

Sistem *Positioning* Robot Four-Omnidirectional Wheels Menggunakan MPU6050 dan *Encoder* Berbasis PD-Fuzzy Controller

Nur Holis
Teknik Elektro
Universitas Negeri Surabaya
Surabaya, Indonesia
nurkholis.19013@mhs.unesa.ac.id

Muhammad Syariffuddien Zuhrie
Teknik Elektro
Universitas Negeri Surabaya
Surabaya, Indonesia
zuhrie@unesa.ac.id

Endryansyah
Teknik Elektro
Universitas Negeri Surabaya
Surabaya, Indonesia
endryansyah@unesa.ac.id

Nurhayati
Teknik Elektro
Universitas Negeri Surabaya
Surabaya, Indonesia
nurhayati@unesa.ac.id

Diterima : Mei 2023
Disetujui : Juni 2023
Dipublikasi : Juli 2023

Abstrak—Robot beroda *four-omnidirectional wheels* merupakan robot *autonomous* yang dapat bergerak ke arah mana saja tanpa mengubah arah hadap robot tersebut. Sehingga tantangan utama dari robot ini yaitu sistem *positioning* yang hingga saat ini belum terdapat solusi absolut untuk mengatasi permasalahan tersebut. Oleh karena itu dibutuhkan sebuah sistem kendali yang bisa memperbaiki respons dan mengurangi terjadinya *positioning error* robot. Pada penelitian ini menggunakan kombinasi dua *controller* yaitu PD Controller dan *fuzzy logic controller* dengan algoritma *switching controller* yang diolah pada *software* STM32CubeIDE. Selain itu, pergerakan robot beroda *four-omnidirectional wheels* menggunakan *inverse kinematika* sehingga robot dapat menuju titik koordinat yang ditentukan dengan baik dan cepat. Sistem *positioning* pada penelitian ini menggunakan sensor MPU6050 sebagai pembaca orientasi hadap robot, dan sensor *rotary encoder* sebagai pembaca posisi robot di lapangan. Pengujian sistem *positioning* ini dilakukan dengan membandingkan antara *positioning* tanpa *controller*, *Positioning Fuzzy Logic Controller*, *Positioning PD Controller* dan *Positioning* dengan PD-Fuzzy controller pada pergerakan *home to point*, *point to point*, dan *point to home*. Diperoleh sistem *Positioning* dengan PD-Fuzzy controller menjadi yang terbaik dengan rata-rata eror 2,33 cm terhadap sumbu x dan 1,33 cm terhadap sumbu y, serta membutuhkan waktu 3,09 detik untuk mencapai *setpoint*.

Kata Kunci—*Positioning robot*, *Four Omnidirectional-Wheels*, *PD-Fuzzy Controller*

Abstract-- A *four-omnidirectional wheels* robot is an *autonomous* robot that can move in any direction without changing the direction it is facing. So that the main challenge of this robot is the *positioning* system which until now there is no absolute solution to overcome this problem. Therefore we need a control system that can improve responses and reduce the occurrence of robot *positioning* errors. In this research, a combination of two controllers was used, namely the PD controller and the fuzzy logic controller with a switching controller algorithm that was processed in the STM32CubeIDE software. In addition, the movement of the *four-omnidirectional wheels* robot uses *inverse kinematics* so that the robot can go to the coordinates that are

determined properly and quickly. The positioning system in this study uses the MPU6050 sensor as a reader for the orientation of the robot, and a rotary encoder sensor as a reader for the position of the robot in the field. Testing of the positioning system is carried out by comparing positioning without a controller, Positioning Fuzzy Logic Controller, Positioning PD Controller and Positioning with PD-Fuzzy controller on home to point, point to point and point to home movements. The Positioning system with PD-Fuzzy controller is the best with an average error of 2.33 cm on the x-axis and 1.33 cm on the y-axis, and takes 3.09 seconds to reach the setpoint.

Keywords—*Positioning robot*, *Four Omnidirectional-Wheels*, *PD-Fuzzy Controller*

I. PENDAHULUAN

Robotika merupakan bidang yang terdiri dari multidisiplin ilmu seperti mekanika konstruksi robot, sistem elektronika dan kontrol hingga kecerdasan buatan [1]. Awal penerapannya robotika banyak dimanfaatkan dalam dunia industri sehingga sering disebut istilah robot industri. Hadirnya robot industri ini dapat membantu bahkan menggantikan tenaga manusia sehingga suatu pekerjaan dapat dilakukan dengan lebih aman dan efisien [2]. Dalam perkembangannya, robot industri yang mulanya berupa robot lengan yang umumnya digunakan untuk memindahkan barang, perakitan produk dan lain sebagainya. Namun, robot lengan ini hanya bisa bergerak dan berputar pada *base*-nya sehingga terciptalah *mobile robot* yang dapat berpindah tempat dengan menggunakan roda untuk pergerakannya [3].

