

# Sistem Kontrol *Mobile* Robot Enam Roda *Omni Wheels* dengan Metode *Position-Based Proportional*

Muhammad Imaduddin Amrullah<sup>1</sup>, Totok Winarno<sup>2</sup>, Gillang Al Azhar<sup>3</sup>

e-mail: [imaduddin76@gmail.com](mailto:imaduddin76@gmail.com), [totok.winarno@polinema.ac.id](mailto:totok.winarno@polinema.ac.id), [gillang\\_al\\_azhar@polinema.ac.id](mailto:gillang_al_azhar@polinema.ac.id)

<sup>1,2,3</sup>Jurusan Teknik Elektro, Politeknik Negeri Malang, Jalan Soekarno Hatta No.9 Malang, Indonesia

## Informasi Artikel

### Riwayat Artikel

### Kata kunci:

*Mobile Robots Control*  
*Omni-directional Robot*  
*Position-based proportional*

### Keywords:

*Mobile Robots Control*  
*Omni-directional Robot*  
*Position-based proportional*

### Penulis Korespondensi:

Muhammad Imaduddin Amrullah,  
Jurusan Teknik Elektro,  
Politeknik Negeri Malang,  
Jl. Soekarno Hatta No. 9, Malang, Jawa Timur, Indonesia, Kode Pos. 65141

## ABSTRAK

Penggunaan robot identik dalam otomasi industri, seperti sebagai pemindah barang. Secara umum, fungsi dasar robot adalah mengubah posisinya sesuai dengan posisi referensi, seperti pada *mobile robot*. Perubahan posisi robot harus mempertimbangkan berbagai faktor, seperti kecepatan robot dan lintasan yang akan dilalui robot. Penentuan jumlah roda *omni-wheels* pada robot berpengaruh dalam kerentanan roda jika terjadi tergelincir. Dengan menggunakan enam roda *omni-wheels* yang dikendalikan menggunakan kontroler PI dengan metode Ziegler-Nichols, diharapkan mampu mengurangi tergelincirnya roda robot ketika melintasi permukaan yang kurang datar, sehingga robot dapat mencapai tujuan dengan cepat. Pada kecepatan 500 RPM, masing-masing motor dengan parameter gain PI  $K_p = 1,2485$  dan  $K_i = 0,0046$  menghasilkan error steady-state dengan kisaran 1,2% hingga 1,6%. Selain itu, untuk mendukung pergerakan robot dalam mencapai tujuan lebih cepat, digunakan metode *position-based proportional*. Metode *position-based proportional* memungkinkan dapat menyempurnakan kecepatan robot saat bergerak, dengan menyeimbangkan jarak tempuh tujuan dengan kecepatan robot. Pergerakan robot secara *position-based proportional* dari posisi dan orientasi (0 m, 0 m, 0°) menuju posisi dan orientasi (0 m, 3 m, 10°), menghasilkan posisi dan orientasi akhir robot (0 m, 2,66m, 9,74°) dengan error vektor senilai 8,62%.

## ABSTRACT

*The use of identical robots in industrial automation, such as moving goods. In general, the basic function of a robot is to change its position according to a reference position, as in a mobile robot. Changing the position of the robot must take into account various factors, such as the speed of the robot and the path the robot will follow. Determining the number of omni-wheels on the robot influences the vulnerability of the wheels if a slip occurs. By using six omni-wheels controlled using a PI controller using the Ziegler-Nichols method, it is hoped that it will be able to reduce the slipping of the robot's wheels when crossing less than flat surfaces, so that the robot can reach its destination quickly. At a speed of 500 RPM, each motor with PI gain parameters  $K_p = 1.2485$  and  $K_i = 0.0046$  produces a steady-state error in the range of 1.2% to 1.6%. Apart from that, to support the robot's movement in reaching its goal more quickly, the position-based proportional method is used. The position-based proportional method makes it possible to fine-tune the robot's speed when moving, by balancing the distance traveled to the destination with the robot's speed. The position-based proportional movement of the robot from pose (0 m, 0 m, 0°) to pose (0 m, 3 m, 10°), produces the robot's final pose (0 m, 2.66m, 9.74°) with vector error is 8.62%.*



## 1. PENDAHULUAN

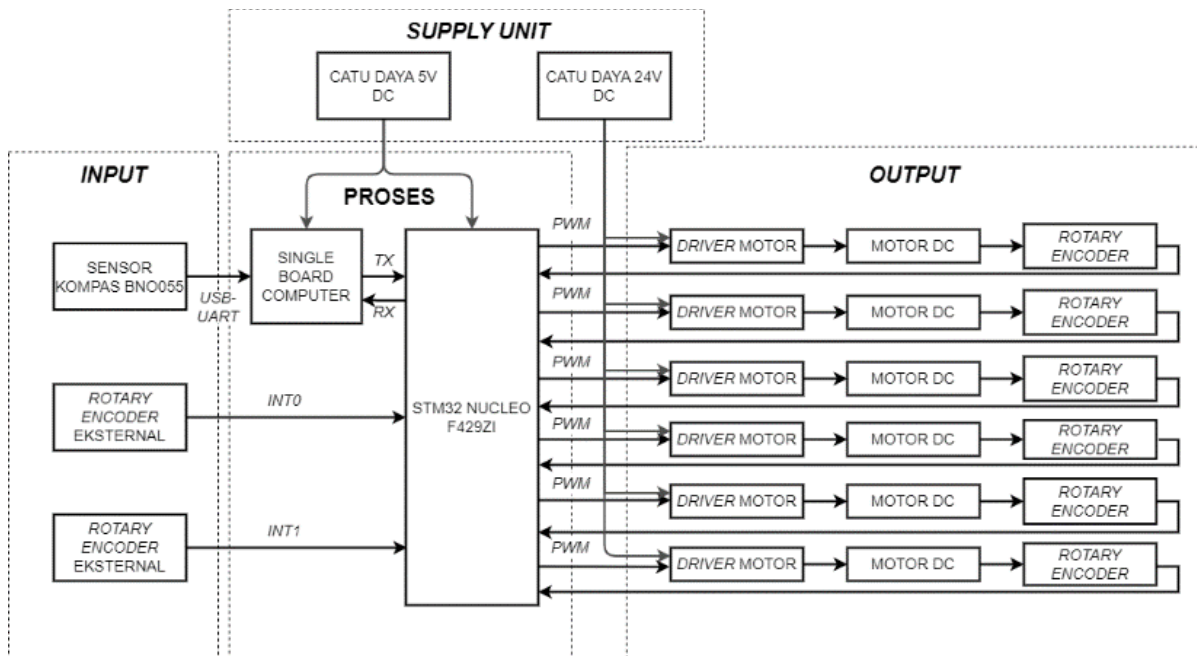
Perkembangan teknologi di bidang robotika semakin pesat untuk membantu kehidupan sehari-hari manusia. Robot sering digunakan untuk menjalankan suatu rutinitas mulai dari hal yang mudah hingga kompleks. Salah satu fungsi robot yang sederhana adalah mengubah posisinya sesuai dengan posisi yang ditentukan. Untuk dapat mengubah posisi robot, diperlukan mekanisme pergerakan tertentu yang dapat diterapkan pada robot. Robot dengan mekanisme pergerakan roda merupakan salah satu jenis mekanisme dasar yang berkaitan erat dengan fungsinya untuk dapat mengubah posisi. Selain itu, robot tersebut sangat sering digunakan karena kesederhanaan dalam pembuatan dan pengendalian.

