

Collision Avoidance pada Mobile Robot Menggunakan Kontrol Kinematik

Muhammad Fattah Al Fattika¹, Totok Winarno², Indrazno Siradjuddin³

[Submission: 26-08-2021, Accepted: 11-10-2021]

Abstract—The robot function that frequently found is the navigation function, where the robot has to move to the destination point. Therefore, a robot that can move flexibly in all directions based on linear cartesian coordinate x and y , and z in angular, and not hit the object around it has developed. To make the robot move to the destination point, a kinematic control is needed. A controlled movement allowed the user to give an order to the robot to move to the destination point. By knowing where the object or obstacle was before, the robot can be ordered to move to the point where the robot will not hit the obstacle.

Keywords — *Omnidirectional Robot, Kinematics Control, Holonomic Move, Obstacle Avoidance*

Intisari—Fungsi robot yang sering dijumpai adalah fungsi navigasi yang mana robot harus bergerak ke titik tujuan. Oleh karena itu, dikembangkan sebuah robot yang dapat bergerak secara fleksibel ke segala arah dengan gerak berdasarkan koordinat kartesian x dan y linear, serta z angular, serta tidak menabrak objek disekelilingnya. Untuk dapat bergerak ke titik tujuan sekaligus tanpa menabrak objek disekitar, diperlukan kontrol kinematik. Dengan mengetahui dimana posisi objek atau rintangan sebelumnya, robot dapat diperintahkan untuk bergerak ke titik dimana robot tidak akan menabrak rintangan.

Kata Kunci— *Omnidirectional Robot, Kinematics Control, Holonomic Move, Obstacle Avoidance*

I. PENDAHULUAN

Perkembangan teknologi robotika telah dimulai sejak lama. Masing-masing jenis robot memiliki tujuan serta fungsi yang khusus, namun pada umumnya fungsi utama dari sebuah robot adalah untuk membantu manusia dalam melakukan pekerjaan. Perkembangan tersebut dilandasi akan kebutuhan manusia untuk membantu pekerjaannya mulai dari yang sederhana seperti memindahkan barang, hingga yang berbahaya seperti terbang atau menyelam ke dalam lautan.

Untuk dapat menggerakkan robot ke titik tujuan tertentu maupun posisi tertentu, kontrol kinematik dibutuhkan pada sebuah robot. Kontrol kinematik digunakan agar robot dapat

bergerak ke titik yang dituju relatif terhadap titik acuannya. Dengan mengaplikasikan kontrol kinematik, *mobile robot* dapat bergerak sesuai dengan titik tujuan yang direncanakan sebelumnya. Pengujian dan desain kontrol kinematik disimulasikan dengan menggunakan bahasa pemrograman *Python*.

Pada penelitian ini, didapatkan hasil bahwa aplikasi kontrol kinematik pada *mobile robot* dapat mengendalikan pergerakan robot ke titik tertentu. Data yang dijadikan acuan sebagai bahan pertimbangan diantaranya *error* terhadap waktu dan *trajectory* robot, yang mana data tersebut disajikan berupa grafik. Gerakan robot yang direncanakan sebelumnya menjadikan robot dapat bergerak ke titik tujuan akhir tanpa menyentuh rintangan di area gerakannya. Hal ini menunjukkan bahwa kontrol kinematik yang didesain dapat bekerja sesuai harapan.

II. KAJIAN PUSTAKA

A. Transformasi Geometri Posisi Pada Bidang Dua Dimensi

Suatu objek yang berada pada bidang 2 dimensi setidaknya memiliki nilai yang merepresentasikan posisinya pada bidang 2 dimensi tersebut. Nilai tersebut dinamai nilai titik koordinat, yaitu nilai koordinat x dan y . Nilai koordinat tersebut relatif terhadap titik pusat sumbu acuannya (*frame*). Titik koordinat memiliki titik acuan yang digunakan sebagai referensi, yang dapat disebut sebagai *global frame*. Pada sebuah bidang dua dimensi, tidak menutup kemungkinan memiliki lebih dari satu *frame*. Namun, untuk *global frame* cukup satu pada bidang dua dimensi. *Global frame* dapat dinotasikan sebagai f_g dan *frame* yang terletak pada f_g dapat dinotasikan dengan f_i .

