

Pengendalian Konvergensi Eksponensial untuk *Omnidirectional Mobile Robot* dengan Empat Roda

Muhammad Jodi Pamenang¹, Budhy Setiawan², Indrazno Siradjuddin³

^{1,2,3}Jurusan Teknik Elektro, Politeknik Negeri Malang, Indonesia

Informasi Artikel

Riwayat Artikel

Diterima 5 Maret 2020

Direvisi 18 April 2020

Diterbitkan 28 April 2020

Kata kunci:

Robot Bergerak
Omnidirectional
Model Kinematika
Kontrol Posisi

Keywords:

Mobile Robot
Omnidirectional
Kinematics Model
Position Control

ABSTRAK

Tujuan mendasar dari kontrol gerak *mobile robot* adalah untuk mengarahkan robot ke posisi yang diberikan secara acak pada ruang 2D. *Mobile robot* dengan roda *omni* memiliki sifat *holonomic* di mana memiliki keunggulan kelincahan dan permasalahan pengendalian gerak hanya pada sisi aktuator, sedangkan *mobile robot* dengan roda konvensional, memiliki permasalahan tambahan pengendalian gerak dalam ruang area operasional robot. Karenanya, robot *omni* lebih gesit untuk bergerak dalam konfigurasi ruang area kerja apa pun. Makalah ini menyajikan model kontrol konvergensi eksponensial berbasis model untuk *mobile robot omnidirectional* roda empat. Kontrol yang diusulkan menjamin penurunan kesalahan secara eksponensial dari gerakan robot ke setiap posisi robot yang diinginkan. Pembahasan meliputi model kinematik dan kontrol dari robot bergerak *omnidirectional* roda empat dan eksperimen simulasi yang telah dilakukan untuk memverifikasi kinerja kontrol yang meliputi lintasan robot 2D, serta nilai *error* atau kesalahan pada kontrol robot. Hasil dari eksperimen simulasi menunjukkan keefektifan kontrol yang diusulkan. *Mobile robot* telah bergerak ke posisi yang diinginkan pada garis lurus dengan tujuan robot yang akurat dan nilai *error* atau kesalahan yang didapat ialah $|0.02735|$ serta grafik *error* telah menurun secara eksponensial.

ABSTRACT

The fundamental objective of a mobile robot motion control is to navigate the robot to any given arbitrary posture in which robot 2D location and its heading are concerned. Mobile robots with omni wheels have a holonomic properties the advantage is of agility and motion control problems only on the actuator, while mobile robots with conventional wheels, have a problem of motion control the robot in task space. Therefore, the omni-wheeled mobile robots are more agile to move in any task space configuration. This paper presents a model based exponential convergence control law for a four-wheeled omnidirectional mobile robot. The proposed control law guarantees an exponential error decay of mobile robot motion to any given desired robot posture. The kinematic model and the control law of a four-wheeled omnidirectional mobile robot are discussed. Simulation experiments have been conducted to verify the control law performances which include the 2D robot trajectory, the error signals, and the robot control signals. Results from simulation experiments show the effectiveness of the proposed control law. Mobile robot has moved to the desired position in a straight line with the aim of the robot that is accurate and the error or error obtained is $|0.02735|$ and the error graph has decreased exponentially.

Penulis Korespondensi:

Muhammad Jodi Pamenang,
Jurusan Teknik Elektro,
Politeknik Negeri Malang,
Jl. Sukarno Hatta No. 9, Malang, Jawa Timur, Indonesia.
Email: m.jodi.p@polinema.ac.id