Salah satu jenis roda yang sangat kerap digunakan adalah roda *omni-wheels*. Roda *omni-wheels* merupakan jenis roda yang memiliki roda di sekeliling inti roda[1]. Roda yang mengelilingi inti roda tersebut dinamakan *freewheel roller*, yang mana berputar secara tegak lurus dengan poros inti roda[2]. Roda *omni-wheels* memiliki kelebihan untuk dapat bergerak ke segala arah tanpa melakukan manuver hingga mengubah orientasi roda, seperti pada roda konvensional. Roda *omni-wheels* yang digunakan pada *mobile robot* umumnya berjumlah tiga sampai empat roda. Jumlah roda *omni-wheels* yang digunakan menentukan stabilitas robot dan kecepatan gerak robot.

Pada *mobile robot* yang menggunakan tiga roda *omni-wheels*, memiliki kelebihan untuk menghasilkan traksi yang lebih besar akibat gaya reaksi hanya didistribusikan pada tiga titik[3]. Akan tetapi, kelemahan pada *mobile robot* yang menggunakan tiga roda *omni-wheels* adalah sulit dalam pergerakan robot yang dikendalikan secara manual. Sedangkan pada *mobile robot* yang menggunakan empat roda *omni-wheels*, memiliki kecenderungan selip jika melintasi permukaan yang tidak datar atau distribusi beban pada *mobile robot* yang rendah. Kelemahan-kelemahan tersebut disebabkan oleh kerentanan *freewheel roller* terhadap getaran vertikal yang dihasilkan oleh guncangan berturut-turut, setiap kali terjadi kontak dengan permukaan[4].

## 2. METODE PENELITIAN

### 2.1 Perancangan Sistem



Gambar 1: Blok Diagram Sistem

Pada Gambar 1 merupakan blok diagram sistem yang secara keseluruhan mendapatkan sumber tegangan 24 VDC. Sumber tegangan tersebut didistribusikan dan diturunkan menjadi 5 VDC untuk subsistem yang menggunakan

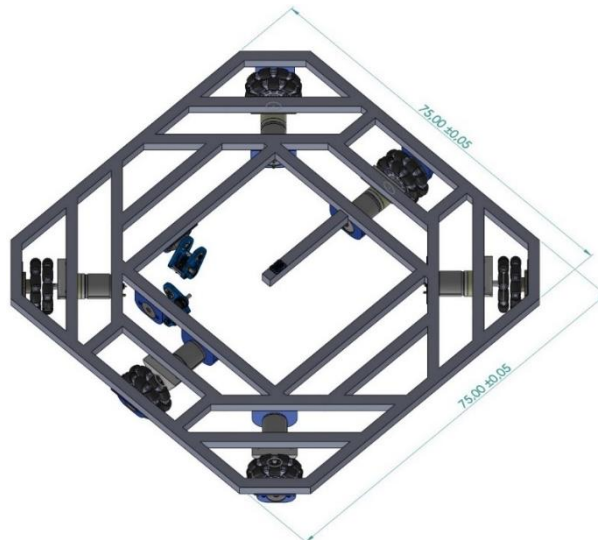


sumber tegangan 5 VDC. *Input* pada sistem ini adalah sensor BNO055 digunakan untuk membaca orientasi robot (sumbu-*z*) dan dua *rotary encoder* eksternal digunakan untuk membaca posisi robot (sumbu-*x* dan sumbu-*y*). Hasil pembacaan sensor BNO055 akan diolah dalam NVIDIA Jetson Nano (*Singel Board Computer*) menjadi kecepatan putaran pada sumbu-*z* ( $^{\circ}/s$ ). sedangkan hasil pembacaan dua *rotary encoder* eksternal diolah dalam STM32 Nucleo F429ZI (Mikrokontroler). Pengolahan pembacaan masing-masing *rotary encoder* eksternal pada STM32 Nucleo F429ZI menghasilkan kecepatan putaran dalam satuan RPM yang akan dikirim kepada NVIDIA Jetson Nano dengan protokol komunikasi serial. Kemudian, kecepatan putaran tersebut diolah menjadi kecepatan linear (m/s).

Berdasarkan pengolahan data dari masing-masing sensor tersebut, pada NVIDIA Jetson Nano dilakukan proses perhitungan *forward velocity kinematics* untuk mendapatkan posisi dan orientasi robot sesuai perpindahan dalam bidang dua dimensi[5]. Posisi dan orientasi (*Pose*) yang akan dituju robot juga ditentukan pada NVIDIA Jetson Nano sebagai posisi dan orientasi tujuan robot. Berdasarkan posisi dan orientasi yang akan dituju robot dan posisi dan orientasi robot sebenarnya, dapat dihasilkan kecepatan enam roda yang dibutuhkan dalam proses perhitungan *inverse velocity kinematics*[6]. Sebagai penyesuaian antara kecepatan robot yang dibutuhkan untuk mencapai posisi dan orientasi tujuan, metode *position-based proportional* diterapkan. Keseluruhan proses pada NVIDIA Jetson Nano tersebut dijalankan menggunakan perangkat lunak ROS (*Robot Operating System*). ROS (*Robot Operating System*) adalah perangkat lunak *open source* yang bertujuan untuk membangun sarana termodularisasi dan terdistribusi untuk semua pengguna dan pengembang robotika, sehingga dapat saling berbagi algoritma, program, dan mengembangkan efisiensi dalam pengembangan robot[7]. ROS (*Robot Operating System*) merupakan perangkat lunak *framework* berbasis Linux yang memberikan fasilitas sistem operasi antara lain, *hardware abstraction*, kendali perangkat *low-level*, fungsi yang seringkali digunakan, penyampaian message antara berbagai proses, dan manajemen *package*[8].

