

Navigasi Pergerakan Robot Berdasarkan Rekam Data Sensor Odometry

Wildan Arif Maulana¹, Totok Winarno, ², Indrazno Siradjuddin³

e-mail: wildanarif1@gmail.com, totok.winarno@polinema.ac.id, indrazno@polinema.ac.id

^{1,2,3}Jurusan Teknik Elektro, Politeknik Negeri Malang, Jalan Soekarno Hatta No.9 Malang, Indonesia

Informasi Artikel

Riwayat Artikel

Diterima 20 November 2022

Direvisi 18 Januari 2023

Diterbitkan 31 Mei 2023

Kata kunci:

Omnidirectional
Tracking Trajectory
Navigasi
Odometry

Keywords:

Omnidirectional
Tracking Trajectory
Navigation
Odometry

Penulis Korespondensi:

Wildan Arif Maulana
Jurusan Teknik Elektro
Politeknik Negeri Malang
Jl. Jalan Soekarno Hatta No.9 Malang, Indonesia
Email: wildanarif1@gmail.com Nomor HP/WA aktif: +6289 687 040 076

ABSTRAK

Berkembangnya teknologi robotika yang sangat cepat dan pesat mengakibatkan peran robotika sangat berpengaruh dalam kehidupan manusia. Pada bidang industri, robot difungsikan salah satunya untuk memindahkan barang ke tempat yang sudah ditentukan. Pada penelitian *mobile robot* beroda ini menggunakan perekaman data menggunakan sensor *odometry*. Sensor *odometry* digunakan untuk memperkirakan perubahan posisi robot dari pergerakan aktuator yang digunakan. Menggunakan sensor rotary encoder yang digunakan untuk menghitung pergerakan *mobile robot* pada koordinat X dan Y pada proses perhitungan *odometry*. Untuk dapat bergerak ke titik yang sudah ditentukan menggunakan metode kontrol kinematik, dengan mengetahui posisi awal, memungkinkan robot bergerak menuju titik tujuan dan arah hadap robot dengan koordinat pergerakan yang diajarkan sebelumnya. Pada penelitian ini menggunakan pengujian secara *realtime* untuk parameter pengujiannya meliputi ketepatan titik tujuan, arah hadap akhir dan kecepatan mencapai tujuan. Dengan menggunakan metode kontrol *Forward* dan *Invers Kinematics*, robot dapat bergerak dari posisi awal menuju ke titik tujuan dengan tepat menggunakan *tracking trajectory* yang telah dibuat berdasarkan rekam data sensor *odometry* dengan rata-rata error posisi sebesar 0,083 meter dan error arah hadap sebesar $4,6^\circ$

ABSTRACT

The rapid development of robotics technology has resulted in the role of robotics being very influential in human life. In the industrial sector, one of the functions of robots is to move goods to a predetermined place. In this research, this wheeled mobile robot uses data using odometry sensors. The odometry sensor is used to estimate changes in the position of the robot from the movement actuator used. Using a rotary encoder sensor is used to calculate the movement of the robot moving at X and Y coordinates in the odometry calculation process. To be able to move to a predetermined point using the kinematic control method, knowing the initial position, allows the robot to move towards the destination point and the direction of the robot with the movement coordinates that were taught previously. This thesis uses real-time testing for the testing parameters including the accuracy of the destination point, the direction towards the end, and the speed of reaching the destination. By using the Forward and Inverse Kinematics control methods, the robot can move from the initial position to the destination point precisely using a tracking trajectory that has been made based on odometry sensor data records with an average position error of 0.083 meters and an error facing direction of 4.6°