Keberadaan *mobile robot* semakin populer semenjak penggunaan roda konvensional sebagai penggeraknya digantikan oleh roda *omnidirectional* yang memungkinkan *mobile robot* untuk bermanuver dengan lebih leluasa dan cepat [4]. Hal inilah yang menyebabkan *mobile robot* dengan *omnidirectional wheels* atau yang biasa disebut robot beroda lebih banyak diminati dan digunakan di berbagai bidang karena kelebihan dan mobilitasnya yang lebih tinggi untuk membantu berbagai pekerjaan manusia. Dengan sistem

autonomous, robot beroda semakin canggih karena dapat mengambil keputusan secara otonom tanpa digerakkan atau dikendalikan secara manual oleh operator [5]. Maka dengan semakin canggih inovasi hadirnya *autonomous mobile robot* atau robot beroda ini dibarengi juga dengan semakin sulitnya kontrol yang harus dirancang dan diterapkan.

Tantangan utama yang dihadapi ketika merancang *autonomous mobile robot* yaitu *positioning* atau pemosisian robot. Robot akan kesulitan menentukan keputusan selanjutnya jika robot tersebut tidak mengetahui posisinya [6]. Penelitian dan survei literatur untuk pemosisian *mobile robot* telah banyak dilakukan. Namun, hingga saat ini tidak ada solusi absolut yang benar-benar mengatasi masalah pemosisian robot ini. Karena kurangnya sebuah metode yang baik, pengembang robot akhirnya mengombinasikan dua metode atau lebih sehingga didapatkan error yang lebih kecil [7]. Metode kontrol yang tepat sangat diperlukan untuk memperoleh akurasi sistem yang tinggi dalam kondisi yang tidak beraturan seperti gesekan, media yang tidak rata serta gangguan lainnya [8]. Menurut Kris N. (2022) dalam penelitiannya yang telah mengombinasikan dua metode kontrol yaitu kontrol PI dan FLC (*Fuzzy Logic Controller*) dengan hasil yang cukup baik. Namun, respons sistem pergerakan yang dihasilkan robot terlalu cepat dan agresif. Hal ini dikarenakan penggunaan kontrol PI yang karakteristiknya agresif sehingga mengakibatkan perangkat keras robot akan lebih cepat mengalami kerusakan [9].

Dengan latar belakang tersebut, didapatkan pengoptimalan kebutuhan *positioning* robot beroda menggunakan MPU6050 dan *rotary encoder* berbasis PD-*fuzzy controller*, karena dengan karakteristik respons kontrol PD yang lebih halus dan cepat [10]. Selanjutnya dikombinasikan *Fuzzy Logic Controller* untuk pembacaan error besar dengan pengaturan IF THEN pada *rules*-nya [11]. Kontrol FLC ini juga telah banyak digunakan di berbagai kendali sistem [12]. Selain itu dengan penggunaan sensor MPU6050 yang memiliki fitur *Gyroscope* dan *Accelerometer* serta dikombinasikan sensor *Rotary Encoder* diharapkan menghasilkan hasil error yang lebih kecil dengan kondisi lapangan yang tak menentu. Maka dari itu tujuan penelitian ini yaitu (1) Merancang sistem *positioning* robot beroda dengan *four omnidirectional wheels* menggunakan sensor MPU6050 dan *Rotary Encoder*. (2) Mengaplikasikan PD-*Fuzzy Controller* pada robot beroda dengan *four omnidirectional wheels*. (3) Mengetahui hasil respons sistem *positioning* robot beroda dengan *four omnidirectional wheels* menggunakan MPU6050 dan *rotary encoder* berbasis PD-*Fuzzy Controller*.