Hasil enam kecepatan roda tersebut akan dikirimkan ke STM32 Nucleo F429ZI sebagai *setpoint* setiap kecepatan motor sebagai aktuator roda *omni-wheels*. Masing-masing motor terdapat *rotary encoder* internal yang digunakan sebagai umpan balik kecepatan masing-masing roda saat berputar. Dengan demikian, digunakan kontroler PI dengan metode Ziegler-Nichols agar masing-masing roda *omni-wheels* dapat berputar dengan stabil. Pada bagian *output* terdapat *driver* motor DC yang digunakan untuk menggerakkan motor DC berdasarkan gelombang PWM yang diterima. PWM tersebut didapatkan dari gelombang PWM yang dibangkitkan oleh STM32 Nucleo F429ZI, sesuai dengan hasil dari kontroler PI yang digunakan.

## 2.2 Perancangan Mekanik



Gambar 2: Desain Robot Enam Roda *Omni-Wheels*

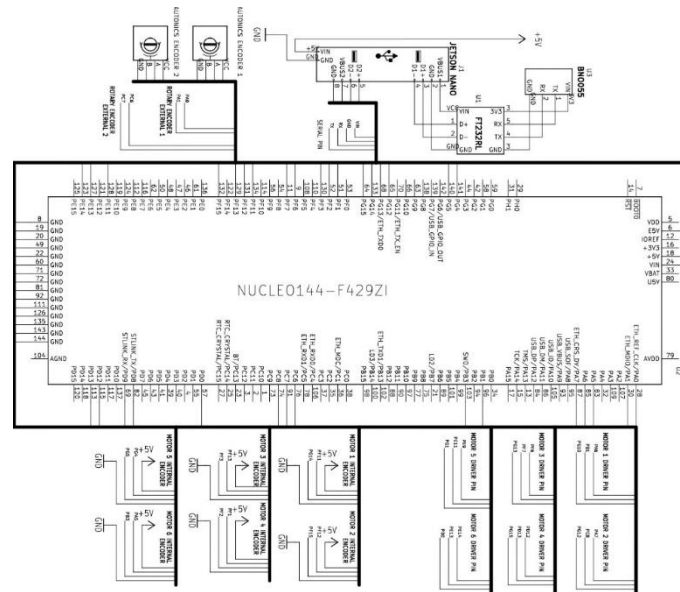
Pada Gambar 2 merupakan perancangan mekanik robot enam roda *omni-wheels* yang memiliki dimensi yakni, panjang 75 cm, lebar 75 cm, dan tinggi 9,85 cm. Bahan dari base robot adalah profil aluminium dengan dimensi 2 cm



× 2 cm dengan tebal 1,5 cm. Setiap profil dihubungkan dengan plat aluminium dengan tebal 1 mm menggunakan paku rivet dengan ukuran 3,2 mm. Setiap motor DC akan dipasang dengan *holder* yang terbuat dari PLA (*polyactic acid*) dan bracket pada bagian gearbox, yang terbuat dari aluminium dengan tebal 3 mm.

Kemudian, bagian-bagian yang telah dirakit tersebut akan dipasang pada base menggunakan mur dan baut dengan ukuran M6. Roda *omni-wheels* yang digunakan memiliki diameter 10 cm dan memiliki dua sisi. Pada salah satu sisi roda *omni-wheels* dipasang dengan as dari *gearbox* motor DC. Sisi lain dari roda *omni-wheels* dipasangkan dengan *pillow block bearing* dengan menambahkan as dengan ukuran 10 mm.

### 2.3 Perancangan Elektrik

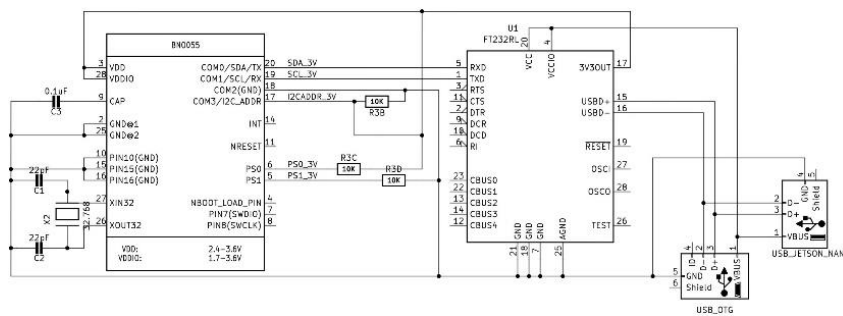


Gambar 3: Rangkaian Skematik Utama Robot

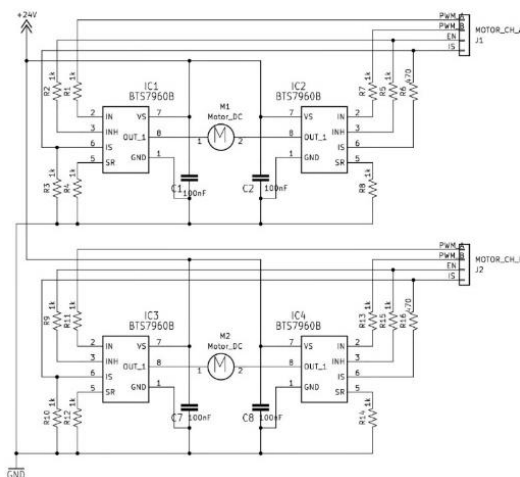
Rangkaian skematik utama robot pada Gambar 3 dirancang dengan menentukan jenis periferil yang akan digunakan pada jenis komponen elektronika sebagai kontroler tingkat rendah (*low-level control*) menggunakan STM32 Nucleo F429ZI. Periferil yang digunakan untuk membangkitkan gelombang PWM untuk *driver* motor DC adalah periferil *timer*. Sedangkan periferil yang digunakan untuk menerima pembacaan *rotary encoder* adalah *interrupt*, baik *rotary encoder* eksternal dan *rotary encoder* pada motor DC. Sedangkan untuk jenis komponen elektronika sebagai kontroler tingkat tinggi (*high-level control*), yakni NVIDIA Jetson Nano hanya menggunakan kabel data USB untuk dapat berkomunikasi secara serial dengan kontroler tingkat rendah.