1. PENDAHULUAN

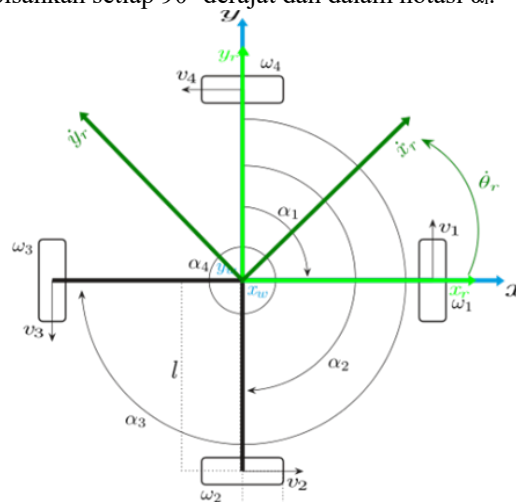
Teknologi Robotika telah menjadi dampak besar dalam kehidupan manusia. Robot yang semakin berkembang dengan banyak variasi dibuat untuk menghemat sumber energi dan efisiensi waktu [1][2][3]. Robot juga dapat dikatakan mampu mencapai berbagai arah tanpa batas seperti beralih ke arah yang diinginkan [3]. Robot konvensional seperti *Differential steering* dan *Ackerman steering* yang memiliki mobilitas rendah, karena hanya dapat bergerak kembali, maju dan berbelok, namun tidak dapat bergerak ke segala arah atau *non-holonomic* [4][5][6]. Oleh karena itu, penulis mengembangkan sebuah robot yang dapat bergerak ke segala arah di bidang $x - y$ Cartesian tanpa harus berbalik dan jenis robot ini disebut. robot *omni-directional* atau *omni wheel robot* yang memiliki mobilitas tinggi [7]. Saat ini masih ada dua jenis desain robot beroda *omni*, yaitu tiga roda [8][9] dan empat roda [10][11][12]. Kinematika adalah studi yang paling mendasar tentang bagaimana sistem mekanis berperilaku [13]. Proses untuk menentukan gerakan robot dimulai dengan proses menggambarkan kontribusi masing-masing roda sehingga dapat berjalan ke posisi yang diinginkan.

Dalam makalah ini, robot *omni* disimulasikan untuk bergerak secara sedirinya atau otomatis sesuai lintasan yang diberikan. Hal ini termasuk bagaimana untuk mendapatkan nilai Rpm pada masing-masing roda *omni-directional* dari penggunaan kinematika robot dan setiap *omni wheel* dapat dikontrol secara independen. Tulisan ini disusun sebagai berikut. Pada bagian II tentang konfigurasi mekanik empat roda *omni wheel mobile robot*. Bagian III, model kinematik dari model robot tersebut. Bagian IV menyajikan simulasi hasil kinematik robot yang mengubah nilai obyektif. Bagian V adalah sebuah kesimpulan..

2. Metode Penelitian

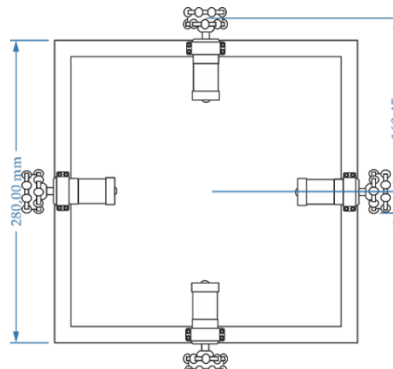
2.1. Konfigurasi geometris robot omni

Dalam membuat model kinematika penting untuk membuat skema gambar untuk konfigurasi geometris dari objek atau robot. Pada gambar 1 menyajikan konfigurasi robot beroda empat yang menggunakan *omni wheel*, serta semua sumbu, arah roda bergerak dan posisi robot yang relevan dengan sistem matematika. Konfigurasi roda tiga memiliki segmen gandar yang dipisahkan setiap 120° derajat dalam notasi α_i , sedangkan konfigurasi empat roda juga memiliki segmen gandar yang dipisahkan setiap 90° derajat dan dalam notasi α_i .



Gambar 1. Konfigurasi geometris dari empat roda *omni-directional robot*

Desain mekanis dan dimensi dari *omni robot* ditunjukkan pada gambar 2.



Gambar 2. Desain empat roda *omni-directional*

Parameter *omni robot* dan notasi yang digunakan sepanjang artikel ini ditunjukkan pada tabel 1.