II. METODE

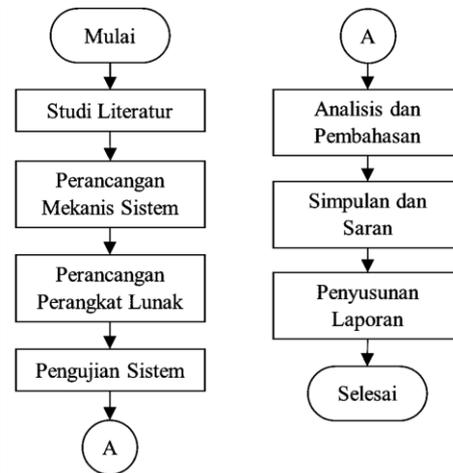
A. Pendekatan Penelitian

Berdasarkan pendekatannya, penelitian ini menggunakan metode penelitian kuantitatif dengan tujuan agar peneliti dapat mengembangkan sebuah teori, model matematis dan ide dasar yang berkaitan untuk mendapatkan sesuatu lebih baik dan bersifat baru [10]. Pada penelitian ini, instrumen yang digunakan yaitu perangkat lunak STM32CubeIDE versi 1.8.0 sebagai master *microcontroller* dan Arduino IDE versi 2.0.3. sebagai *sleeve microcontroller* untuk memprogram dan mengolah data sensor MPU6050 untuk mengumpulkan dan menganalisis data yang terbaca oleh sensor-sensor. Selain itu dibutuhkan pula rol meteran yang digunakan untuk mengukur data faktual posisi robot.

Data yang dihasilkan untuk dianalisis yaitu berupa respons kecepatan dan keakuratan *positioning* robot beroda

B. Langkah-langkah Penelitian

Gambar 1 menunjukkan langkah-langkah penelitian yang dilaksanakan untuk mendapatkan data hasil *positioning* robot beroda. Langkah pertama dalam melaksanakan penelitian rancang bangun sistem *positioning* pada robot beroda ini yaitu melakukan studi literatur dengan mempelajari dan mengkaji penelitian yang berkaitan dengan penelitian ini. Perancangan mekanika robot meliputi pekerjaan desain rangka serta peletakan *plant* dan sensor yang digunakan. Sedangkan perancangan *software* dan *hardware* meliputi rangkaian elektronika robot dan perangkat lunak yang digunakan sebagai instrumen penelitian. Pada tahap pengujian alat inilah yang menjadi penentu apakah penelitian yang dilakukan akan berhasil atau tidak. Pengujian harus dilakukan dengan parameter dan variabel yang sesuai sehingga proses analisisnya memenuhi standar keilmuan yang ada. Analisis yang dilakukan berdasarkan hasil pengujian yang telah dilakukan. Lalu diambil sebuah kesimpulan berdasarkan hasil analisis serta harus linear dengan tujuan penelitian yang telah ditetapkan.



Gambar 1. Diagram Alir Tahapan Penelitian

C. Rancangan Penelitian

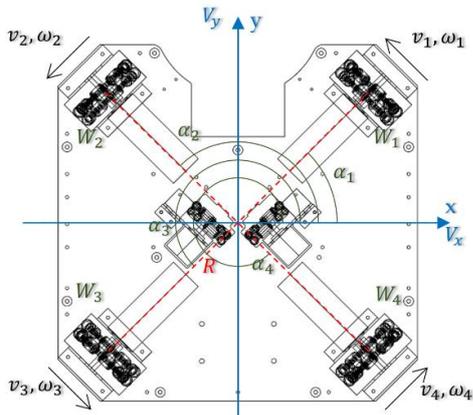
Rancangan Mekanis Robot

Robot beroda yang digunakan pada penelitian ini yaitu berjenis robot *holonomic* dengan penggerak motor dan roda *omnidirectional* sehingga robot dapat dengan mudah berpindah tempat tanpa mengubah arah hadapnya. Maka dari itu robot beroda ini memiliki empat roda *omnidirectional* yang digerakkan oleh masing-masing motor DC *Brushed* PG45. Roda *omnidirectional* yang dipakai berjenis *double omni* seperti pada Gambar 2.



Gambar 2. Double Omnidirectional Wheels

Selanjutnya roda *omnidirectional wheels* dikopel dengan motor DC *Brushed* PG45 dan disusun pada *base plate* robot dengan membentuk *x-frame* sehingga membentuk sudut masing-masing 45° terhadap sumbu x dan sumbu y. Jadi dengan model kinematika robot di bawah ini dapat ditentukan posisi robot untuk bergerak [13].



Gambar 3. Desain Sistem Robot Beroda dengan Four Omnidirectional Wheels

Penjelasan simbol desain sistem yang digambarkan dari Gambar 3 disajikan dalam Tabel 1.

TABEL 1. KETERANGAN SIMBOL DESAIN SISTEM

Simbol	Keterangan
$\alpha_1, \alpha_2, \alpha_3, \alpha_4$	Sudut antara sumbu referensi robot terhadap roda
$\omega_1, \omega_2, \omega_3, \omega_4$	Kecepatan sudut roda
v_1, v_2, v_3, v_4	Kecepatan linear roda
W_1, W_2, W_3, W_4	Roda omnidirectional
R	Jarak antara pusat robot dengan roda
v_y	Kecepatan linear robot pada sumbu y
v_x	Kecepatan linear robot pada sumbu x

Sehingga kecepatan linear robot terhadap *base plate* robot diurai dalam persamaan berikut.