Sensor IMU (*Inertial Measurement Unit*) BNO055 yang digunakan sebagai pembaca orientasi robot mengirimkan data pembacaan kepada NVIDIA Jetson Nano melalui Modul FT232RL sebagai perantara komunikasi serial USB. Hal tersebut dikarenakan sensor IMU BNO055 merupakan sensor yang telah terintegrasi dengan mikrokontroler cortex M0+ 32-bit yang menjalankan perangkat lunak *sensor fusion*, dengan protokol komunikasi dasar I2C (*Inter-Integrated Circuit*) [9]. Sehingga, untuk berinteraksi dengan sensor IMU BNO055 dibutuhkan modul komunikasi dengan rangkaian skematik seperti pada Gambar 4 berikut.





Gambar 4: Rangkaian Skematik Sensor IMU BNO055 dan Modul FT232RL



Gambar 5: Rangkaian Skematik *Driver* Motor DC 2-Channel

Pada Gambar 5 merupakan rangkaian skematik *driver* motor DC. Rangkaian driver motor ini memiliki *output* 2-channel, sehingga satu buah *driver* motor DC ini dapat digunakan untuk menggerakkan dua motor DC dengan kecepatan dan arah putar yang independen. *Driver* motor DC ini menggunakan IC BTS 7960 yang memuat dua MOSFET dengan topologi *half-bridge* beserta dengan *gate driver* MOSFET tersebut. Sehingga untuk menggerakkan satu motor DC yang dapat berputar dua arah, dibutuhkan dua IC BTS 7960. Jenis IC BTS 7960 ini bekerja pada gelombang PWM dengan siklus kerja logika *active high*. Gelombang PWM yang diterima oleh IC BTS 7960 dikirimkan oleh STM32 Nucleo F429ZI, dengan menggunakan periferial timer untuk membangkitkan gelombang PWM.

### 2.4 Perancangan Kontrol

Berdasarkan spesifikasi mekanik dari robot, akan dilakukan perancangan kontrol yang mengacu pada permodelan matematis mengenai pengendalian robot. Permodelan matematis secara spesifik mengacu pada kinematika robot beroda yang menggunakan roda *omni-wheels*. Kinematika robot tersebut terdiri dari *forward velocity kinematics* dan *inverse velocity kinematics*. Selain itu, untuk mendukung pergerakan robot, masing-masing aktuator robot juga diterapkan kontroler PI. Berikut pada Tabel I merupakan spesifikasi robot yang menjabarkan dimensi robot.

TABEL I: SPESIFIKASI MEKANIK UNTUK PERANCANGAN KONTROL ROBOT

No.	Spesifikasi	Nilai
1	Jarak roda <i>omni-wheels</i> dengan titik tengah robot ( $L$ )	$\pm 52$ cm
2	Jarak roda <i>rotary encoder</i> eksternal dengan titik tengah robot ( $l$ )	$\pm 40$ cm
3	Sudut yang dibentuk antara putaran searah jarum jam dari roda <i>omni-wheels</i> dengan arah hadap robot ( $\alpha_n$ )	$\alpha_1 = 0^\circ, \alpha_2 = 45^\circ,$ $\alpha_3 = 135^\circ, \alpha_4 = 180^\circ,$ $\alpha_5 = -135^\circ, \alpha_6 = -45^\circ$



4	Sudut yang dibentuk antara poros dari roda <i>omni-wheels</i> dengan arah hadap robot ( $\gamma_n$ )	$\gamma_1 = 90^0, \gamma_2 = 45^0,$ $\gamma_3 = -45^0, \gamma_4 = -90^0,$ $\gamma_5 = -135^0, \gamma_6 = 135^0$
5	Sudut yang dibentuk anantara putaran searah jarum jam dari roda <i>rotary encoder</i> eksternal dengan arah hadap robot ( $\alpha_i$ )	$\alpha_a = -90^0, \alpha_b = 180^0$
6	Diameter roda <i>omni-wheels</i> ( $R$ )	10 cm
7	Diameter roda <i>rotary encoder</i> eksternal ( $r$ )	5 cm

**2.4.1 Permodelan Matematis *Forward Velocity Kinematics***

*Forward velocity kinematics* akan memodelkan spesifikasi roda *rotary encoder* eksternal pada robot. Permodelan ini akan menentukan posisi relatif robot terhadap posisi global berdasarkan kecepatan putaran roda *rotary encoder* eksternal tersebut ( $\omega_a, \omega_b$ ). Berdasarkan spesifikasi pada tabel di atas, dapat dirumuskan permodelan *forward velocity kinematics* sebagai berikut dalam model *state-space*.

$$\dot{x} = J^{-1}R(\theta)^{-1}\dot{q} \tag{1}$$

Pada persamaan (1) dapat dijabarkan menjadi bentuk matriks seperti sebagai berikut:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\alpha_a) & \sin(\alpha_a) & l_a \\ \cos(\alpha_b) & \sin(\alpha_b) & l_b \\ 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \dot{q}_a \\ \dot{q}_b \\ 0 \end{bmatrix} \frac{2\pi r}{60} \tag{2}$$

Berdasarkan penjabaran dari permodelan dalam persamaan (2), akan didapatkan kecepatan linear robot dan kecepatan sudut robot secara global ( $\dot{x}, \dot{y}, \dot{\theta}$ ). Kecepatan linear dan kecepatan sudut robot akan diubah menjadi estimasi posisi dan orientasi robot secara global melalui proses integral, yang mana akan dijabarkan dalam persamaan (3) dan (5) berikut[10].