1. PENDAHULUAN

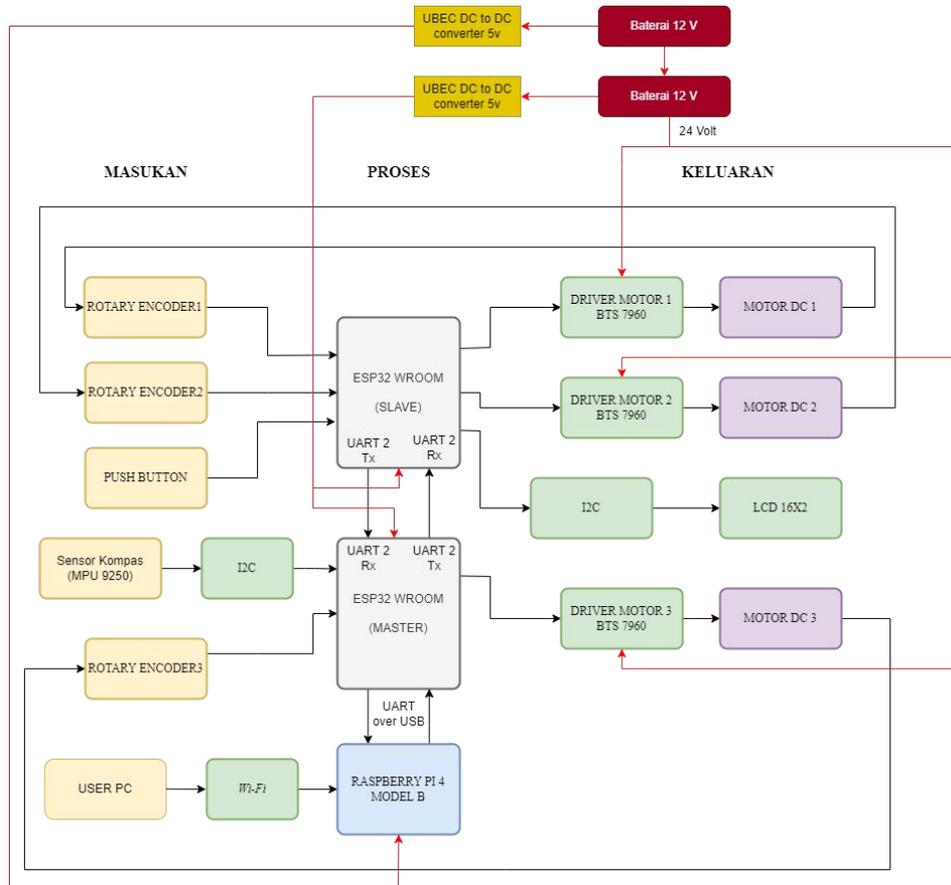
Berkembangnya teknologi robotika yang sangat cepat dan pesat mengakibatkan peran robotika sangat berpengaruh dalam kehidupan manusia. Pada bidang industri, robot difungsikan salah satunya untuk memindahkan barang ke tempat yang sudah ditentukan. Pada penelitian *mobile robot* beroda ini menggunakan perekaman data menggunakan sensor *odometry*. Sensor *odometry* digunakan untuk memperkirakan perubahan posisi robot dari pergerakan aktuator yang digunakan. Menggunakan sensor rotary encoder yang digunakan untuk menghitung pergerakan robot *omni* pada koordinat X dan Y pada proses perhitungan *odometry*. Untuk dapat bergerak ke titik yang sudah ditentukan menggunakan metode kontrol kinematik. Dengan mengetahui posisi awal, memungkinkan robot bergerak menuju titik tujuan dan arah hadap robot dengan koordinat pergerakan yang diajarkan sebelumnya.

Persamaan kinematik menggunakan sensor sebagai masukannya. Sensor yang digunakan yaitu *rotary encoder* dan sensor kompas. *Rotary encoder* berfungsi untuk mendeteksi perubahan posisi robot sedangkan sensor kompas digunakan untuk mendeteksi perubahan arah rotasi robot. Berdasarkan latar belakang ini maka penulis mengangkat permasalahan Algoritma Navigasi pada *mobile robot* dan menggunakan metode Persamaan Kinematik. Pembuatan robot ini diharapkan dapat memberikan kontribusi terhadap perkembangan dunia robotika dan menjadi motivasi untuk modul pembelajaran Laboratorium Robotika Politeknik Negeri Malang.

2. METODE PENELITIAN

2.1 Diagram Blok Sistem

Diagram blok sistem merupakan diagram yang digunakan sebagai alur dari setiap penggunaan komponen pada sistem. Pada Gambar 1 merupakan diagram blok sistem yang meliputi perangkat masukan, proses dan keluaran.