$$(x, y, \theta) \quad (1)$$

$$V_x(t) = \frac{dx(t)}{dt}; V_y(t) = \frac{dy(t)}{dt}; \omega(t) = \frac{d\theta(t)}{dt} \quad (2)$$

$$\begin{bmatrix} V_1(t) \\ V_2(t) \\ V_3(t) \\ V_4(t) \end{bmatrix} = \begin{bmatrix} -\sin \alpha_1 & \cos \alpha_1 & 1 \\ -\sin \alpha_2 & \cos \alpha_2 & 1 \\ -\sin \alpha_3 & \cos \alpha_3 & 1 \\ -\sin \alpha_4 & \cos \alpha_4 & 1 \end{bmatrix} \begin{bmatrix} V_x(t) \\ V_y(t) \\ \omega(t) \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} V_1(t) \\ V_2(t) \\ V_3(t) \\ V_4(t) \end{bmatrix} = \begin{bmatrix} -\sin \alpha_1 \cdot V_x(t) + \cos \alpha_1 \cdot V_y(t) + \omega(t) \\ -\sin \alpha_2 \cdot V_x(t) + \cos \alpha_2 \cdot V_y(t) + \omega(t) \\ -\sin \alpha_3 \cdot V_x(t) + \cos \alpha_3 \cdot V_y(t) + \omega(t) \\ -\sin \alpha_4 \cdot V_x(t) + \cos \alpha_4 \cdot V_y(t) + \omega(t) \end{bmatrix} \quad (4)$$

Karena,

$$\alpha_1 = 45^\circ; \alpha_2 = 135^\circ; \alpha_3 = 225^\circ; \alpha_4 = 315^\circ \quad (5)$$

Maka,

$$V_1(t) = -0,707V_x(t) + 0,707V_y(t) + \omega(t) \quad (6)$$

$$V_2(t) = -0,707V_x(t) - 0,707V_y(t) + \omega(t) \quad (7)$$

$$V_3(t) = 0,707V_x(t) - 0,707V_y(t) + \omega(t) \quad (8)$$

$$V_4(t) = 0,707V_x(t) + 0,707V_y(t) + \omega(t) \quad (9)$$

Berdasarkan Persamaan 1 hingga Persamaan 9, maka dengan memberikan perintah data kecepatan linear robot pada lapangan (x, y, θ) dapat menggerakkan keempat motor sekaligus. Dalam desain sistem ini, menggunakan perancangan sistem *gyrodometry* untuk memperkirakan posisi relatif robot terhadap lokasi awal yang ditentukan [14].

Sensor orientasi hadap robot (θ) didapatkan dari sensor MPU6050 [15]. Serta *encoder* yang terdapat pada keempat motor dan dua *rotary encoder* eksternal memberikan pembacaan jarak pergerakan robot. *Encoder* motor PG45 membaca pergerakan putaran roda *omnidirectional* yang

terpasang seperti pada Gambar 3 di atas. Sehingga didapatkan persamaan *odometry* sebagai berikut.

$$X = \left(\frac{1}{2}\right) (r_4 - r_2) \cos 45^\circ + \left(\frac{1}{2}\right) (r_1 - r_3) \sin 45^\circ \quad (10)$$

$$Y = \left(\frac{1}{2}\right) (r_4 - r_2)(-\sin 45^\circ) + \left(\frac{1}{2}\right) (r_1 - r_3) \cos 45^\circ \quad (11)$$

$$Z = \text{pembacaan MPU6050} \quad (12)$$

Keterangan :

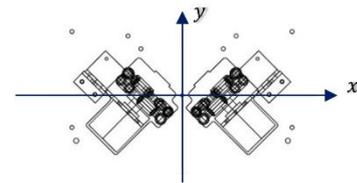
X = pembacaan jarak pada sumbu x (m)

Y = pembacaan jarak pada sumbu y (m)

Z = orientasi hadap robot ($^\circ$)

r_1, r_2, r_3, r_4 = pembacaan jarak pada setiap *encoder* motor

Sedangkan *rotary encoder* eksternal yang berperan untuk meminimalkan eror ketika roda slip atau tergelincir dirancang sesuai dengan Gambar 4.