$$x_t = x_{t-1} + \int \dot{x} dt \tag{3}$$

$$y_t = y_{t-1} + \int \dot{y} dt \tag{4}$$

$$\theta_t = \theta_{t-1} + \int \dot{\theta} dt \tag{5}$$

Proses perhitungan integral ini dilakukan pada sistem yang bekerja secara diskrit. Sehingga diperlukan pendekatan integral numerikal seperti pada persamaan (6) hingga (8), yakni menggunakan integral *rectangular*[11].

$$x_t = \int_0^\infty \dot{x} dt \approx \sum_{i=1}^n \dot{x}(t) \times \Delta t \tag{6}$$

$$y_t = \int_0^\infty \dot{y} dt \approx \sum_{i=1}^n \dot{y}(t) \times \Delta t \tag{7}$$

$$\theta_t = \int_0^\infty \dot{\theta} dt \approx \sum_{i=1}^n \dot{\theta}(t) \times \Delta t \tag{8}$$

**2.4.2 Permodelan Matematis *Inverse Velocity Kinematics***

Permodelan ini akan menentukan kecepatan setiap roda *omni-wheels* yang dibutuhkan untuk mencapai posisi dan orientasi tujuan robot. Kecepatan tersebut dihasilkan berdasarkan selisih atau *error* antara hasil permodelan *forward velocity kinematics* berupa estimasi posisi dan orientasi robot ( $\bar{x}, \bar{y}, \bar{\theta}$ ) dengan posisi dan orientasi tujuan ( $x^*, y^*, \theta^*$ ). Berdasarkan spesifikasi pada tabel di atas dapat dirumuskan permodelan matematis *inverse velocity kinematics* dalam model *state-space* pada persamaan (9) sebagai berikut.

$$\omega = J^{-1}R^{-1}\dot{e} \tag{9}$$





Sebelum menentukan kecepatan tiap roda *omni-wheels*, akan dijabarkan terlebih dahulu permodelan matematis dari spesifikasi salah satu roda *omni-wheels* dalam bentuk matriks pada persamaan (10) sebagai berikut[12].

$$u_i = \begin{bmatrix} \sin(\alpha_i) \\ \cos(\alpha_i) \\ l(\sin(\alpha_i)\cos(\gamma_i) - \cos(\alpha_i)\sin(\gamma_i)) \end{bmatrix}^T = \begin{bmatrix} \sin(\alpha_i) \\ \cos(\alpha_i) \\ l(\sin(\alpha_i - \gamma_i)) \end{bmatrix}^T \tag{10}$$

Berdasarkan bentuk matriks di atas, dapat diketahui permodelan robot enam roda *omni-wheels* dalam bentuk invers matriks Jacobian seperti pada persamaan (10) berikut.

$$J^{-1} = \begin{bmatrix} \cos(\alpha_1) & \sin(\alpha_1) & l(\sin(\alpha_1 - \gamma_1)) \\ \cos(\alpha_2) & \sin(\alpha_2) & l(\sin(\alpha_2 - \gamma_2)) \\ \cos(\alpha_3) & \sin(\alpha_3) & l(\sin(\alpha_3 - \gamma_3)) \\ \cos(\alpha_4) & \sin(\alpha_4) & l(\sin(\alpha_4 - \gamma_4)) \\ \cos(\alpha_5) & \sin(\alpha_5) & l(\sin(\alpha_5 - \gamma_5)) \\ \cos(\alpha_6) & \sin(\alpha_6) & l(\sin(\alpha_6 - \gamma_6)) \end{bmatrix} \tag{10}$$

Hasil dari perhitungan *inverse velocity kinematics* merupakan kecepatan putaran yang dibutuhkan masing-masing roda dalam satuan RPM. Maka, dengan mendapatkan *error* posisi dan orientasi robot terhadap posisi dan orientasi tujuan dapat diketahui kecepatan linear robot yang dibagi dengan waktu. Sehingga, untuk mengendalikan robot dengan kecepatan yang proporsional sesuai dengan posisi dan orientasi terhadap tujuannya, maka pada persamaan (11) berikut menunjukkan permodelan matematis untuk metode *position-based proportional*.

$$\dot{e} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \times \frac{60}{2\pi r} = \begin{bmatrix} x^*(t) - \bar{x}(t) / \Delta t \\ y^*(t) - \bar{y}(t) / \Delta t \\ \theta^*(t) - \bar{\theta}(t) / \Delta t \end{bmatrix} \times \frac{60}{2\pi r} \tag{11}$$

Sehingga didapatkan vektor kecepatan putaran enam roda *omni-wheels* berdasarkan penjabaran pada persamaan (9) hingga persamaan (11), yang akan dijabarkan sesuai pada model *state-space* pada persamaan (9) dalam persamaan (12) berikut.

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \\ \omega_5 \\ \omega_6 \end{bmatrix} = \begin{bmatrix} \cos(\alpha_1) & \sin(\alpha_1) & l(\sin(\alpha_1 - \gamma_1)) \\ \cos(\alpha_2) & \sin(\alpha_2) & l(\sin(\alpha_2 - \gamma_2)) \\ \cos(\alpha_3) & \sin(\alpha_3) & l(\sin(\alpha_3 - \gamma_3)) \\ \cos(\alpha_4) & \sin(\alpha_4) & l(\sin(\alpha_4 - \gamma_4)) \\ \cos(\alpha_5) & \sin(\alpha_5) & l(\sin(\alpha_5 - \gamma_5)) \\ \cos(\alpha_6) & \sin(\alpha_6) & l(\sin(\alpha_6 - \gamma_6)) \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^*(t) - \bar{x}(t) / \Delta t \\ y^*(t) - \bar{y}(t) / \Delta t \\ \theta^*(t) - \bar{\theta}(t) / \Delta t \end{bmatrix} \times \frac{60}{2\pi r} \tag{12}$$

### 2.4.3 Permodelan Matematis Kontroler PI untuk *Low-Level Control*

Kontroler PI pada perancangan ini akan diterapkan pada *low-level control* dari robot, yakni untuk mengendalikan motor DC sebagai aktuator roda *omni-wheels*. Metode untuk menyesuaikan parameter konstanta *gain* kontroler PI yang digunakan adalah metode Ziegler-Nichols. Pada persamaan (13) dan (14) merupakan penjabaran model matematis dari kontroler PI dengan metode Ziegler-Nichols.