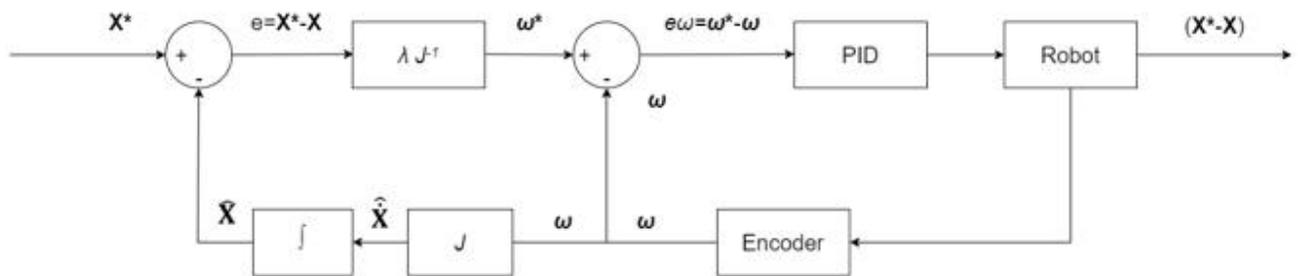


Gambar 1: Blok Diagram Sistem



Pada Gambar 1 merupakan blok diagram sistem yang terdiri dari : masukan, proses dan keluaran. Masukan pada sistem ini adalah *Rotary Encoder* yang terdapat pada internal Motor PG28 dan Kompas (MPU 9250). Rotary Encoder digunakan untuk membaca jumlah putaran pada motor dan Kompas (MPU 9250) digunakan untuk membaca arah orientasi robot berupa nilai derajat sudut. ESP32 yang berjumlah 2, yang digunakan sebagai kontrol motor dibagi menjadi 2 bagian. ESP32 (*Slave*) sebagai kontrol motor 1 dan motor 2. ESP32 (*Master*) sebagai kontrol motor 3. Dan Raspberry Pi 4 digunakan sebagai perhitungan kinematik untuk pergerakan robot. Raspberry Pi 4 dengan mikrokontroler (*Master*) menggunakan protokol komunikasi serial (*UART*), 3 buah motor DC yang dikendalikan oleh masing-masing mikrokontroler (*Master*) dan (*Slave*), driver motor DC menerima keluaran dari mikrokontroler berupa sinyal PWM (*Pulse Width Modulation*)

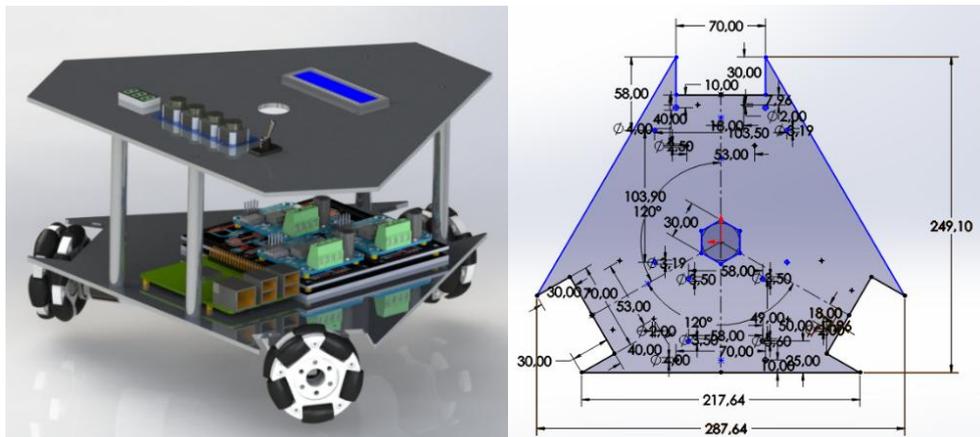
2.2 Diagram Blok Kontrol



Gambar 2: Diagram Blok Kontrol

Pada langkah awal, \mathbf{X}^* atau (x^*, y^*, θ^*) adalah posisi x^*, y^* dan orientasi (arah hadap) θ^* yang ingin dituju robot. \mathbf{X} atau (x, y, θ) adalah posisi dan orientasi aktual robot saat ini terhadap kecepatan translasi robot aktual. Error posisi e merupakan hasil dari setpoint posisi \mathbf{X}^* dikurangi posisi aktual \mathbf{X} . Nilai e dikalikan λJ^{-1} akan menghasilkan setpoint kecepatan roda robot (ω^*). Kontrol PID digunakan untuk menstabilkan kecepatan roda robot dalam kontrol loop tertutup sehingga robot dapat bergerak dengan stabil. Pembacaan kecepatan roda secara real time dilakukan oleh rotary encoder, rotary encoder inilah yang memberikan *feedback* berupa nilai kecepatan putar roda saat ini.