Gambar 4. Posisi Rotary Encoder Eksternal

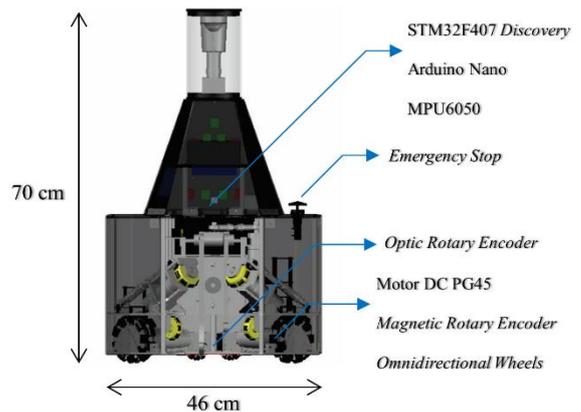
Berdasarkan rancangan posisi *rotary encoder* pada Gambar 4, didapatkan persamaan sebagai berikut.

$$X = \left(\frac{1}{2}\right) r_1 \cdot \cos 45^\circ + \left(\frac{1}{2}\right) r_2 \cdot \sin 45^\circ \quad (13)$$

$$Y = \left(\frac{1}{2}\right) r_1 \cdot (-\sin 45^\circ) + \left(\frac{1}{2}\right) r_2 \cdot \cos 45^\circ \quad (14)$$

$$Z = \text{pembacaan MPU6050} \quad (15)$$

Adapun spesifikasi umum dan ilustrasi robot beroda dengan *four omnidirectional wheels* ditunjukkan pada Gambar 5 di bawah ini dengan panjang 46 cm, lebar 46 cm, tinggi robot 77 cm serta massa robot 37 kg.

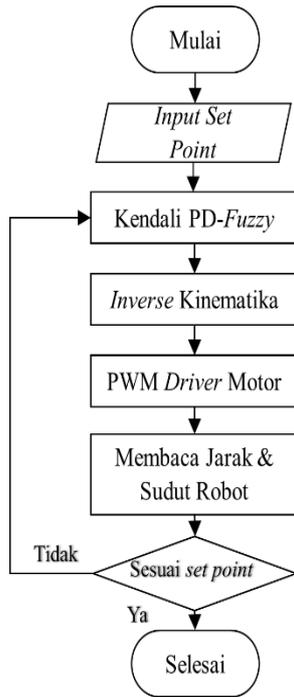


Gambar 5. Robot Beroda dengan Four Omnidirectional Wheels

Rancang Bangun Software

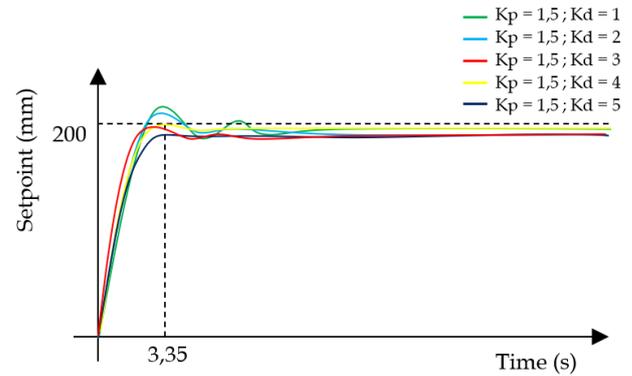
Perangkat lunak (*software*) utama yang digunakan dalam penelitian ini yaitu STM32CubeIDE. STM32CubeIDE menjadi *software* untuk memrogram (*editing, compiling, uploading/flashing*), serta pengolahan dan analisis data dilakukan pada STM32F407G, MCU pada *Discovery Board*. Tahapan proses tersebut ditunjukkan pada diagram alir Gambar 6.

Dalam mengolah data *input* tersebut, STM32F407 *Discovery* memproses *gyrodometry* pada data MPU6050 dan *encoder*, serta memproses *inverse kinematika* dan *PD-Fuzzy Controller* pada motor penggeraknya dengan mengirimkan nilai PWM pada *driver* motor [16]. Kemudian robot bergerak menuju *set point* yang telah ditentukan yang terbaca oleh *encoder* dan MPU6050 serta ditampilkan pada layar LCD. Apabila posisi akhir robot belum sesuai dengan *set point*, maka akan terdeteksi sebagai *error* dan *delta error*. Nilai *error* tersebut diproses kembali oleh *PD-Fuzzy Controller* dan mengirim perintah kembali berupa nilai PWM untuk *driver* motor. Proses ini dilakukan berulang-ulang hingga posisi robot sesuai dengan *set point*.