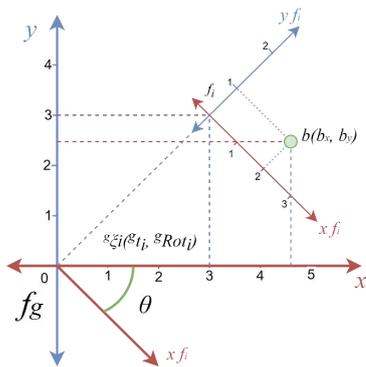
Pengukuran titik koordinat pada bidang dua dimensi harus memiliki titik acuan yang tetap. Pada kasus ini, *global frame* merupakan f_g . Ketika terdapat dua objek pada bidang 2 dimensi yang sama, pengukuran dapat dilakukan dengan mudah diukur dari titik f_i .

p-ISSN: 2356-0533; e-ISSN: 2355-9195

¹Program Studi D-IV Teknik Elektronika Jurusan Teknik Elektro Politeknik Negeri Malang, Jln. Soekarno Hatta no. 9 Malang, Jawa Timur, 65141, Indonesia; e-mail: mfattahalfattika@gmail.com

^{2,3} Program Studi D-IV Teknik Elektronika Jurusan Teknik Elektro Politeknik Negeri Malang, Jln. Soekarno Hatta no. 9 Malang, Jawa Timur, 65141, Indonesia; e-mail: totok.winarno@polinema.ac.id, indrazno@polinema.ac.id





Gambar 1 : Representasi f_b pada f_g

Namun, pengukuran dari titik f_i harus tetap berdasarkan f_g , sehingga dibutuhkan suatu persamaan untuk mengetahui posisi koordinat suatu objek pada bidang dua dimensi, yang pada kasus ini, dinotasikan dengan b pada f_g berdasarkan data dari f_i , begitu pula sebaliknya.

Pada gambar 1 terdapat beberapa simbol dan variabel yang dapat dimasukkan ke dalam persamaan untuk mencari posisi dari benda b . Benda b memiliki nilai koordinat posisi terhadap f_g dan f_i . $b^o[x_b^o, y_b^o]$ merupakan nilai titik koordinat objek b terhadap *frame global*. $b^i[x_b^i, y_b^i]$ merupakan nilai titik koordinat objek b terhadap f_i , dimana f_i juga memiliki nilai translasi dan rotasi terhadap *global frame*, yang dinotasikan secara berurutan yaitu t_i^o dan Rot_i^o , dimana dari nilai translasi dan rotasi f_i terhadap f_g dapat diperoleh nilai sudut θ yang dinotasikan sebagai $\xi_i^o[t_i^o, Rot_i^o]$, sehingga diperoleh persamaan berikut

$$b^o = \xi_i^o \cdot b^i \quad (1)$$

dengan ξ_i^o adalah $[t_i^o, Rot_i^o]$, sehingga dapat menjadi :

$$b^o = Rot_i^o \cdot b^i + t_i^o \quad (2)$$

$$\begin{bmatrix} x_b^o \\ y_b^o \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_b^i \\ y_b^i \end{bmatrix} + \begin{bmatrix} x_i^o \\ y_i^o \end{bmatrix} \quad (3)$$

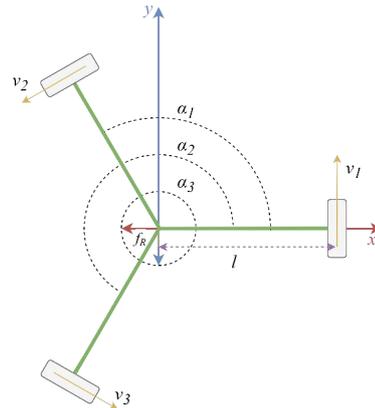
dengan menggunakan persamaan 2 maka dapat diketahui nilai b^o jika mengetahui nilai titik koordinat b^i . Oleh demikian, nilai titik koordinat b^i dapat diperoleh b^o , maka persamaannya dapat dirubah menjadi

$$b^i = Rot_i^{-1} \cdot (b^o - t_i^o) \quad (4)$$

$$\begin{bmatrix} x_b^i \\ y_b^i \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}^{-1} \cdot \left(\begin{bmatrix} x_b^o \\ y_b^o \end{bmatrix} - \begin{bmatrix} x_i^o \\ y_i^o \end{bmatrix} \right) \quad (5)$$