2.3 Perancangan Mekanik



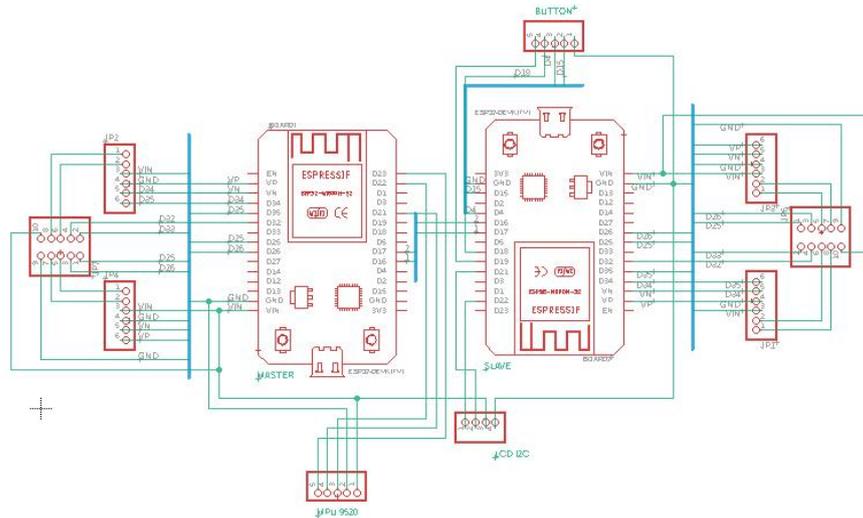
Gambar 3: Perancangan Mekanik Robot

Pada Gambar 3 merupakan perancangan mekanik robot dengan menggunakan plat alumunium berukuran 3mm untuk base atas dan base bawah, menggunakan 3 roda *omniwheels* dengan dimensi 60mm. Pada bagian atas robot terdapat saklar, *voltmeter*, *push button*, dan LCD untuk menampilkan beberapa fungsi dari fitur robot.



2.4 Perancangan Elektronik

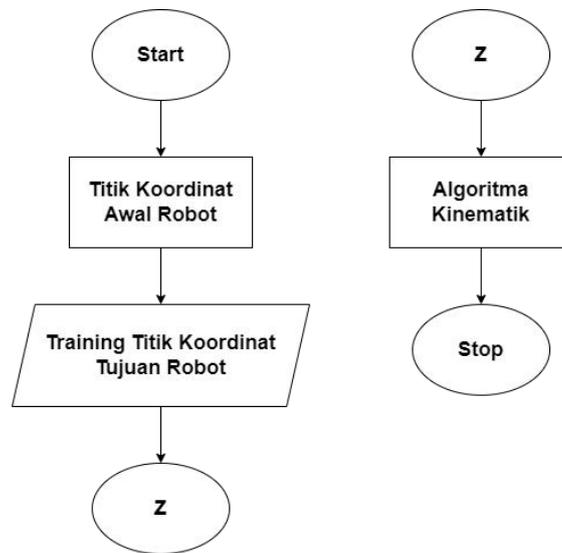
Untuk dapat bergerak bekerja dan bergerak secara maksimal, robot memerlukan rangkaian elektronik. Dari rangkaian *Power Management*, *Main Board*, maupun driver motor yang terhubung dengan motor dc untuk penggerak pada robot. Rangkaian *Main Board* pada *mobile robot* ini berfungsi sebagai *low control* pada robot yang terdiri dari 2 ESP, yaitu ESP *Slave* dan ESP *Master*. Untuk berkomunikasi, kedua ESP tersebut menggunakan komunikasi serial (UART).



Gambar 4: Rancangan Elektronik *Main Board*

2.5 Perancangan Software

Algoritma yang dijalankan robot adalah robot akan menyimpan data odometry sebagai posisi tujuan robot nantinya, robot dapat bergerak sesuai data yang disimpan hingga akhir dari data tersebut selesai dijalankan.