Gambar 6. Flowchart Pemrosesan Sistem Positioning

Konfigurasi nilai kendali PD harus diatur terlebih dahulu agar PD *Controller* dapat memberikan kendali yang baik dan sesuai dengan kebutuhan sistem *positioning* robot beroda [17]. Metode *tuning* PD *Controller* yang diterapkan pada penelitian ini yaitu metode *trial and error* dengan menguji coba nilai PD *Controller* pada robot secara aktual berdasarkan observasi dan pengalaman sehingga didapatkan nilai PD *Controller* yang sesuai [18]. *Tuning* PD dilakukan secara manual yaitu dengan (1) Memberi masukan nilai parameter *gain proportional* (K_p) dengan mengabaikan parameter *derivative* (K_d) terlebih dahulu. Pemberian parameter K_p ini mengakibatkan robot berosilasi. (2) Jika telah mendapatkan nilai parameter K_p yang tepat, selanjutnya bisa menentukan nilai parameter *derivative* (K_d) hingga osilasi berkurang atau tidak berosilasi lagi. Hasil *tuning* PD *controller* didapatkan nilai $K_p = 1,5$ dan $K_d = 5$ karena memiliki *respons* yang terbaik seperti yang ditunjukkan oleh Gambar 7.



Gambar 7. Hasil Respons Tuning PD Controller

Fuzzy logic controller melakukan proses pembacaan nilai *error* dan *delta error*. *Fuzzy logic controller* bekerja ketika *error* yang dihasilkan besar, sedangkan PD *Controller* bekerja ketika *error* yang ada bernilai kecil. Proses pembagian peran antara PD *controller* dengan *fuzzy logic controller* tersebut maka diperlukan algoritma *switching* seperti pada Persamaan 16 [9].

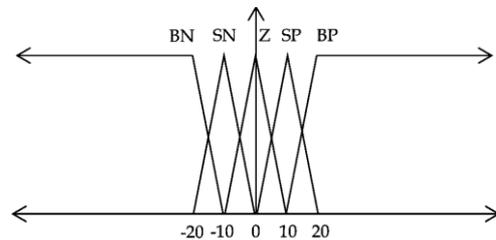
$$switching = \begin{cases} s \geq \varepsilon, & \text{fuzzy logic controller} \\ s \leq \varepsilon, & \text{PD controller} \end{cases} \quad (16)$$

Keterangan :

s = *error* posisi robot

ε = ambang batas nilai untuk *switch*

Pada *fuzzy logic controller*, dibutuhkan *membership function* yang ditunjukkan dalam Gambar 8 dalam proses pembacaan dan pengolahan *error positioning* robot.



Gambar 8. Membership Function Delta Error Positioning

$$f_{BN}(x) = \begin{cases} 1, & x < -20 \\ \frac{-10-x}{10}, & -20 \leq x \leq -10 \\ 0, & x > -10 \end{cases} \quad (17)$$

$$f_{SN}(x) = \begin{cases} 0, & x < -20 \\ \frac{x+20}{10}, & -20 \leq x \leq -10 \\ \frac{-x}{-10}, & -10 \leq x \leq 0 \\ 0, & x > 0 \end{cases} \quad (18)$$

$$f_Z(x) = \begin{cases} 0, & x < -10 \\ \frac{x+10}{-10}, & -10 \leq x \leq 0 \\ \frac{10-x}{10}, & 0 \leq x \leq 10 \\ 0, & x > 10 \end{cases} \quad (19)$$

$$f_{SP}(x) = \begin{cases} 0, & x < -20 \\ \frac{x-10}{10}, & 0 \leq x \leq 10 \\ \frac{20-x}{10}, & 10 \leq x \leq 20 \\ 0, & x > 20 \end{cases} \quad (20)$$

$$f_{BP}(x) = \begin{cases} 0, & x < -20 \\ \frac{x-10}{10}, & 10 \leq x \leq 20 \\ 1, & x \geq 20 \end{cases} \quad (21)$$

Membership function ini diterjemahkan ke dalam bahasa pemrograman pada *software* STM32CubeIDE sebagai metode kendali kecepatan motor. Jika robot telah mencapai *setpoint*, maka robot akan berhenti. Sedangkan jika belum mencapai *setpoint*, maka akan terbaca sebagai delta eror (Δ eror) yang menjadi input untuk kendali *fuzzy logic controller*. Proses tersebut dilakukan berulang-ulang hingga robot mencapai *setpoint* yang telah ditentukan.

III. HASIL DAN PEMBAHASAN

A. Kalibrasi Sensor MPU6050

Sensor MPU6050 perlu dilakukan kalibrasi dan pengujian agar didapatkan nilai orientasi hadap robot beroda yang presisi. Pengujian dilakukan dengan memutar arah robot pada porosnya sesuai lalu membandingkan antara pengukuran arah hadap robot sesungguhnya dengan nilai yang dibaca oleh sensor MPU6050 pada tampilan LCD robot dan didapatkan rata-rata eror yang kecil berdasarkan Tabel 2 yaitu sebesar $1,05^\circ$.