B. Transformasi Geometri Posisi Pada Bidang Dua Dimensi

Untuk mengetahui posisi robot terhadap frame, dibutuhkan perhitungan kinematik yang berdasarkan dari kecepatan putar masing-masing roda penggerak. Gambar 2 merupakan frame sebuah robot 3 roda dengan roda penggerak *omniwheels*. Nilai α_1, α_2 , dan α_3 memiliki perbedaan sudut sebesar 120° , dan nilai l merupakan jarak antara titik tengah robot ke titik tengah dari roda penggerak robot.



Gambar 2 : *Frame Robot*

Masing-masing roda memiliki kecepatan rotasi yang dinotasikan berurutan yaitu ω_1, ω_2 , dan ω_3 . Dari kecepatan roda, dapat dihitung kecepatan perpindahan robot terhadap *frame* yang dinyatakan dengan persamaan *forward kinematics* berikut :

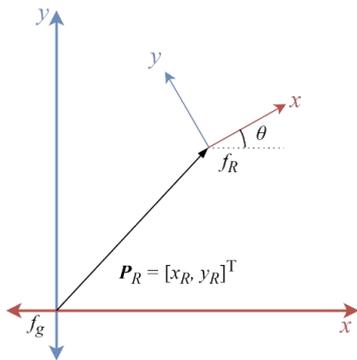
$$\dot{\xi}_R^R(t) = J_R^R(t) \cdot \omega(t) \quad (6)$$

$$\begin{bmatrix} \dot{x}_R^R \\ \dot{y}_R^R \\ \dot{\theta}_R^R \end{bmatrix} = \begin{bmatrix} \cos(\alpha_1)\cos(\alpha_2)\cos(\alpha_3) \\ \sin(\alpha_1)\sin(\alpha_2)\sin(\alpha_3) \\ \frac{1}{l} \quad \frac{1}{l} \quad \frac{1}{l} \end{bmatrix} \cdot \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (7)$$

dimana r adalah jari – jari roda penggerak robot, dan $\dot{\xi}_R^R$ adalah kecepatan perubahan posisi terhadap f_g .

Untuk mendapatkan nilai $\dot{\xi}_R^R$, dapat dilakukan dengan cara melakukan operasi perkalian antara $\dot{\xi}_R^R$ dengan matrik rotasi terhadap f_g , sehingga didapatkan grafik posisi robot terhadap *global frame* berikut :





Gambar3: Posisi robot terhadap *global frame*

Dari grafik tersebut, dapat dituliskan persamaan dari $\dot{\xi}_R$ yaitu sebagai berikut :

$$\dot{\xi}_R(t) = \text{Rot}_R^o(t) \cdot \dot{\xi}_R^R(t) \quad (8)$$

$$\dot{\xi}_R = J_R(t) \cdot \omega(t) \quad (9)$$

Dimana nilai θ diapatkan dari sudut yang terbentuk antara f_g dan f_r . Setelah didapatkan nilai kecepatan perubahan posisi robot terhadap peta, maka dapat diketahui posisi robot pada waktu ke t melalui persamaan berikut.

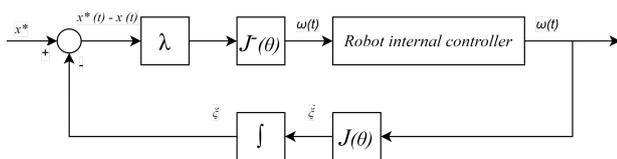
$$x_R(t) = x_r(t-1) + \dot{x}_R(t) \cdot \Delta t \quad (10)$$

$$y_R(t) = y_r(t-1) + \dot{y}_R(t) \cdot \Delta t \quad (11)$$

$$\theta_R(t) = \theta_r(t-1) + \dot{\theta}_R(t) \cdot \Delta t \quad (12)$$

III. METODE PENELITIAN

C. Desain Sistem Kendali



Gambar4 : Desain sistem kendali

Sistem kendali yang diaplikasikan bersifat reaktif terhadap *error*. Sistem kendali menerima masukan berupa titik tujuan robot yang berada pada *global frame*. *Error* didapatkan dari pembacaan posisi robot menggunakan *odometry* dan perhitungan kinematik posisi robot terhadap *global frame*. Keluaran dari sistem kendali merupakan kecepatan robot yang

nantinya dikalikan dengan matriks *Jacobian* dan diintegrasikan sehingga didapatkan posisi aktual robot.