Gambar 5: *Flowchart* Perancangan Software

2.6 Perancangan Kinematik pada Robot Beroda 3 Omni

Dalam pembuatan *mobile robot* menggunakan metode kinematics perlu memperhitungkan pembuatan mekanik karena akan berpengaruh terhadap pergerakan robot tersebut. Perhitungan kinematik membutuhkan masukkan dari spesifikasi robot. Berikut merupakan spesifikasi robot:



Tabel 1: Spesifikasi Robot Beroda 3 Omni

No	Jenis	Nilai
1	α_1	90°
2	α_2	30°
3	α_3	150°
4	γ_1	0°
5	γ_2	60°
6	γ_3	60°
7	l	0,13 m
8	r	0,03 m

- α : Sudut arah putar roda CW (*Clock Wise*) terhadap sumbu x robot
- γ : Sudut titik tengah robot terhadap titik tengah roda
- l : Jarak titik tengah robot dengan titik tengah roda (m)
- r : Jari-jari roda pada robot (m)

3. HASIL DAN PEMBAHASAN

3.1 Pengujian *Real Time Forward & Inverse Kinematics*

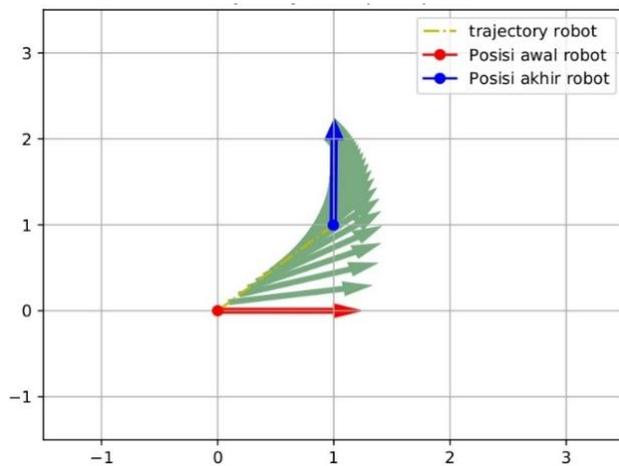
Untuk mengetahui respon nyata dari kontrol kinematik yang diterapkan pada robot, perlu adanya pengujian *forward & inverse kinematics*. Karena pastinya, akan ada perbedaan antara simulasi melalui komputer dengan hasil melalui pengujian *realtime* pada pergerakan robot langsung. Tabel 2 merupakan tabel data pengujian pembacaan posisi robot dengan memasukkan koordinat posisi x , y dan $theta$.

Tabel 2: Data Pengujian *Real Time Forward & Inverse Kinematics*

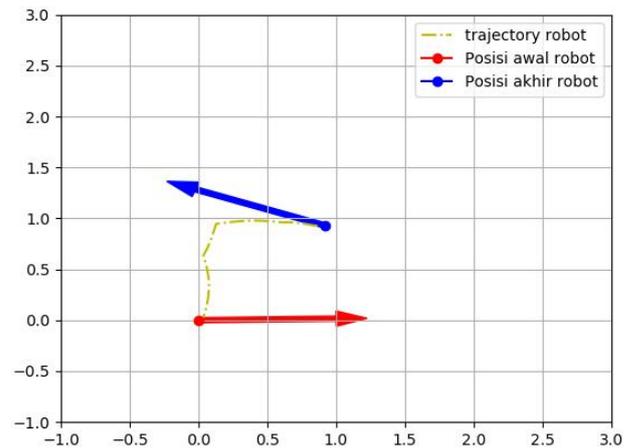
Masukan [x,y, θ]	Keluaran [x,y, θ]	Error Posisi [x,y]	Error Derajat
[1, 0, 90°]	[0.89, 0.01, 93°]	[0.11]	3°
[1, 0, 90°]	[0.94, 0.008, 83°]	[0.04]	7°
[1, 0, 90°]	[0.96, 0.009, 85°]	[0.041]	5°
[1, 0, 90°]	[0.93, 0.01, 84°]	[0.22]	6°
[1, 0, 90°]	[0.98, 0.03, 92°]	[0.006]	2°
Rata-rata <i>Error</i>		[0.083] meter	4,6°

Berdasarkan pada Tabel 2 merupakan pengujian *Forward & Inverse Kinematics* menurut masukan dan keluaran yang diinginkan dapat dikatakan bahwa *forward* dan *inverse* yang diinginkan dapat sesuai dengan masukan. Robot dapat bergerak sesuai dengan titik tujuan dan arah hadap yang diinginkan. Tentunya terdapat *error* posisi dan arah hadap, namun masih dalam jangkauan yang normal. Hasil pengujian yang dilakukan sebanyak 5 kali dengan 1 titik tujuan mendapatkan error posisi rata-rata sebesar [0.083] meter 8.3 cm dan untuk error derajat sebesar 4,6°. Hal tersebut disebabkan oleh diantaranya slip roda pada pembacaan rotary encoder dan kurangnya filter pada kompas yang digunakan.