TABEL 2. KALIBRASI DAN PENGUJIAN SENSOR MPU6050

Sudut	Pembacaan MPU6050	Eror
0°	0°	0°
45°	$46,7^\circ$	$1,7^\circ$
90°	$92,3^\circ$	$2,3^\circ$
135°	$136,9^\circ$	$1,9^\circ$
180°	$180,8^\circ$	$0,8^\circ$
225°	$223,6^\circ$	$1,4^\circ$
270°	$269,9^\circ$	$0,1^\circ$
315°	$314,6^\circ$	$0,4^\circ$
360°	$359,1^\circ$	$0,9^\circ$
Rata-rata eror		$1,05^\circ$

B. Kalibrasi sensor Rotary Encoder

Proses pengujian dan kalibrasi pembacaan posisi robot beroda oleh *rotary encoder* dilakukan agar didapatkan nilai posisi robot yang akurat pada lapangan sehingga eror atau ketidaksesuaian posisi dapat dikurangi bahkan dihilangkan. Untuk mendapatkan nilai eror posisi robot, dilakukan perbandingan antara nilai posisi koordinat x dan y yang terbaca oleh *rotary encoder* dengan pengukuran secara langsung posisi robot pada lapangan. Berdasarkan Tabel 3 didapatkan rata-rata eror sebesar 3,25 terhadap sumbu x dan y .

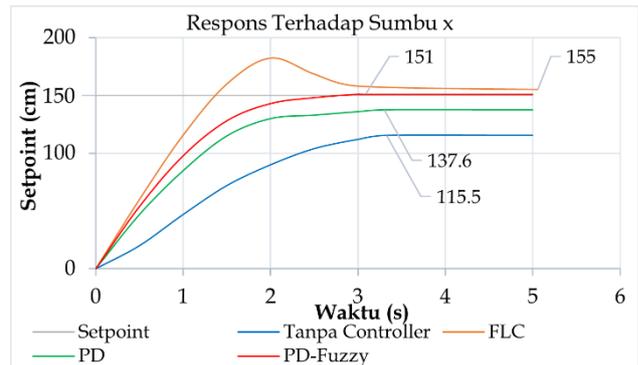
TABEL 3. HASIL PENGUJIAN DAN KALIBRASI ROTARY ENCODER

Koordinat (cm)		Nilai Rotary Encoder		Eror	
x	y	x	y	x	y
0	0	0	0	0	0
200	0	197	-4	3	4
0	200	4	204	4	4
300	450	306	455	6	5
Rata-rata eror				3,25	3,25

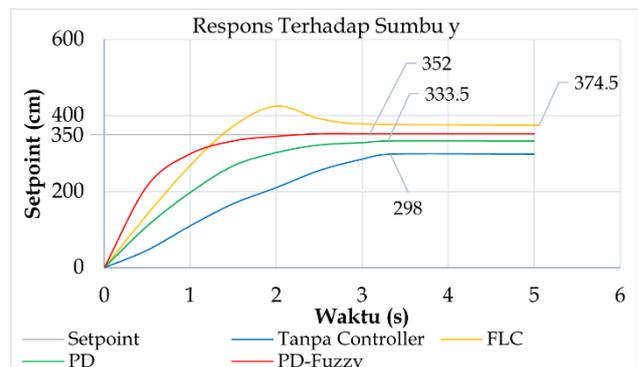
C. Pengujian Positioning Home to Point

Pengujian *positioning* ini bertujuan untuk mengetahui respons sistem *positioning* yang dimulai dari titik *Home* (0,0) menuju *point* yang sudah ditentukan yaitu (150,350). Pengujian dilakukan sebanyak empat kali yaitu (1) *Positioning* tanpa *controller*, (2) *Positioning* dengan *Fuzzy Logic Controller*, (3) *Positioning* dengan *PD-Controller*, (4) *Positioning* dengan *PD-Fuzzy Controller*. Grafik hasil respons *positining* robot beroda ditunjukkan oleh Gambar 9 terhadap sumbu x dan Gambar 10 terhadap sumbu y . Dan

secara detail nilai parameter yang didapatkan disajikan pada Tabel 4.



Gambar 9 Respons Sistem *Positing Home to Point* Terhadap Sumbu x



Gambar 10 Respons Sistem *Positing Home to Point* Terhadap Sumbu y

TABEL 4. HASIL RESPONS SISTEM *POSITIONING HOME TO POINT*

	Koordinat		Eror		Waktu (s)
	x	y	x	y	
I	115,5	298	34,5	52	3,33
II	137,6	333,5	12,4	16,5	5,06
III	155	374,5	5	24,5	3,30
IV	152	352	1	2	3,09

Keterangan:

- I = *Positioning* Tanpa *Controller*
- II = *Positioning Fuzzy Logic Controller*
- III = *Positioning PD-Controller*
- IV = *Positioning PD-Fuzzy Controller*