D. Perancangan Forward Kinematics Beroda 2 Omniwheels

Kinematik dua roda *omniwheels* digunakan untuk melakukan pembacaan posisi robot terhadap *global frame*. Perancangan yang berbeda antara *forward* dan *inverse* dilakukan karena pada aplikasinya, sensor yang digunakan memiliki jumlah dan konfigurasi pemasangan yang berbeda. Untuk perhitungan *forward kinematics*, sensor yang digunakan yaitu *encoder* dengan jumlah 2 buah, sedangkan untuk perhitungan *inverse kinematics*,

Penggunaan yang berbeda antara perhitungan *forward kinematics* dan *inverse kinematics* dilakukan untuk mengatasi masalah selip yang sering terjadi pada roda penggerak. Dengan menggunakan *encoder* yang terpisah antara *encoder* pada motor penggerak dengan *encoder* pembacaan posisi robot, maka perubahan posisi hanya akan terjadi jika robot berpindah posisi dari satu tempat ke tempat lain, hal ini dapat memperkecil kemungkinan kesalahan pembacaan posisi robot terhadap lapangan. Namun terdapat kelemahan dari penggunaan dua *encoder* beroda *omniwheels* yaitu pada pembacaan arah hadap robot. Robot tidak dapat membaca gerak rotasi dengan baik, oleh karena itu, digunakan BNO055 untuk mendeteksi perubahan arah hadap robot.

Persamaan kinematik roda dua *omniwheels* memiliki beberapa nilai variabel yang didapatkan dari spesifikasi robot yang nantinya dimasukkan ke persamaan. Persamaan kinematik roda dua *omniwheel* hampir serupa dengan persamaan roda tiga *omniwheels* karena roda *encoder* dipasang serupa dengan susunan roda tiga *omniwheels*, hanya terdapat perbedaan pada jari-jari roda, jarak antara titik pusat roda dengan titik tengah robot, dan beberapa penyesuaian nilai derajat sudut antara arah hadap robot dengan arah putar roda normal (α_i) dan nilai derajat sudut antara arah hadap robot dengan titik pusat roda penggerak (γ_i).

Beberapa variabel yang dimasukkan ke dalam persamaan kinematik pembacaan posisi robot adalah :

- $\alpha_1 = -45^\circ$
- $\alpha_2 = -135^\circ$
- $\alpha_3 = 0^\circ$, karena hanya terdapat dua roda
- $l = 0.1$
- $r = 0.05 \text{ m}$



Dari beberapa nilai variabel tersebut, jika dimasukkan ke dalam Matriks *Jacobian* maka didapatkan matriks *Jacobian* sebagai berikut.

$$J_R(t) = 0.05 \begin{bmatrix} \cos(-45) & \cos(-135) & 0 \\ \sin(-45) & \sin(-135) & 0 \\ 1/0.1 & 1/0.1 & 0 \end{bmatrix} \quad (13)$$

Jika matriks *Jacobian* diatas disubstitusikan kedalam persamaan kinematik untuk mengetahui posisi robot, maka didapatkan persamaan sebagai berikut.