Tabel 1. Spesifikasi robot *omni*

No	Simbol	Keterangan
1	x, y, θ	posisi robot x, y di unit meteran dan θ sudut hadap depan robot yang ditentukan
2	$\dot{x}, \dot{y}, \dot{\theta}$	tujuan posisi dari robot
3	$\alpha_1, \alpha_2, \alpha_3, \alpha_4$	sudut yang terbentuk antara roda dan terhadap sumbu y
4	$\omega_1, \omega_2, \omega_3, \omega_4$	kecepatan putar roda
5	v_1, v_2, v_3, v_4	kecepatan laju linier roda
6	L	jarak antara tengah roda dengan pusat robot
7	R	jari-jari roda
8	R_θ	rotasi Matrix
9	Ξ	posisi awal robot
10	ξ	posisi tujuan robot
11	Λ	penguatan

2.2. Geometris Kinematika *Omni-directional Robot* dan Model Kontrol

Kinematik pada robot sangat penting karena diperlukan untuk menentukan kecepatan dan arah robot tersebut. Berdasarkan sifat dari jenis *omni robot*, termasuk dalam kategori robot *holonomic*, robot ini dapat mengubah arah dan menerjemahkan ke kanan dan kiri atau berbagai arah tanpa harus mengubah orientasi robot itu sendiri dengan cara menggunakan kinematik terbalik (*invers kinematics*). Sebelum menuju ke kinematik terbalik kita harus membuat *Jacobian* dari robot (J_R) harus dihitung dengan menggunakan persamaan di bawah ini:

$$J_R = r \begin{bmatrix} -c(\alpha_1) & -c(\alpha_2) & -c(\alpha_3) & -c(\alpha_4) \\ s(\alpha_1) & s(\alpha_2) & s(\alpha_3) & s(\alpha_4) \\ \frac{1}{l} & \frac{1}{l} & \frac{1}{l} & \frac{1}{l} \end{bmatrix} \quad (1)$$

Setelah mendapatkan persamaan *Jacobian* dari *omni* roda empat (1), kinematika maju (*forward kinematics*) harus ditemukan. Kinematika maju adalah fungsi yang bertujuan untuk menemukan nilai posisi $[x, y, \theta]$ dan dapat disebut juga ξ dan hasil kinematika maju ditampilkan sebagai berikut:

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta} \end{bmatrix} = r \begin{bmatrix} -c(\alpha_1) & -c(\alpha_2) & -c(\alpha_3) & -c(\alpha_4) \\ s(\alpha_1) & s(\alpha_2) & s(\alpha_3) & s(\alpha_4) \\ \frac{1}{l} & \frac{1}{l} & \frac{1}{l} & \frac{1}{l} \end{bmatrix} \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \\ \omega_4(t) \end{bmatrix} \quad (2)$$

$$= r \begin{bmatrix} 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ \frac{1}{l} & \frac{1}{l} & \frac{1}{l} & \frac{1}{l} \end{bmatrix} \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \\ \omega_4(t) \end{bmatrix} \quad (3)$$

$$\dot{\xi}(t) = J_R \omega(t) \quad (4)$$

Rotasi matriks adalah matriks yang digunakan untuk merubah *local frame* menjadi *global frame*. Dalam hal ini *omni robot* berotasi yang dilakukan pada sumbu z (R_θ) sehingga memerlukan persamaan rotasi seperti di bawah ini (5):

$$R(\theta) = \begin{bmatrix} \cos(\theta(t)) & -\sin(\theta(t)) & 0 \\ \sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Dengan mendapatkan persamaan rotasi matriks, kita dapat mengalikan persamaan (R_θ) dengan *Jacobian robot* untuk mengembalikan posisi robot saat bergerak atau ketika berputar untuk kembali ke posisi awal (*global frame*) lagi setelah menuju ke posisi tujuan. Jadi menyajikan sebuah persamaan yang disebut *Jacobian world* ditampilkan dalam persamaan (9):

$$\dot{\xi}(t) = R(\theta) J_R \omega(t) \quad (6)$$

$$= r \begin{bmatrix} c(\theta(t)) & -s(\theta(t)) & 0 \\ s(\theta(t)) & c(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ \frac{1}{l} & \frac{1}{l} & \frac{1}{l} & \frac{1}{l} \end{bmatrix} \omega(t) \quad (7)$$

$$= r \begin{bmatrix} -s(\theta(t)) & 0 & c(\theta(t)) & s(\theta(t)) & c(\theta(t)) \\ c(\theta(t)) & s(\theta(t)) & c(\theta(t)) & -s(\theta(t)) \\ \frac{1}{l} & \frac{1}{l} & \frac{1}{l} & \frac{1}{l} \end{bmatrix} \omega(t) \quad (8)$$

