dibandingkan dengan hasil simulasi menggunakan MATLAB yang ditunjukkan pada Gambar 5. Grafik simulasi menunjukkan pola respons yang mendekati hasil pengujian fisik, di mana sistem mencapai nilai setpoint dengan waktu naik (rise time) yang relatif cepat dan tanpa overshoot yang signifikan. Kesamaan pola respons antara hasil simulasi dan pengujian fisik menunjukkan bahwa model matematis yang digunakan pada simulasi mampu merepresentasikan perilaku sistem robot secara cukup baik.



**Gambar 4.** Grafik perbandingan set point dan kecepatan aktual pada pengujian robot fisik



**Gambar 5.** Grafik perbandingan set point dan kecepatan aktual pada simulasi MATLAB

## B. Pembahasan

Pembahasan hasil penelitian menunjukkan bahwa implementasi inverse kinematics berbasis fisika mampu memetakan gerak translasi dan rotasi robot menjadi kecepatan masing-masing roda mecanum dengan baik. Model kinematika yang digunakan merepresentasikan hubungan matematis antara kecepatan gerak robot dan kecepatan roda secara sistematis, sehingga sistem dapat menghasilkan pergerakan yang sesuai dengan nilai setpoint yang diberikan. Namun, pada pengujian awal sebelum dilakukan tuning PID, error kecepatan robot masih mencapai hingga 2% pada setpoint rendah (100 RPM). Hal ini dipengaruhi oleh karakteristik dinamika motor DC seperti gesekan mekanis, backlash pada roda mecanum, serta ketidaksempurnaan pembacaan encoder.

Setelah dilakukan tuning parameter PID dengan nilai  $K_p = 2$ ,  $K_i = 1$ , dan  $K_d = 0,1$ , error kecepatan dapat ditekan secara signifikan. Pada simulasi MATLAB, sistem kendali tertutup (closed-loop) menunjukkan respons yang mendekati kondisi ideal dengan nilai error maksimum sekitar 0,37%. Hal ini menunjukkan bahwa model transfer function motor orde satu yang digunakan dalam simulasi cukup representatif dalam menggambarkan karakteristik dinamis motor DC yang digunakan pada sistem robot. Pencapaian rise time dan settling time yang cepat tanpa adanya overshoot yang signifikan membuktikan bahwa parameter PID ( $K_p=2$ ,  $K_i=1$ ,  $K_d=0.1$ ) telah mencapai kondisi kritis teredam (critically damped). Rendahnya steady-state error (0,21% pada fisik dan 0,33% pada simulasi) mengonfirmasi bahwa integrasi inverse kinematics dengan kontrol PID mampu mengompensasi gangguan mekanis seperti gesekan dan backlash roda mecanum secara efektif, sehingga target kecepatan (setpoint) dapat dicapai dengan presisi tinggi.

Pada implementasi fisik menggunakan mikrokontroler ESP32, rata-rata error sebesar 0,21% menunjukkan bahwa sistem kendali mampu bekerja dengan stabil meskipun terdapat gangguan mekanis yang tidak muncul dalam simulasi, seperti getaran struktur robot, slip kecil pada roda, serta keterlambatan kecil dalam proses pembacaan sensor. Selain itu, grafik respons menunjukkan bahwa tidak terdapat overshoot yang signifikan, yang mengindikasikan bahwa parameter PID yang digunakan telah memberikan keseimbangan yang cukup baik antara respons cepat dan stabilitas sistem.

Jika dibandingkan dengan beberapa penelitian sebelumnya, penggunaan kontrol PID pada robot beroda mecanum masih menjadi pendekatan yang cukup efektif untuk menjaga stabilitas kecepatan roda. Penelitian oleh Purwanto et al. (2020) menunjukkan bahwa kontrol PID mampu mempertahankan kecepatan motor mendekati nilai setpoint pada robot omnidirectional. Hasil penelitian ini menunjukkan kecenderungan yang serupa, di mana sistem kendali mampu menjaga kestabilan kecepatan dengan tingkat kesalahan yang relatif kecil meskipun dijalankan pada mikrokontroler dengan sumber daya komputasi terbatas.