Dengan alasan untuk mengikuti standar bahwa bagian depan robot menghadap ke sumbu x, maka arah hadap perlu dirotasi, untuk itu matriks *Jacobian* perlu dikalikan dengan matriks rotasi, sehingga matriks *Jacobian* kinematik posisi robot terhadap *global frame* adalah sebagai berikut.

$$J_R(t) = 0.05 \begin{bmatrix} \cos(\theta(t)) & \cos(\theta(t)) & 0 \\ \sin(\theta(t)) & \sin(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(-45) & \cos(-135) & 0 \\ \sin(-45) & \sin(-135) & 0 \\ 1/0.1 & 1/0.1 & 0 \end{bmatrix} \quad (14)$$

Sehingga jika matriks *Jacobian* tersebut disubstitusikan ke dalam persamaan kinematik posisi robot terhadap *global frame* maka didapatkan persamaan berikut.

$$\dot{\xi}_R(t) = J_R(t)\omega(t) \quad (15)$$

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} = 0.05 \begin{bmatrix} \cos(\theta(t)) & \cos(\theta(t)) & 0 \\ \sin(\theta(t)) & \sin(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(-45) & \cos(-135) & 0 \\ \sin(-45) & \sin(-135) & 0 \\ 1/0.1 & 1/0.01 & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ 0 \end{bmatrix} \quad (16)$$

Dimana nilai ω_3 adalah nol karena hanya terdapat dua roda *omniwheels* untuk pembacaan posisi robot terhadap *global frame*. Hal ini menyebabkan pembacaan $\dot{\theta}_R$ tidak sesuai. Oleh karena itu, BNO055 digunakan untuk pembacaan arah.

E. Perancangan Inverse Kinematics Beroda 3 Omniwheels

Kontrol kinematik robot terhadap posisi tujuan menggunakan *Inverse Kinematics*. Robot yang bergerak kearah titik tujuan maka menyebabkan jarak posisi antara robot dengan tujuan semakin kecil. Pembacaan jarak antara robot dengan titik tujuan menggunakan perhitungan *Forward Kinematics*. Untuk dapat bergerak, robot memerlukan sebuah sistem kendali yang mampu mengendalikan kecepatan roda penggerak sehingga robot dapat bergerak ke titik tujuan dan memperkecil nilai jarak antara robot dengan titik tujuan hingga mendekati nol.

Perhitungan kinematik membutuhkan masukan dari spesifikasi robot. Berikut nilai masukan yang dimasukkan ke *Jacobian* persamaan kinematik roda tiga *omniwheels*.

- $\alpha_1 = 30^\circ$
- $\alpha_2 = 150^\circ$
- $\alpha_3 = -90^\circ$
- $l = 0.3 \text{ m}$
- $r = 0.075$

Dimana α_i adalah nilai sudut antara garis tengah titik robot atau x_r dan arah putar searah jarum jam masing masing roda atau x_{wi} .

Dengan nilai konstanta yang telah didapatkan, dapat dimasukkan kedalam matriks *Jacobian* sehingga didapatkan *Jacobian* persamaan kinematiknya yaitu sebagai berikut.

$$J_R(t) = 0.075 \begin{bmatrix} \cos(30)\cos(150)\cos(-90) \\ \sin(30)\sin(150)\sin(-90) \\ \frac{1}{0.3} & \frac{1}{0.3} & \frac{1}{0.3} \end{bmatrix} \quad (17)$$

Kemudian, untuk dapat dimasukkan ke dalam persamaan *inverse kinematik*, maka matriks *Jacobian* perlu di *inverse* sehingga didapatkan persamaan *inverse kinematics* sebagai berikut

$$\dot{\xi}_R(t) = J_R(t)\omega(t) \quad (18)$$

$$J_R^{-1}(t) \dot{\xi}_R(t) = J_R^{-1}(t)J_R(t)\omega(t) \quad (19)$$

$$\omega(t) = J_R^{-1}(t) \dot{\xi}_R(t) \quad (20)$$

$$\begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \end{bmatrix} = 0.075 \begin{bmatrix} \cos(30)\cos(150)\cos(-90) \\ \sin(30)\sin(150)\sin(-90) \\ \frac{1}{0.3} & \frac{1}{0.3} & \frac{1}{0.3} \end{bmatrix} \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} \quad (21)$$

Untuk mendapatkan nilai $\dot{\xi}_R(t)$ digunakan pendekatan nilai *error* jarak posisi antara robot dengan titik tujuan. Pendekatan dilakukan dengan melakukan operasi pengurangan antara posisi tujuan dengan posisi robot.

$$\mathbf{error} = \xi^*(t) - \xi(t) \quad (22)$$

$$\mathbf{error} = \begin{bmatrix} x^*(t) \\ y^*(t) \\ \theta^*(t) \end{bmatrix} - \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} \quad (23)$$