$$u(t) = K_U \times \left[ \left( e(t) \times \frac{1}{2,2} \right) + \left( \frac{1}{T_U} \times \frac{1,2}{2,2} \times \int_0^t e(t) dt \right) \right] \tag{13}$$

$$u(t) = \frac{K_P}{0,454K_U} \times e(t) + \frac{K_I}{0,545K_U} \times \int_0^t e(t) dt \tag{14}$$

## 3. HASIL DAN PEMBAHASAN

### 3.1 Pengujian Sensor IMU BNO055

Pengujian dan analisa sensor IMU BNO055 bertujuan untuk mengkalibrasi hasil pembacaan sudut *yaw*( $\theta$ ) dari sensor IMU BNO055 yang terletak pada robot. Sudut *yaw* tersebut didapatkan dengan memperoleh data kecepatan



sudut pada sumbu- $z$  ( $\theta$ ) dan mengolahnya dengan melakukan perhitungan secara integral. Hasil tersebut selanjutnya akan dibandingkan dengan alat ukur menggunakan busur, yang disajikan pada Tabel II berikut.

TABEL II: PENGUJIAN SENSOR IMU BNO055 UNTUK SUDUT *YAW* ROBOT

No.	Pembacaan Busur (°)	Pembacaan Sensor IMU BNO055 (°)	Error $\theta$ (%)
1	45	44,873	0,282
2	60	58,942	1,763
3	90	90,009	0,01
4	135	133,108	1,401
5	180	177,990	1,116
<b>Rata-rata</b>			0,914

Berdasarkan pada data hasil pengujian sensor IMU BNO055 untuk pembacaan sudut *yaw* robot memiliki nilai rata-rata error sebesar 0,914%. *Error* tersebut dapat dihasilkan dari tingkat sensitivitas saat sensor IMU BNO055 membaca. Tingkat sensitivitas tersebut mempengaruhi pembacaan kecepatan sudut *yaw*. Sebagaimana sebuah getaran kecil yang apabila diamati dengan mata tidak mengubah orientasi robot, akan tetapi pembacaan sensor IMU BNO055 memberikan perubahan nilai. Hal tersebut dapat mempengaruhi saat proses perhitungan integral *rectangular*, yang mana dapat terakumulasi seiring berjalanya waktu. *Error* yang terakumulasi tersebut dinamakan sebagai *truncation error*, yakni error yang terjadi pada sebuah pendekatan integral numerikal pada suatu sistem yang bekerja secara diskrit.

### 3.2 Pengujian Rotary Encoder Eksternal

Pengujian *rotary encoder* eksternal bertujuan untuk mengolah hasil pembacaan *rotary encoder* eksternal menjadi kecepatan relatif robot pada sumbu- $x$  dan sumbu- $y$ . Hasil tersebut selanjutnya diolah menjadi kecepatan mutlak robot pada sumbu- $x$  dan sumbu- $y$  secara global dengan menggunakan permodelan matematis *forward velocity kinematics*. Hasil dari *forward velocity kinematics* diolah menjadi posisi mutlak robot pada sumbu- $x$  dan sumbu- $y$  secara global dengan penerapan integral *rectangular*, yang mana disajikan pada Tabel III dan Tabel IV berikut.

TABEL III: PENGUJIAN *ROTARY ENCODER* EKSTERNAL SEBAGAI PEMBACAAN POSISI SUMBU- $X$

Percobaan ke-n	1 m		2 m	
	Hasil (m)	Error (%)	Hasil (m)	Error (%)
1	1,0029	0,29	1,9934	0,33
2	1,0019	0,19	1,9682	1,59
3	0,9956	0,44	1,9802	0,99
<b>Rata-rata</b>	1,0001	0,306	1,9806	0,97

TABEL IV: PENGUJIAN *ROTARY ENCODER* EKSTERNAL SEBAGAI PEMBACAAN POSISI SUMBU- $Y$

Percobaan ke-n	1 m		2 m	
	Hasil (m)	Error (%)	Hasil (m)	Error (%)
1	1,0108	1,08	2,0287	1,43
2	1,0070	0,7	2,0313	1,56
3	0,997	0,3	2,0278	1,39
<b>Rata-rata</b>	1,005	0,693	2,0293	1,46

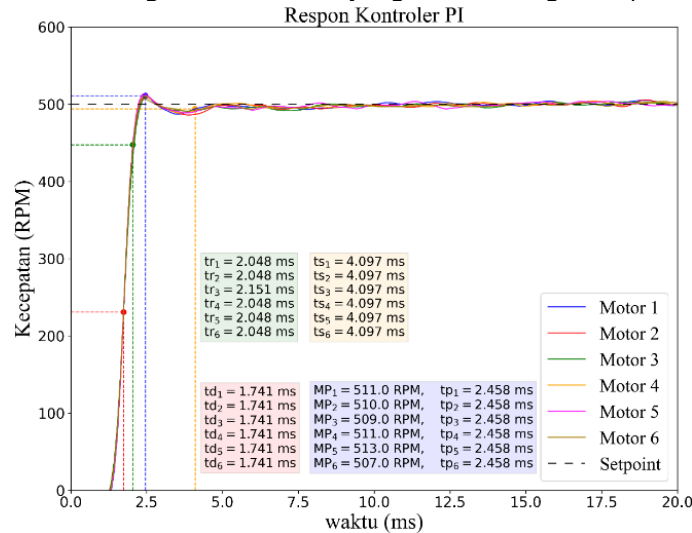
Pada hasil pengujian *rotary encoder* eksternal untuk mengetahui posisi robot pada sumbu- $x$  dan sumbu- $y$ , menunjukkan nilai *error* yang berbeda untuk jarak dan jumlah percobaan yang sama. Rata-rata error posisi robot pada sumbu- $x$  sebesar 0,306% dan pada sumbu- $y$  sebesar 0,693%. Perbedaan tersebut dihasilkan akibat *truncation error* saat melakukan perhitungan integral dari kecepatan robot secara global berdasarkan *forward velocity kinematics* menjadi posisi robot secara global.