$$\dot{\xi}(t) = J_R^W \omega(t) \quad (9)$$

Hasil dari perkalian matriks rotasi dengan *Jacobian robot* menghasilkan *Jacobian world* (J_R^W). Setelah mendapatkan J_w kita dapat menemukan nilai dari masing-masing motor pada robot *omni*, dengan menggunakan kinematika terbalik dari persamaan J_R^{-1} (10) dan $\dot{\xi}$ yang dapat menghasilkan nilai rpm pada masing-masing motor DC robot dan di bawah ini adalah persamaan dari kinematika terbalik:

$$J_R^{-1}(t) = r \frac{1}{4} \begin{bmatrix} 0 & 2 & l \\ 2 & 0 & l \\ 0 & -2 & l \\ -2 & 0 & l \end{bmatrix} \tag{10}$$

$$\omega(t) = J_R^{-1}(t)\dot{\xi}(t) \tag{11}$$

Untuk mengetahui apakah robot pergi ke titik tujuan dengan benar, membutuhkan selisih atau *Error* dari perbedaan dalam nilai yang dimaksudkan dari titik awal robot. Ini mendapatkan *Error* kesalahan untuk dapat mengoreksi apakah robot telah mencapai posisi yang diinginkan sehingga dapat dihitung dengan persamaan di bawah ini.

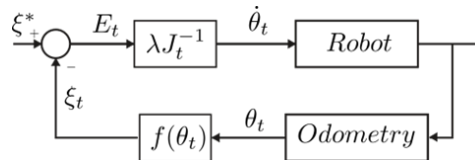
$$E = \xi(t) - \dot{\xi}(t) \tag{12}$$

$$\dot{E} = \xi(t) \tag{13}$$

dari $\xi^*(t) = 0$. Dengan kesalahan eksponensial menurun, $\dot{E} = \lambda E$, kemudian.

$$\omega(t) = \lambda J_R^{-1}(t)E(t) \tag{14}$$

Gambar 3 menunjukkan diagram blok kontrol posisi sehingga robot dapat dikoreksi apakah robot ini sesuai dengan posisi yang diinginkan atau belum mencapai posisinya.



Gambar 3. Blok diagram kontrol posisi untuk *omni robot*

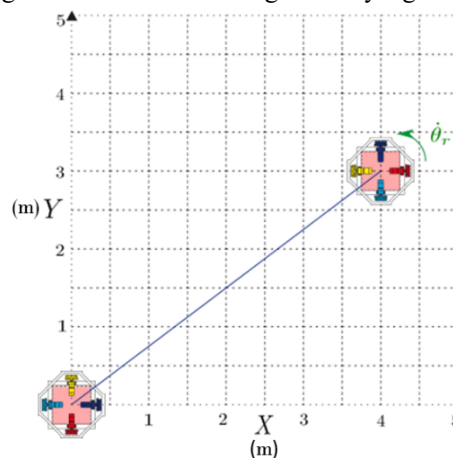
Dari gambar 3 blok diagram dan persamaan (14) di atas dapat menjelaskan sifat kontroler yang dirancang adalah berbentuk eksponensial sesuai dengan perubahan kecepatan berdasarkan jarak tujuan, yaitu robot semakin dekat dengan target, maka kecepatan dari robot akan semakin rendah atau mendekati nilai minimal dari kecepatannya.

3. Hasil dan Pembahasan

Bagian ini menyajikan simulasi kinematika sebuah robot *omni-directional*, yaitu simulasi dari empat roda *omni* kinematika, apakah hasil dari kinematika bekerja dengan baik atau belum bekerja. Simulasi ini menggunakan pemrograman Python 2,7. Simulasi dilakukan dengan referensi:

- $\dot{x} = 4.0$; $\dot{y} = 3.0$; $\dot{\theta} = 90.0^\circ$
- $\dot{x} = 2.0$; $\dot{y} = 0.0$; $\dot{\theta} = 0.0^\circ$
- $\dot{x} = 0.0$; $\dot{y} = 2.0$; $\dot{\theta} = 0.0^\circ$
- $\dot{x} = 2.0$; $\dot{y} = 2.0$; $\dot{\theta} = 45.0^\circ$

Pada gambar 4 adalah contoh gerakan robot dengan nilai tujuan $[\dot{x} ; \dot{y} ; \dot{\theta}] = [4.0; 3.0; 90^\circ]$ dengan ilustrasi gambar 4 apakah simulasi juga menghasilkan arah robot dengan nilai yang sama.