Karena robot yang bergerak menuju posisi tujuan memiliki kecepatan, sehingga untuk dapat dikalkulasi dengan posisi



tujuan, kecepatan robot perlu diturunkan terhadap waktu, sehingga didapatkan persamaan sebagai berikut.

$$\frac{dE(t)}{dt} = \xi^*(t) - \xi(t) \quad (24)$$

$$\dot{E}(t) = -\dot{\xi}(t) \quad (25)$$

Turunan dari posisi tujuan adalah nol, karena posisi tujuan bersifat tetap, tidak berpengaruh terhadap waktu.

Sistem kendali gerak robot dirancang untuk bekerja secara eksponensial, gerak robot akan semakin lambat jika semakin dekat dengan posisi tujuan. Persamaannya dapat dituliskan sebagai berikut.

$$E(t) = e^{-\lambda(t)} \quad (26)$$

$$\dot{E}(t) = -\lambda e^{-\lambda(t)} \quad (27)$$

$$\dot{E}(t) = -\lambda E(t) \quad (28)$$

Sehingga persamaan (3.11) dapat disubstitusikan ke persamaan (3.8)

$$\dot{\xi}(t) = \lambda E(t) \quad (29)$$

Jika persamaan (3.12) disubstitusikan ke persamaan (3.3) atau persamaan untuk mendapatkan kecepatan gerak motor, maka didapatkan persamaan berikut.

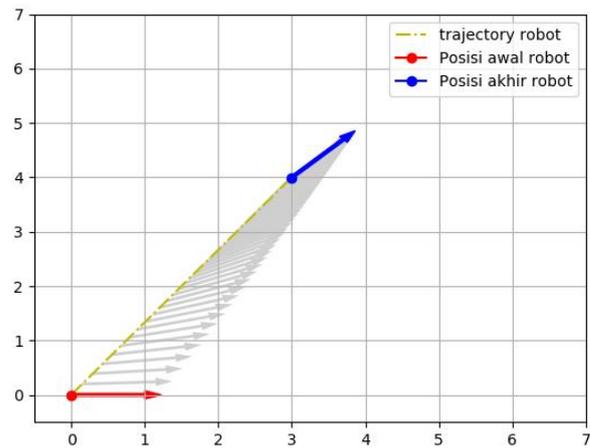
$$\omega(t) = J_R^{-1}(t) (\xi^*(t) - \xi(t)) \quad (30)$$

IV. HASIL DAN ANALISA

Hasil yang dijadikan pembahasan berupa grafik respon sistem yang didapatkan dari simulasi komputer dengan menggunakan bahasa pemrograman *Python*. Terdapat dua pengujian yaitu pengujian respon sistem dengan satu titik tujuan, dan pengujian respon sistem dengan beberapa titik tujuan hingga titik tujuan akhir.

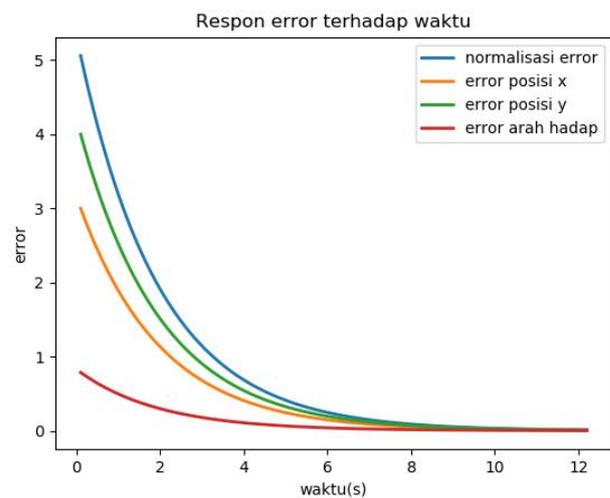
F. Hasil Simulasi Dengan Satu Titik Tujuan

Simulasi pergerakan robot ke satu titik tujuan dilakukan untuk memastikan bahwa robot dapat bergerak ke titik tujuan sesuai dengan perintah. Robot diperintahkan untuk bergerak ke titik tertentu terhadap *frame global*. Robot diperintahkan bergerak ke titik [3,4] dengan arah hadap akhir 45°.