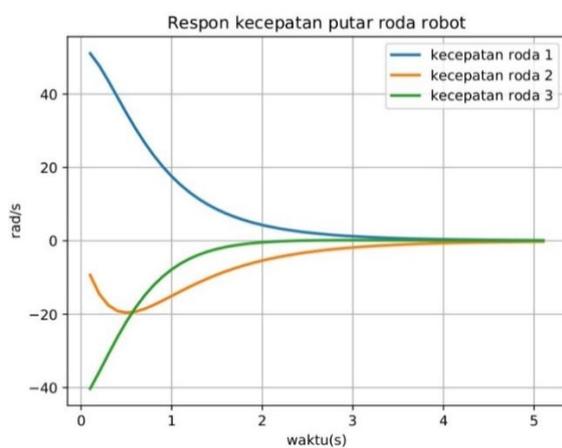
Gambar 6: *Trajectory Robot Secara Simulasi*



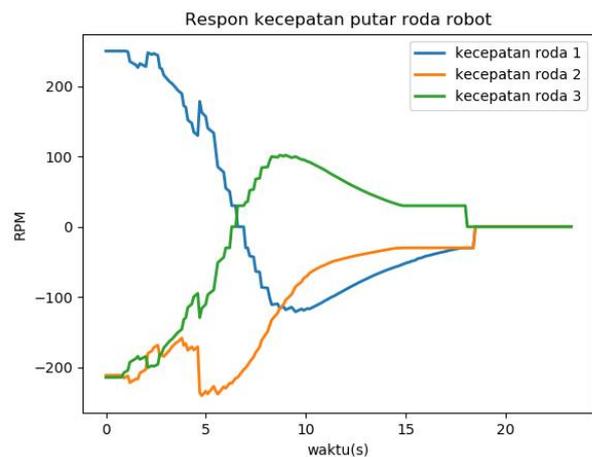
Gambar 7: *Trajectory Robot Secara Real Time*

Gambar 6 merupakan pengujian *trajectory* robot menggunakan simulasi pada komputer dengan memasukkan titik koordinat tujuan $[1, 1, 90^\circ]$ dan tentunya hasil dari simulasi tersebut sangat bagus dan menunjukkan pergerakan robot tanpa adanya *error*.

Pada Gambar 7 merupakan hasil dari *real time trajectory* robot dengan memasukkan tujuan $[1, 1, 90^\circ]$ dan hasilnya adalah arah hadap robot masih terdapat kesalahan dan masih dalam toleransi kesalahan dengan titik koordinat tujuan pada $[0.9, 1.6, 97^\circ]$. Pada saat *start* robot bergerak langsung mencari arah hadap 90° lalu kemudian robot bergerak ke titik koordinat X dan Y dengan toleransi yang masih normal. Adanya kesalahan



Gambar 8: Kecepatan Putar Roda Robot Simulasi



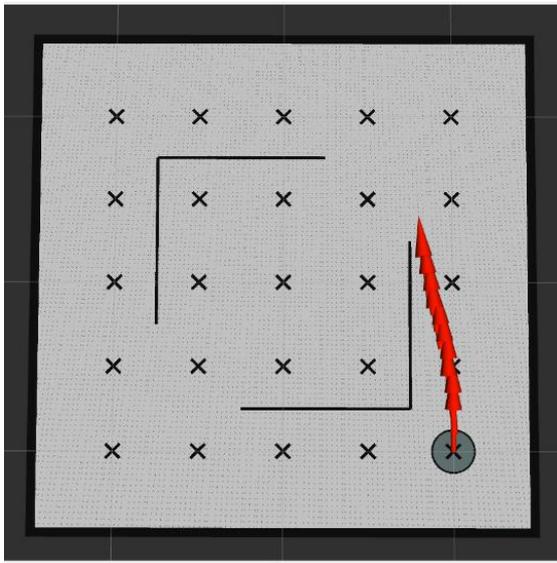
Gambar 9: Kecepatan Putar Roda Robot *Real Time*

Pada Gambar 9 didapatkan respon kecepatan putar roda robot secara *real time*, grafik tersebut berbeda dengan hasil simulasi pada Gambar 8 yang grafik tersebut halus. Namun pada grafik *real time* tersebut didapatkan respon relatif yang cukup baik dari respon motor. Grafik *real time* menunjukkan respon tiap motor yang pastinya terdapat *error* karena slip pada roda *omniwheels* terhadap bahan alas lapangan yang digunakan.