Gambar 4. Area kerja dan koordinat robot

Dalam simulasi ini, nilai rotasi robot atau θ harus dikalikan dengan $\frac{\pi}{180}$ karena Python bekerja di radian. Hasilnya dapat dilihat dalam persamaan (15).

$$\dot{\theta}(t) = \theta * \frac{\pi}{180} \tag{15}$$

Dari persamaan (14) kita bisa melakukan simulasi dengan memasukkan nilai $l = 0,16$; $r = 0,2$; $\lambda = 5$ dan menghasilkan nilai seperti ditunjukkan di bawah ini:

$$\lambda = \begin{bmatrix} 5. & 0 & 0 & 0 \\ 0 & 5. & 0 & 0 \\ 0 & 0 & 5. & 0 \\ 0 & 0 & 0 & 5. \end{bmatrix} \tag{16}$$

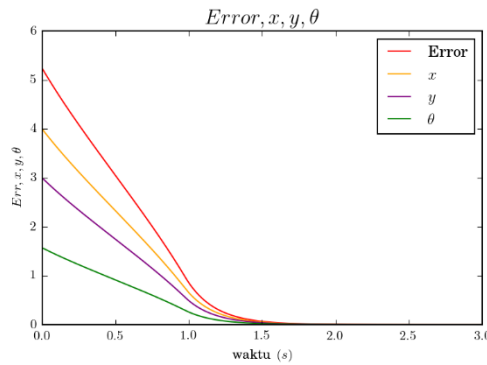
$$J_R^{-1}(t) = \begin{bmatrix} 0 & 25. & 2. \\ 25. & 0 & 2. \\ 0 & -25. & 2. \\ -25. & 0 & 2. \end{bmatrix} \tag{17}$$

$$\dot{\xi}(t) = \begin{bmatrix} 4. \\ 3. \\ 1.57 \end{bmatrix} \tag{18}$$

Setelah menyelesaikan perhitungan persamaan (14) hasil rpm untuk setiap roda *omni* dengan eksperimen maksimum Rpm ialah 100 Rpm/s dan nilai masing-masing roda (ω_i) adalah:

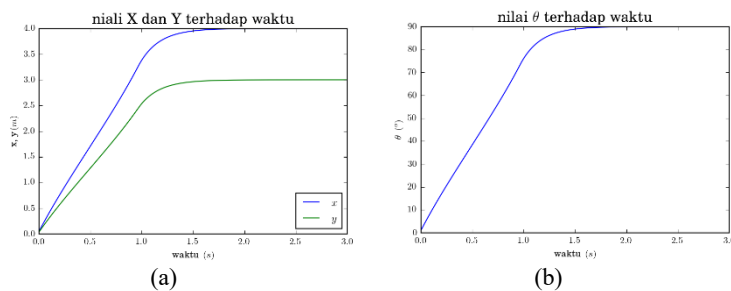
$$\omega_1(t) = 75.76 ; \omega_2(t) = 100.0 ; \omega_3(t) = -69.66 ; \omega_4(t) = -93.99$$

Pada Gambar 5 menyajikan hasil dari seluruh kesalahan dari empat roda *omni* robot mendapatkan nilai kesalahan kecil. Namun, untuk penyelesaian robot untuk mencapai pengaturan waktu *omni* robot empat roda adalah sekitar 1.6 (s). Waktu penyelesaian dari titik awal ke waktu yang diperlukan adalah waktu kesalahan sebelumnya seperti yang ditunjukkan pada gambar 5.



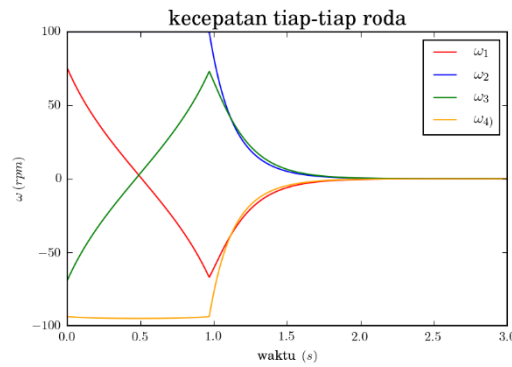
Gambar 5. Kesalahan parameter posisi robot terhadap waktu

gambar 6 memberikan hasil waktu dari nilai posisi awal ke posisi yang diinginkan.