### 3.3 Pengujian *Low-Level Control* pada Robot dengan Kontroler PI

Pengujian *low-level control* dilakukan untuk menyesuaikan kecepatan putaran seluruh enam motor DC sebagai aktuator roda omni-wheels agar sesuai dengan *setpoint* yang diberikan. Pada *low-level control* yang diterapkan pada masing-masing motor, menggunakan kontroler PI dengan metode Ziegler-Nichols untuk menentukan konstanta *gain* kontroler PI. Hasil yang didapatkan dari metode Ziegler Nichols untuk konstanta *gain* kontroler PI adalah  $K_p$  senilai 1,2485 dan  $K_i$  senilai 0,0046. Gambar 6 berikut ini merupakan grafik respon kontroler PI pada enam motor DC dengan konstanta *gain* kontroler PI yang identik dengan *setpoint* senilai 500 RPM.



GAMBAR 6: GRAFIK RESPON KONTROLER PI PADA ENAM MOTOR DC

Berdasarkan grafik respon kontroler PI pada enam motor DC pada Gambar 6, dapat diketahui karakteristik respon enam motor DC pada *domain* waktu yang dijabarkan pada Tabel V berikut ini.

TABEL V: PENGUJIAN KONTROLER PI UNTUK *LOW-LEVEL CONTROL* PADA ROBOT

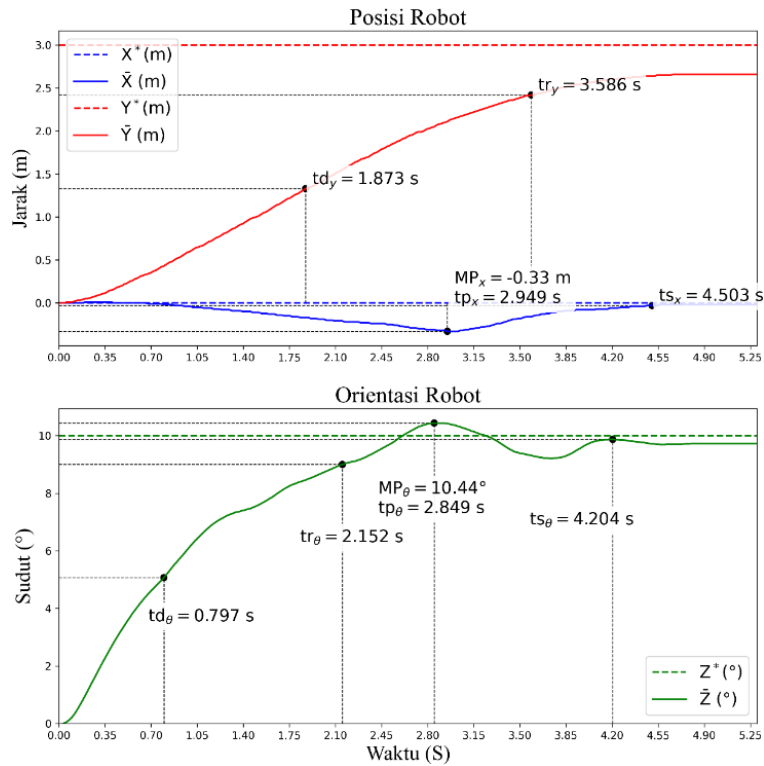
Motor ke-i	$t_d$	$t_r$	$t_s$	$t_p$	$c(t_p)$	$M_p$	$e_{ss}$
1	1,741 ms	2,048 ms	4,097 ms	2,458 ms	511 RPM	2,2%	1,5%
2	1,741 ms	2,048 ms	4,097 ms	2,458 ms	510 RPM	2%	1,6%
3	1,741 ms	2,151 ms	4,097 ms	2,458 ms	509 RPM	1,8%	1,2%
4	1,741 ms	2,048 ms	4,097 ms	2,458 ms	511 RPM	2,2%	1,3%
5	1,741 ms	2,048 ms	4,097 ms	2,458 ms	513 RPM	2,6%	1,4%
6	1,741 ms	2,048 ms	4,097 ms	2,458 ms	507 RPM	1,4%	1,5%

Persentase *maximum overshoot* ( $M_p$ ) dari enam motor memiliki nilai terbesar 2,6% dan nilai terkecil 1,4%. Sedangkan persentase *error steady-state* ( $e_{ss}$ ) terbesar bernilai 1,6% dan terkecil 1,2%. Hasil respon enam motor dapat dikatakan stabil karena dapat berada di bawah jangkauan toleransi dari *error steady-state*, yakni 3% hingga 5% dari nilai *setpoint*.

### 3.3 Pengujian Pergerakan Robot Menuju Posisi dan Orientasi Tujuan

Pengujian ini dilakukan untuk mengetahui lintasan yang dihasilkan robot saat bergerak dari posisi dan orientasi awal (0 m, 0 m, 0°) menuju posisi dan orientasi tujuan (0 m, 3 m, 10°). Permodelan matematis dari *inverse velocity kinematics* dan *forward velocity kinematics* yang telah dirancang akan diterapkan pada pengujian ini untuk mengetahui respon pergerakan robot.





GAMBAR 7: RESPON PERGERAKAN POSISI DAN ORIENTASI ROBOT

Berdasarkan grafik respon pergerakan posisi dan orientasi robot pada Gambar 7, dapat diketahui karakteristik respon pergerakan posisi dan orientasi robot pada *domain* waktu yang dijabarkan pada Tabel VI berikut ini. Posisi dan orientasi tersebut dapat direpresentasikan sebagai vektor, untuk mengetahui *error* vektor pada pengujian ini. Pada Tabel VII berikut ini merupakan *error* vektor pada robot yang dihasilkan pada pengujian ini.