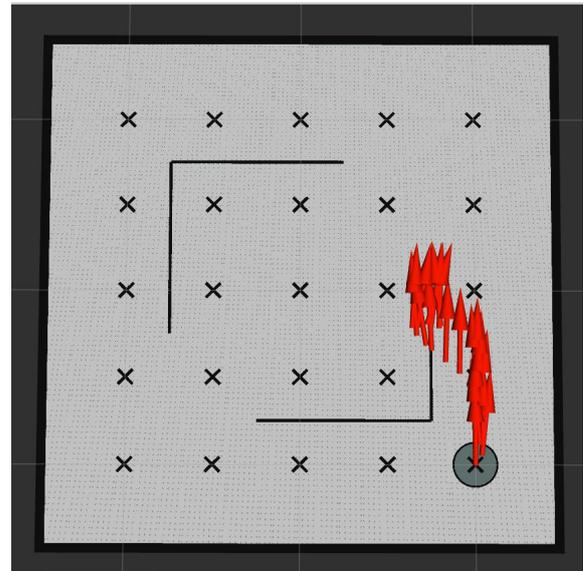
3.2 Pengujian Perekaman Data Menggunakan Sensor Odometry pada Proses Pembelajaran Navigasi Robot

Pada pengujian perekaman data menggunakan sensor odometry tujuannya robot harus mampu bergerak ke titik tujuan berdasarkan rekam data sensor odometry yang telah dibuat dan disimpan untuk tujuan tersebut. Pengujian perekaman data sensor odometry dilakukan dengan menggunakan simulasi komputer dan pengujian secara *realtime* berdasarkan pada pergerakan robot. Gambar 11 dan 12 merupakan hasil *real time trajectory* robot menuju titik akhir $[0.9, 0, 0^\circ]$.



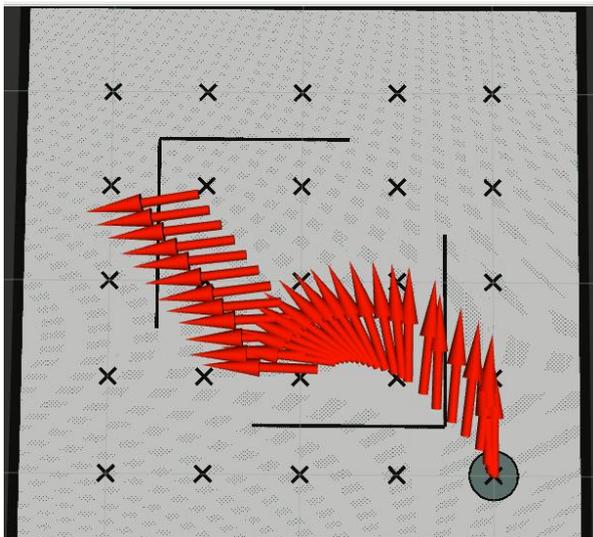


Gambar 11: Perekaman Odometry Secara Manual

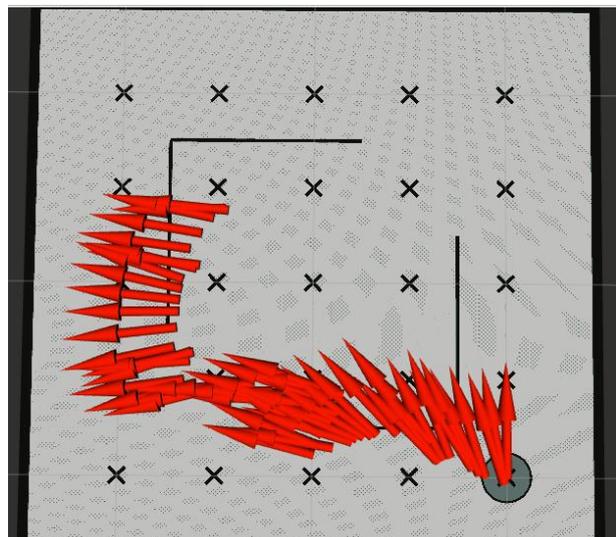


Gambar 12: Robot Bergerak Berdasarkan Data Odometry

Pada Gambar 11 adalah gambar pada saat perekaman data odometry didorong manual dan visualisasinya menggunakan Rviz. Robot didorong secara manual dengan tangan dari titik $[0,0,0]$ menuju titik tujuan $[0.9,0,0]$. Pada Gambar 12 merupakan gambar *trajectory* robot menggunakan perekaman data odometry, terlihat robot dapat menuju titik tujuan namun terdapat *error* dari arah hadap robot yang merupakan keluaran dari kompas sebesar 2° . Gambar 13 dan 14 merupakan hasil *real time trajectory* robot menuju titik akhir $[1.5, 1.5, 90^\circ]$.