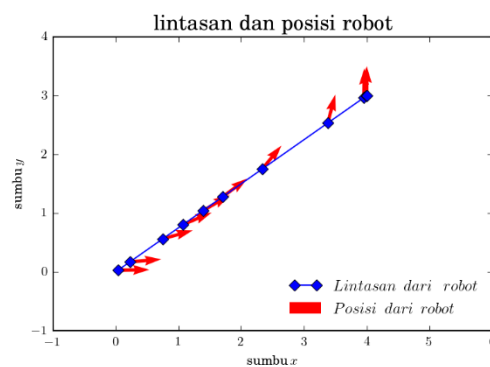
Gambar 6. (a) Respon dari sumbu x dan y terhadap waktu (b) Nilai θ terhadap waktu

Pada Gambar 7 menjelaskan bahwa hasil setiap nilai *omega* adalah sama dengan nilai output matematis sebelumnya. Sementara nilai masing-masing motor DC dalam *omni robot* menurun menjadi 0 ketika robot telah mencapai tujuannya.



Gambar 7. Kecepatan dari robot terhadap waktu

Gambar 8 mengungkapkan adalah lintasan hasil robot yang diberikan nilai objektif. Hasil mendapatkan hasil yang sesuai dengan nilai yang dimaksud.



Gambar 8. Posisi robot pada referensi [4, 3, 90°]

Di sisi lain, hasil simulasi, pada gambar 9 nilai kesalahan akhir dari kedua robot cukup kecil. Empat-roda robot mencapai mendapatkan nilai kesalahan akhir | 0,027 |.

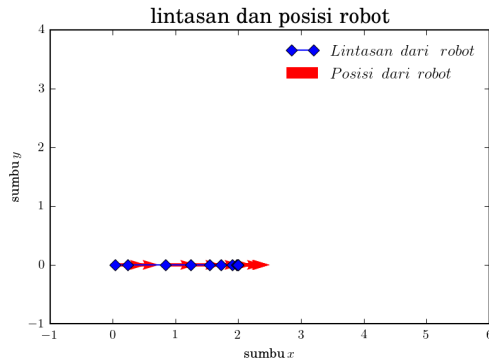
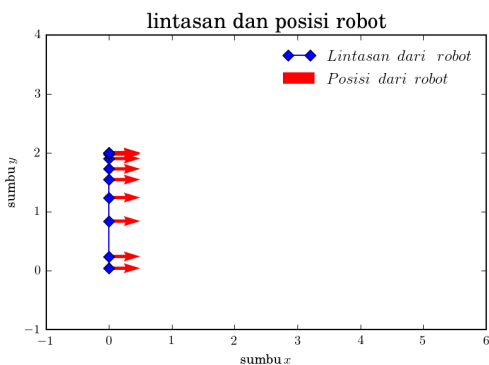
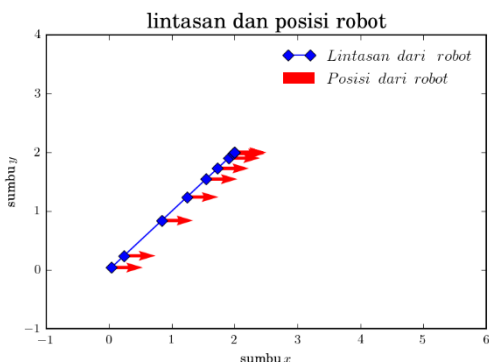
```
Titik Tujuan : Matrix([[4.00000000000000], [3.00000000000000],
[1.57079632679490]])
Posisi Aktual : Matrix([[3.97893938361203], [2.98420453770902],
[1.56252584207933]])
Error : 0.0275943311036407
```

Gambar 9. Pemantauan nilai kesalahan terakhir pada referensi [4, 3, 90°]

Tabel 2. Hasil *error* pada percobaan simulasi

No	Titik Tujuan (x; y; θ)	Posisi Aktual (x; y; θ)	<i>Error</i>
1	4; 3; 90	3.97; 2.98; 90	0.0275
2	2; 0; 0	2; 0; 0	0.027
3	0; 2; 0	0; 2; 0	0.027
4	2; 2; 45	2; 2; 0	0.0285
<i>Error Rata-rata</i>			0.02735