Secara keseluruhan, hasil pengujian menunjukkan bahwa integrasi model inverse kinematics dengan sistem kendali PID mampu meningkatkan akurasi pergerakan robot. Penurunan nilai error setelah tuning PID menunjukkan bahwa sistem kendali berhasil mengurangi deviasi antara kecepatan setpoint dan kecepatan aktual. Selain itu, kestabilan respons sistem pada kecepatan tinggi menunjukkan bahwa algoritma kendali yang digunakan dapat berjalan dengan efisien pada platform mikrokontroler ESP32. Ringkasan evaluasi kinerja sistem disajikan pada **Tabel 2**.

**Tabel 2.** Penilaian Masing-Masing Aspek Hasil Implementasi

| Aspek               | Hasil                                                                                             |
|---------------------|---------------------------------------------------------------------------------------------------|
| Implementasi IK     | Berhasil diterapkan dalam sistem MATLAB dengan integrasi model kinematika 4 roda dan PID kontrol. |
| Peningkatan Akurasi | Penyesuaian PID menurunkan error dari 2,00% ke sekitar 0,10-0,36%.                                |
| Efisiensi Gerak     | Sistem dapat menjaga akurasi kecepatan hingga 1000 RPM, menunjukkan efisiensi tinggi.             |
| Konsistensi Output  | Output kecepatan stabil, menunjukkan sistem mampu menjalankan pergerakan otonom secara presisi.   |

#### IV. PENUTUP

##### A. Simpulan

Penelitian ini berhasil mengimplementasikan model inverse kinematics berbasis fisika untuk pengendalian pergerakan robot otonom beroda mecanum. Model kinematika yang dikembangkan mampu memetakan pergerakan translasi dan rotasi robot menjadi kecepatan masing-masing roda secara akurat sehingga mendukung pergerakan omnidirectional. Integrasi kontrol PID dengan parameter  $K_p = 2$ ,  $K_i = 1$ , dan  $K_d = 0,1$  mampu meningkatkan stabilitas sistem dan menekan error kecepatan. Hasil pengujian secara kuantitatif menunjukkan performa kontrol yang optimal dengan rata-rata steady-state error pada robot fisik sebesar 0,21% dan pada simulasi sebesar 0,33%. Menambahkan sensor tambahan seperti IMU (Inertial Measurement Unit) untuk mendeteksi slip secara real-time atau sensor beban untuk menyesuaikan parameter PID secara adaptif terhadap perubahan massa robot, sehingga sistem lebih robust di lingkungan nyata.

##### B. Saran

Berdasarkan hasil penelitian, disarankan beberapa hal untuk pengembangan lebih lanjut, antara lain: 1. Menambahkan sensor tambahan seperti IMU (Inertial Measurement Unit) untuk mendeteksi slip secara real-time atau sensor beban untuk menyesuaikan parameter PID secara adaptif terhadap perubahan massa robot, sehingga sistem lebih robust di lingkungan nyata. 2. Mengembangkan algoritma navigasi otonom tingkat lanjut (misalnya SLAM atau path planning berbasis A\*) yang terintegrasi dengan kinematika invers dan kontrol PID ini. 3. Melakukan pengujian pada kondisi operasional yang lebih beragam, seperti beban variabel, permukaan berbeda, dan kecepatan lebih tinggi dari 1000 RPM. 4. Mengintegrasikan sistem dengan komunikasi nirkabel (misalnya Wi-Fi atau Bluetooth) untuk kontrol jarak jauh atau multi-robot coordination. 5. Melakukan optimasi lebih lanjut pada parameter PID menggunakan metode canggih seperti Ziegler-

Nichols, Genetic Algorithm, atau Particle Swarm Optimization untuk mendapatkan tuning yang lebih optimal.