Gambar 5 : Respon *trajectory* robot terhadap *frame global*

Berdasarkan respon yang didapat pada Gambar 5 didapatkan bahwa robot dapat bergerak ke titik tujuan dengan baik serta *trajectory* yang sangat efektif. Robot dapat bergerak langsung ke titik tujuan yang diharapkan, dan dengan arah hadap yang diinginkan. Kecepatan pergerakan robot dapat diatur dengan cara mengatur penguatannya (λ).



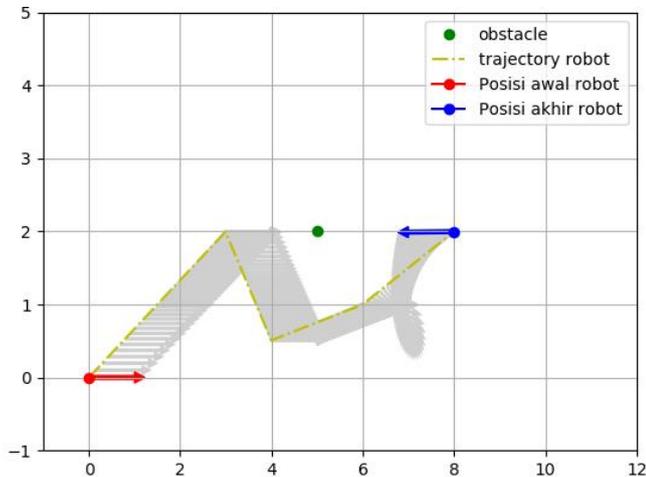
Gambar 6 : Respon error posisi terhadap waktu

Gambar 6 merupakan respon nilai *error* posisi terhadap titik tujuan. Didapatkan bahwa respon *error* yang didapatkan sesuai dengan sistem kendali yang diterapkan. Respon *error* didesain agar berbentuk menyerupai grafik eksponensial. Hal ini dapat diperoleh dengan desain sistem kendali yang memiliki sifat reaktif terhadap nilai *error*. Nilai *error* yang didapat memiliki nilai normalisasi yang mana jika nilai *error* yang didapat sudah mencapai nilai normalisasi, robot akan menganggap posisinya sudah berada pada titik tuuan yang diinginkan.



G. Hasil Simulasi Dengan Banyak Titik Tujuan

Pengujian dengan beberapa titik tujuan dilakukan untuk melihat respon sistem jika diberikan tidak hanya satu titik tujuan. Hal ini dilakukan untuk melihat apakah robot dapat bergerak tanpa menyentuh rintangan dengan cara diberikan masukkan berupa titik tujuan tertentu yang memungkinkan robot bergerak tanpa menyentuh rintangan.



Gambar 7: Trajectory robot dengan beberapa titik tujuan

Pada gambar 4.3, didapatkan robot dapat bergerak ke titik tujuan akhir [8,2] dengan arah hadap 180° terhadap arah hadap awalnya. Didapatkan bahwa robot dapat bergerak dengan baik serta memiliki *trajectory* yang efektif untuk bergerak ke titik tujuan akhir. Robot dapat bergerak ke titik tujuan akhir tanpa menyentuh rintangan dengan cara diberikan titik tujuan tertentu, yaitu titik tujuan 1 [2.5, 2.5], titik tujuan 2 [4, 0.5], titik tujuan tiga [6, 1], dan titik tujuan akhir [8, 2] dengan arah hadap 180°

V. PENUTUP

Dari hasil penelitian, didapatkan kesimpulan sebagai berikut :

1. Hasil simulasi dan hasil pengujian secara *real time* dari *trajectory* pergerakan robot, *error* posisi dan *error* normalisasi terhadap waktu, serta respon kecepatan motor terhadap waktu, robot mampu bergerak ke titik tujuan akhir yang diinginkan dengan cara memberikan robot titik tujuan tertentu agar robot tidak menyentuh rintangan ketika perjalanan ke titik tujuan akhir,
2. Robot dapat bergerak dengan akurasi titik tujuan dengan nilai normalisasi *error* senilai 0.25 meter. Waktu yang

ditempuh untuk robot mencapai titik tujuan akhir dipengaruhi oleh respon kecepatan putar roda robot, yang mana dapat dikontrol tinggi rendahnya dengan mengatur nilai penguatan (λ) pada kontrol kinematik.