Berdasarkan hasil simulasi percobaan pertama menunjukkan hasil yang cukup baik dalam mencapai hasil jalan yang diinginkan. Nilai simulasi kemudian dapat dilihat pada gambar di bawah ini:

Gambar 10. Posisi robot pada referensi $[2, 0, 0^\circ]$ Gambar 11. Posisi robot di referensi $[0, 2, 0^\circ]$ Gambar 12. Posisi robot pada referensi $[2, 2, 45^\circ]$

hasil simulasi berikutnya, diwakili dalam gambar 10, 11 dan 12. Hasil akhir tidak jauh berbeda dari hasil simulasi sebelumnya karena kedua robot dapat diarahkan ke arah yang dituju.

4. KESIMPULAN

Dalam makalah ini, hasil simulasi dari kinematik empat-roda *omni-directional robot* mencapai hasil simulasi yang cukup baik. Karena nilai dari masing-masing motor DC dapat diberikan ke pada robot sehingga robot dapat bergerak sesuai dengan jalur dan berhenti di tujuan yang ditentukan, Serta nilai kesalahan atau *error* akhir mendapatkan nilai yang cukup kecil yaitu bernilai $|0.02735|$. Oleh karena itu kinematik dapat diterapkan secara signifikan.

5. UCAPAN TERIMA KASIH

Para penulis ingin mengucapkan terima kasih kepada pengulas atas komentar mereka yang berharga dalam meningkatkan kualitas pekerjaan ini.

6. DAFTAR PUSTAKA

- [1] Hashemi, Ehsan, Maani Ghaffari Jadidi, and Omid Bakhshandeh Babarsad. (2009). "Trajectory planning optimization with dynamic modeling of four wheeled omni-directional mobile robots." 2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation-(CIRA). IEEE.
- [2] Moreira, António Paulo, Paulo J. Costa, and André Scolari Conceição. (2006). "Model identification of a four wheeled omni-directional mobile robot." *CONTROL* 2006.
- [3] Oliveira, Hélder P., et al. (2009). "Modeling and assessing of omni-directional robots with three and four wheels." *Contemporary Robotics: Challenges and Solutions*.
- [4] Fahmizal, Fahmizal, Dhiya Uddin Rijalussalam, and Afrizal Mayub. (2019). "Trajectory Tracking pada Robot Omni dengan Metode Odometry." *Jurnal Nasional Teknik Elektro dan Teknologi Informasi* 8.1: 35-44.
- [5] M. West, H. Asada. (1992). "Design of a holonomic omnidirectional vehicle," in *Proc. IEEE International Conf. on Robotics and Automation*, pp. 97–103
- [6] Correia, Mariane Dourado, André Gustavo, and Scolari Conceição. (2012). "Modeling of a three wheeled omnidirectional robot including friction models." *IFAC Proceedings Volumes* 45.22: 7-12.
- [7] Barua, Ranjit, Sajal Mandal, and Samiran Mandal. (2015). "Motion analysis of a mobile robot with three omni-directional wheels." *International Journal of Innovative Science, Engineering and Technology* 2.11: 644-648.
- [8] Lee, Ming-Han, and Tzoo-Hseng S. Li. (2015). "Kinematics, dynamics and control design of 4WIS4WID mobile robots." *The Journal of Engineering* 2015.1: 6-16.
- [9] Krinkin, Kirill, Elena Stotskaya, and Yury Stotskiy. (2015). "Design and implementation Raspberry Pi-based omni-wheel mobile robot." 2015 Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT). IEEE.
- [10] Mariappan, Muralindran, et al. (2014). "Simultaneous rotation and translation movement for four omnidirectional wheels holonomic mobile robot." 2014 IEEE International Symposium on Robotics and Manufacturing Automation (ROMA). IEEE.
- [11] Şahin, Osman Nuri, Onur Çelik, and Mehmet İsmet Can Dede. (2017). "Fault-tolerance experiments with a kinematically redundant holonomic mobile robot." *International Workshop on Computational Kinematics*. Springer, Cham.
- [12] Huang, L., et al. (2004). "Design and analysis of a four-wheel omnidirectional mobile robot." 2nd International Conference of Autonomous Robots and Agents.
- [13] T.K. Nagy, R. D'Andrea, P. Ganguly. (2004). "Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle," *Robotics and Autonomous Systems*, vol. 47, no. 1, pp. 47–64.