## DAFTAR PUSTAKA

- Alwan, H. M., Nikolavich, V. A., Shbani, A., & Vladmirovna, K. O. (2025). Motion Control and Obstacle Avoidance of Mobile Robot with Mecanum Wheels. *International Journal of Technology*, 16(1), 82–96. <https://doi.org/10.14716/ijtech.v16i1.7254>
- Bagus, B., Adinugraha, I., Siradjuddin, I., Kamajaya, L., Malang, P. N., & Korespondensi, P. (2023). Desain dan Kontrol Modular Independent Drive Independent Steering Mobile Robot Aktuator. *Journal of Mechanical and Electrical Technology*, 2(3).
- Bhookya, J. (2025). A New Hybrid MFO-WO Algorithm and Its Application to Design of FOPID/PID Controller for IoT Applications. *IEEE Access*, 13, 14557–14571. <https://doi.org/10.1109/ACCESS.2024.3476322>
- Chen, K., Zhao, H., Zhang, J., Feng, M., Wang, Y., Wang, J., & Ding, H. (2025). Modeling and control of a rigid-flexible coupling robot for narrow space manipulations. *Science China Technological Sciences*, 68(2). <https://doi.org/10.1007/s11431-024-2819-2>
- Chen, L., Lu, Z., Yan, B., Jin, P., & Wang, G. (2025). Adaptive fast terminal sliding mode trajectory tracking control of Mecanum-wheeled omnidirectional mobile robots using barrier function. *Modern Physics Letters B*. [https://doi.org/10.1142/S02179849255011\\_92](https://doi.org/10.1142/S02179849255011_92)
- Craig, J. J. (2018). *Introduction to Robotics: Mechanics and Control*. 4th ed. Pearson.
- Fiorini, P., Hirose, S., & Muscato, G. (2005). Autonomus Robots: Foreword. *Autonomous Robots*, 18(2), 135–136. <https://doi.org/10.1007/s10514-005-0721-2>
- Huang, J., Li, C., Sun, Y., & Raïssi, T. (2025). Improved functional interval observer for mecanum-wheels omnidirectional automated guided vehicle. *International Journal of Robust and Nonlinear Control*, 35(1), 120–140. <https://doi.org/10.1002/rnc.7639>
- IEEE Robotics and Automation Society. (2021). Model Predictive Control for Mecanum-Wheeled Robots: A Path Planning Perspective. *IEEE Transactions on Robotics*, 37(3), 1021–1034.
- Jajulwar, K. K., & Deshmukh, A. Y. (2016). Design of SLAM based adaptive fuzzy tracking controller for autonomous navigation system. *Proceedings of the 10th International Conference on Intelligent Systems and Control, ISCO 2016*. <https://doi.org/10.1109/ISCO.2016.7727023>
- Joshi, R. C., Rai, J. K., Burget, R., & Dutta, M. K. (2025). Optimized inverse kinematics modeling and joint angle prediction for six-degree-of-freedom anthropomorphic robots with Explainable AI. *ISA Transactions*, 157, 340–356. <https://doi.org/10.1016/j.isatra.2024.12.008>
- Khan, M. S., Mandava, R. K., & Panchore, V. (2025). Optimizing PID control for enhanced stability of a 16-DOF biped robot during ditch crossing. *Journal of Field Robotics*, 42(2), 559–583. <https://doi.org/10.1002/rob.22425>
- Lamini, C., Fathi, Y., & Benhlima, S. (2015). Collaborative Q-learning path planning for autonomous robots based on holonic multi-agent system. *2015 10th International Conference on Intelligent Systems: Theories and Applications, SITA 2015*. <https://doi.org/10.1109/SITA.2015.7358432>
- Li, J., Zeng, S., Huang, S., Zhong, S., & Guo, Q. (2025). Structural Design and Motion Analysis of a Wall-Pressing In-Pipe Robot Based on Mecanum Wheels. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*: Vol. 15209 LNAI. [https://doi.org/10.1007/978-981-96-0789-1\\_16](https://doi.org/10.1007/978-981-96-0789-1_16)
- Luo, Z., Wang, J., Ju, B., Li, C., & Hu, C. (2025). Design of a Trajectory-Tracking Controller for OMRs Based on Minimizing Tire Wear. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial*

Intelligence and Lecture Notes in Bioinformatics): Vol. 15201 LNAI. [https://doi.org/10.1007/978-981-96-0771-6\\_5](https://doi.org/10.1007/978-981-96-0771-6_5)