Gambar 13: Perekaman Odometry Secara Manual



Gambar 14: Robot Bergerak Berdasarkan Data Odometry

Gambar 13 adalah gambar pada saat perekaman data odometry didorong manual dan visualisasinya menggunakan Rviz. Robot didorong secara manual dengan tangan dari titik $[0, 0, 0^\circ]$ menuju titik tujuan $[1.5, 1.5, 90^\circ]$. Pada Gambar 14 merupakan gambar *trajectory* robot menggunakan perekaman data odometry, dapat dilihat robot dapat menuju titik tujuan namun terdapat *error* dari arah hadap robot yang merupakan nilai dari kompas sebesar 5° .

4. KESIMPULAN

Pada hasil pengujian *real time* perekaman data menggunakan sensor odometry, robot mampu bergerak menuju titik koordinat tujuan berdasarkan rekam data sensor odometry yang telah direkam sebelumnya. Pada saat



perekaman data manual [0.9, 0.1, 0°] dan dengan menggunakan masukan dari trajectory tersebut. Pergerakan akhir dari robot [0.8, 0.1, 2°]. Robot terlihat dapat menyesuaikan terhadap titik tujuan pada pembacaan posisi menggunakan odometry meskipun terdapat *error* karena slip roda *omniwheels* terhadap lapangan yang digunakan. Penggunaan adanya Rviz dapat memudahkan monitoring pergerakan robot.

DAFTAR PUSTAKA

- [1] Awaludin, M. F. K., Winarno, T., & Siradjuddin, I. (2021). Motion Planning Robot dengan Kontrol Kinematik. *Jurnal Elektronika Otomasi Industri*, 8(2), 30-38.
- [2] Baridwan, M. Z. (2021). *Rancang Bangun Alat Hand Sanitizer Otomatis Menggunakan Esp32 Sensor Ultrasonik Hc-Sr04 Dengan Tampilan Menarik Bagi Anak* (Doctoral dissertation, Politeknik Harapan Bersama Tegal).
- [3] Fahmizal, F., Rijalussalam, D. U., & Mayub, A. (2019). Trajectory Tracking pada Robot Omni dengan Metode Odometry. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, 8(1), 35-44.
- [4] Handaya, D., & Nugraha, Y. P. (2017, October). Analisis Efisiensi Penggunaan Daya Pada Variasi Konfigurasi Rangkaian Pengendali Motor Dc. In *Prosiding Seminar Nasional Fisika (E-JOURNAL)* (Vol. 6, pp. SNF2017-CIP).
- [5] Mudarris, M., & Zain, S. G. (2020). Implementasi Sensor Inertial Measurement Unit (IMU) untuk Monitoring Perilaku Roket. *Aviation Electronics, Information Technology, Telecommunications, Electricals, Controls*, 2(1), 55-64.
- [6] Nataliana, D., Syamsu, I., & Giantara, G. (2014). Sistem Monitoring Parkir Mobil menggunakan Sensor Infrared berbasis RASPBERRY PI. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 2(1), 68.
- [7] Prabowo, B. A. (2010). Pemodelan Sistem Kontrol Motor DC dengan Temperatur Udara sebagai Pemicu. *INKOM Journal*, 2(1), 39-43.
- [8] Rachmawan, A. (2017). *Penentuan Posisi Robot Sepak Bola Beroda Menggunakan Rotary Encoder Dan Kamera* (Doctoral dissertation, Institut Teknologi Sepuluh Nopember).
- [9] Sofwan, A., Mulyana, H. R., Afrisal, H., & Goni, A. (2019, September). Development of Omni-Wheeled Mobile Robot Based-on Inverse Kinematics and Odometry. In 2019 6th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE) (pp. 1-6).
- [10] Taufiqqurohman, M., & Sari, N. F. (2018, August). Odometry Method and Rotary Encoder for Wheeled Soccer Robot. In *IOP Conference Series: Materials Science and Engineering* (Vol. 407, No. 1, p. 012103). IOP Publishing.