3. Untuk mempercepat transisi robot dari satu titik tujuan ke titik tujuan selanjutnya hingga di titik tujuan akhir, nilai *error* normalisasi dapat diperbesar. Nilai *error* normalisasi yang digunakan pada penelitian ini adalah 0.35 meter.
4. Sistem kontrol yang reaktif terhadap nilai *error* dapat bekerja dengan baik pada robot,
5. Dengan kontrol kinematik, robot dapat mempertahankan posisi terakhirnya dengan cara bergerak kembali ke posisi terakhirnya jika dipindah ke posisi lain dengan paksa. Dengan catatan robot dipindah dengan kondisi roda *encoder* tetap menempel dengan lantai lapangan.

REFERENSI

- [1] I. Siradjuddin, *Kinematics and Control A Three Wheeled Omnidirectional Mobile Robot*, *Int. J. Electr. Electron. Eng.*, vol. 6, no. 12, pp. 1–6, 2019, doi: 10.14445/23488379/ijeee-v6i12p101.
- [2] R. T. Yunardi, D. Arifianto, F. Bachtiar, and J. I. Prananingrum, *Holonomic Implementation of Three Wheels Omnidirectional Mobile Robot using DC Motors*, *J. Robot. Control*, vol. 2, no. 2, 2021, doi: 10.18196/jrc.2254.
- [3] T. D. Viet, P. T. Doan, N. Hung, H. K. Kim, and S. B. Kim, *Tracking Control of a Three-Wheeled Omnidirectional Mobile Manipulator System With Disturbance and Friction*, *J. Mech. Sci. Technol.*, vol. 26, no. 7, pp. 2197–2211, 2012, doi: 10.1007/s12206-012-0541-1.
- [4] I. Siradjuddin, M. L. M. Faizin, G. Al Azhar, and A. Murdani, *Desain dan pemodelan kontrol kinematik pergerakan robot beroda dengan menggunakan 6 roda omni-wheels*, *J. ELTEK*, vol. 18, no. 1, p. 116, 2020, doi: 10.33795/eltek.v18i1.226.
- [5] G. Al Azhar, *Implementasi Kontrol Kinematik dan Algoritma Lokalisasi Robot pada Robot Sepakbola Beroda*. Malang: Politeknik Negeri Malang, 2019.
- [6] M. F. K. Awaludin, *Motion Planning Robot Dengan Kontrol Kinematik*. Malang: Politeknik Negeri Malang, 2020.
- [7] R. L. Putra, H. K. Safitri, and I. Siradjuddin, *Pemodelan dan Analisis Sistem Kontrol Kinematik*, *J. ELKOLIND*, vol. 2, no. 1, pp. 80–87, 2015.
- [8] M. M. H. Ma'arif, *Sistem Navigasi Pada Mobile Robot Menggunakan Sensor Kompas*. Surabaya: Institut Bisnis dan Informatika STIKOM Surabaya, 2017.
- [9] R. A. Rochmanto, *Implementasi Robot Three Omni-Directional*



Menggunakan Kontroler Pid Pada Robot Kontes Robot Abu Indonesia (KRAI), 2014.

- [10] M. Taufiqqurohman and N. F. Sari, *Odometry Method and Rotary Encoder for Wheeled Soccer Robot*, *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 407, no. 1, 2018, doi: 10.1088/1757-899X/407/1/012103.
- [11] J. Inthiam and C. Deelertpaiboon, *Self-localization and navigation of holonomic mobile robot using omni-directional wheel odometry*, *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, vol. 2015-Janua, pp. 1–5, 2015, doi: 10.1109/TENCON.2014.7022281.
- [12] M. J. C. Manullang, M. K. D. Hardhienata, and K. Priandana, *Kendali Robot Beroda Otonom dengan Inverse Kinematics*, *J. Ilmu Komput. dan Agri-Informatika*, vol. 7, no. 1, pp. 62–73, 2020, doi: 10.29244/jika.7.1.62-73.