TABEL VI: RESPON KARAKTERISTIK DOMAIN WAKTU PADA PERGERAKAN POSISI DAN ORIENTASI ROBOT

Parameter	$t_d$	$t_r$	$t_s$	$t_p$	$c(t_p)$	$M_p$	$e_{ss}$
$x$	-	-	4,503 s	2,949 s	-0,33 m	-0,33%	-0,02%
$y$	1,873	3,586 s	$\infty$	-	-	0%	11,33%
$z$	0,797	2,152 s	4,204 s	2,849 s	10,44°	4,4%	2,6%

TABEL VII: ERROR VEKTOR POSISI DAN ORIENTASI ROBOT

Posisi dan Orientasi Robot	Tujuan ( $X^*$ )	Posisi Akhir ( $\bar{X}$ )	$ X^* - \bar{X} $
$x$	0 m	-0,0205 m	0,0205 m
$y$	3 m	2,66 m	0,34 m
$z$	10°	9,74°	0,26°
$\frac{ X^* - \bar{X} }{ X^* } \times 100\%$			8,62%

#### 4. KESIMPULAN

Berdasarkan perancangan dan hasil pengujian yang telah dilakukan, dapat diambil kesimpulan terkait penelitian antara lain sebagai berikut:

1. Pada pengujian pembacaan sensor IMU BNO055 dengan melakukan integral kecepatan sudut pada sumbu- $z$  menjadi sudut *yaw*, nilai *error* rata-rata yang dihasilkan senilai 0,914%. Pada pengujian pembacaan *rotary encoder*



- eksternal untuk mendapatkan posisi robot pada sumbu- $x$  memiliki nilai rata-rata *error* terbesar senilai 0,97%. Sedangkan pada sumbu- $y$  memiliki nilai rata-rata *error* terbesar senilai 1,46%.
2. Pengujian *low-level control* pada enam motor dengan menggunakan kontroler PI, didapatkan nilai *error steady-state* terbesar senilai 1,6%. Respon pada enam motor memiliki nilai *maximum overshoot* ( $M_p$ ) terbesar 2,6%, yang mana dapat dikatakan stabil karena dapat berada di bawah jangkauan toleransi *error steady-state* ( $e_{ss}$ ).
  3. Hasil pengujian pergerakan robot menuju posisi dan orientasi tujuan menunjukkan bahwa pergerakan robot pada sumbu- $z$  menghasilkan *error steady-state* ( $e_{ss}$ ) senilai 2,6%. Sedangkan *error steady-state* ( $e_{ss}$ ) pada sumbu- $x$  senilai -0,33% dan sumbu- $y$  senilai 11,33%, yang mana dipengaruhi saat robot mengalami *overshoot* pada sumbu- $z$  sebesar 4,4%.

## DAFTAR PUSTAKA

- [1] Wahab, F., Zaskie, C., & Arthaya, B. M. (2021). Kendali Kecepatan Robot Beroda Omni dengan Kemampuan Menuju Posisi dan Orientasi yang Diinginkan Berbasis Pengendali PID. *Jurnal Teknologi Rekayasa*, 6(2), 273-284. <http://dx.doi.org/10.31544/jtera.v6.i2.2021.273-284>
- [2] Hijikata, M., Miyagusuku, R., & Ozaki, K. (2022). Wheel Arrangement of Four Omni Wheel Mobile Robot for Compactness. *Appl. Sci.*, 12(12), Artikel 5798. doi: 10.3390/app12125798
- [3] Kuppuswamy, N. S., Cho, S., Stonier, D., Choi, S., & Kim, J. (2014). DESIGN OF AN OMNIDIRECTIONAL ROBOT FOR FIRA ROBOSOT. Department of Electrical Engineering and Computer Science. Korea Advanced Institute of Science and Technology (KAIST).
- [4] Loh, W. K., Low, K. H., & Leow, Y. P. (2003). Mechatronics Design and Kinematic Modelling of a Singularityless Omni-Directional Wheeled Mobile Robot. 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422) (pp. 3237-3242). Institute of Electrical and Electronics Engineers (IEEE). doi: 10.1109/ROBOT.2003.1242089
- [5] Sirajuddin, I., Azhar, G. A., Wibowo, S., Ronilaya, F., Rahmad, C., & Rohadi, E. (2021). A General Inverse Kinematic Formulation and Control Scheme for Omnidirectional Robots. *Engineering Letters*, 29(4).
- [6] Wahab, F., Zaskie, C., & Arthaya, B. M. (2021). Kendali Kecepatan Robot Beroda Omni dengan Kemampuan Menuju Posisi dan Orientasi yang Diinginkan Berbasis Pengendali PID. *Jurnal Teknologi Rekayasa*, 6(2), 273-284. <http://dx.doi.org/10.31544/jtera.v6.i2.2021.273-284>
- [7] Sen, Z., Lei, S., Zhongliang, C., Lishuang, Z., & Jingtai, L. (2015). A ROS-based Smooth Motion Planning Scheme for a Home Service Robot. In Q. Zhao, et al., *Proceedings of the 34th Chinese Control Conference* (pp. 5119-5124). Institute of Electrical and Electronics Engineers (IEEE). doi: 10.1109/ChiCC.2015.7260438
- [8] Mahtani, A., Sánchez, L., Fernández, E., & Martínez, A. (2016). *Effective Robotics Programming with ROS* (3rd ed.). Packt Publishing Ltd.
- [9] Selvaraj, M., Maganaris, C. N., Baltzopoulos, V., Cullen, J., Kot, P., Shaw, A., & O'Brien, T. (2018). Stair Fall Risk Detection Using Wearable Sensors. 2018 11th International Conference on Developments in eSystems Engineering (DeSE) (pp. 108-112). Institute of Electrical and Electronics Engineers (IEEE). doi: 10.1109/DeSE.2018.00023
- [10] Utama, K. V., Fatekha, R. A., Prayoga, S., Pamungkas, D. S., & Hudhajanto, R. P. (2018). Positioning and Maneuver of an Omnidirectional Robot Soccer. 2018 International Conference on Applied Engineering (ICAE) (pp. 1-5). Institute of Electrical and Electronics Engineers (IEEE). doi: 10.1109/incae.2018.8579148
- [11] Finney, R. L., Demana, F. D., Waits, B. K., & Kennedy, D. (2012). *Calculus: Graphical, numerical, Algebraic* (4th ed.). Prentice Hall.
- [12] Sirajuddin, I., Azhar, G. A., Murdani, A., & Faizin, M. (2020). Desain dan Pemodelan Kontrol Kinematik pergerakan robot beroda dengan menggunakan 6 roda omni-wheels. *JURNAL ELTEK*, 18(1), 116-127. doi: 10.33795/eltek.v18i1.226