- Muktafin, E. H., Kusrini, K., & Luthfi, E. T. (2021). Analisis Sistem Kendali Robot USMAN untuk Sterilisasi Lantai Masjid dengan Algoritma Proportional Integral Derivative. *Jurnal Eksplora Informatika*, 10(2), 80–91. <https://doi.org/10.30864/eksplora.v10i2.468>
- Nakagawa, Y., Igo, N., & Hoshino, K. (2025). Controlling a Mecanum-Wheeled Robot with Multiple Swivel Axes Controlled by Three Commands. *Sensors*, 25(3). <https://doi.org/10.3390/s25030709>
- Purwanto, A., Suryanto, T., & Wijaya, R. (2020). Implementasi kontrol PID pada robot omnidirectional dengan roda mecanum. *Jurnal Teknik Elektro dan Robotika*, 12(2), 45–56.
- Sachan, S., & Pathak, P. M. (2025). Addressing unpredictable movements of dynamic obstacles with deep reinforcement learning to ensure safe navigation for omni-wheeled mobile robot. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 239(4), 1267–1293. <https://doi.org/10.1177/0954406224128115>
- Santoso, B. & Haryanto, D. (2019). Analisis kinerja kontrol Fuzzy-PID untuk robot beroda mecanum. *Jurnal Ilmu Komputer dan Robotika*, 7(1), 78–89.
- Selvam, R., Raghavan, S., Gokul, N., Raaj, K. V. K., & Damoodaran, S. (2025). Finite element analysis and functional performance of mecanum wheel. In *Recent Innovations in Sciences and Humanities: Highlights the state-of-art developments and innovations which impacts science and engineering*. <https://doi.org/10.1201/9781003606611-30>
- Shin, J., & Jung, H. (2022). UWB/GPS Sensor Fusion Using Kalman Filter for Outdoor Autonomous Robot. *International Conference on Control, Automation and Systems*, 2022-Novem, 448–451. <https://doi.org/10.23919/ICCAS55662.2022.10003948>
- Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). *Robotics: Modelling, Planning and Control*. Springer.
- Suchanek, G., & Ciesielka, W. (2018). Design and experimental research of a quadrocopter flying robot. *E3S Web of Conferences*, 46. <https://doi.org/10.1051/e3sconf/20184600012>
- Taufiqurrahman, I., Ulus Rahayu, A., Permana, M. Y., Studi, P., & Elektro, T. (2024). E-JOINT (Electronica and Electrical Journal of Innovation Technology) Desain Kendali Kecepatan Motor DC Pada Mobile Robot Mecanum Wheels (Vol. 05, Issue 2).
- Wada, M. (2022). Cooperative control of multiple modular mobile systems with active-caster omnidirectional drive mechanisms. *Communications in Information and Systems*, 22(4), 527–547. <https://doi.org/10.4310/CIS.2022.v22.n4.a5>
- Wahyu Kurniawan, G., Setyawan, N., Azizul Hakim, E., Elektro, T., & Muhammadiyah Malang, U. (2022). PID Trajectory Tracking Control 4 Omni-Wheel Robot.
- Wen, S., Su, Y., & Qu, H. (2025). Design of a Cascade Controller of Trajectory Tracking for Omnidirectional AGV Driven by Mecanum Wheels | Mecanum 轮全向 AGV 轨迹跟踪级联控制器设计. *Huanan Ligong Daxue Xuebao/Journal of South China University of Technology (Natural Science)*, 53(1), 49–61. <https://doi.org/10.12141/j.issn.1000-565X.240207>
- Yin, R., Wu, Z., & Feng, L. (2025). Pulse Width Modulation Control of Mecanum-Wheeled Mobile Robot With Random Noise. *IEEE Access*, 13, 13044–13051. <https://doi.org/10.1109/ACCESS.2025.3530405>
- Zhao, J., Tian, Z., Zhao, Z., Yang, X., Zhao, L., Jiang, Z., & Liu, H. (2025). Constraint-integrated inverse kinematics method for dual-arm motion. *Acta Astronautica*, 228, 755–768. <https://doi.org/10.1016/j.actaastro.2024.12.048>