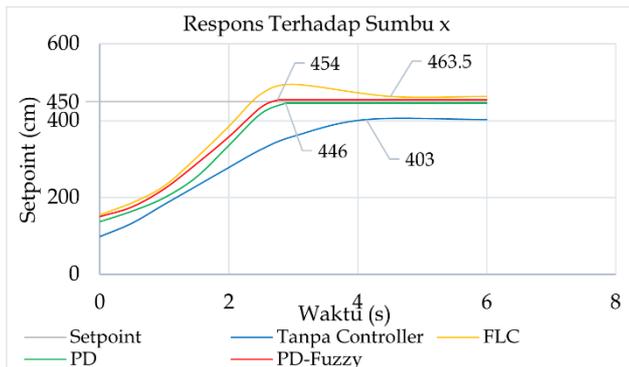
Berdasarkan Tabel 4, dapat disimpulkan bahwa respons sistem *positioning* terbaik yaitu pada pengujian *positioning* dengan *PD-Fuzzy Controller* karena memiliki eror paling kecil sebesar 1 cm terhadap sumbu x , dan 2 cm terhadap sumbu y . Selain itu pada *positioning* dengan *PD-Fuzzy Controller* menempuh waktu paling cepat yaitu 3,09 detik.

D. Pengujian Positioning Point to Point

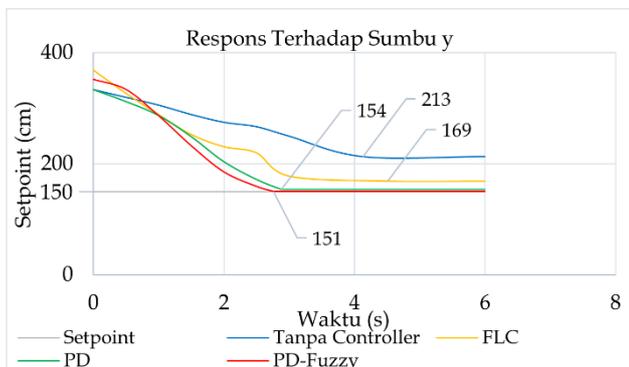
Pengujian *positioning* ini bertujuan untuk mengetahui respons sistem *positioning* yang dimulai dari titik *Point* (150,350) menuju *point* lainnya yang sudah ditentukan yaitu (450,150). Pengujian dilakukan sebanyak empat kali yaitu (1) *Positioning* tanpa *controller*, (2) *Positioning* dengan *Fuzzy Logic Controller*, (3) *Positioning* dengan *PD-Controller*, (4) *Positioning* dengan *PD-Fuzzy Controller*. Grafik hasil respons *positining* robot beroda ditunjukkan oleh Gambar 11 terhadap sumbu x dan Gambar 12 terhadap sumbu y . Dan secara detail nilai parameter yang didapatkan disajikan pada Tabel.

Berdasarkan Tabel 5, dapat disimpulkan bahwa respons sistem *positioning* terbaik yaitu pada pengujian *positioning*

dengan PD-Fuzzy Controller karena memiliki eror paling kecil sebesar 4 cm terhadap sumbu x, dan 1 cm terhadap sumbu y. Selain itu pada *positioning* dengan PD-Fuzzy Controller menempuh waktu paling cepat yaitu 2,76 detik.



Gambar 11 Respons Sistem *Positing Point to Point* Terhadap Sumbu x



Gambar 12 Respons Sistem *Positing Point to Point* Terhadap Sumbu y

TABEL 5. HASIL RESPONS SISTEM *POSITIONING HOME TO POINT*

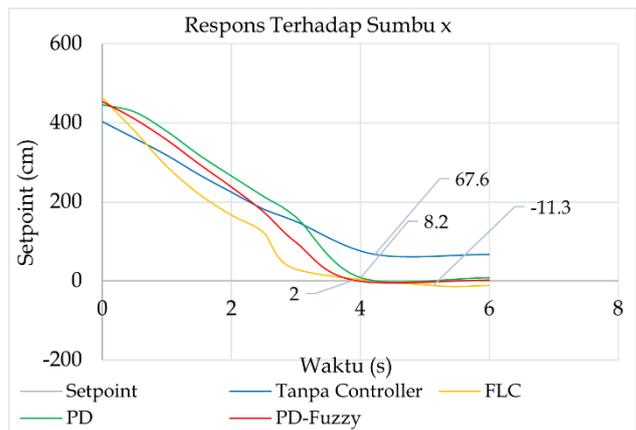
	Koordinat		Error		Waktu (s)
	x	y	x	y	
I	403	213	47	63	4,13
II	446	154	4	4	4,51
III	463,5	169	13,5	19	2,88
IV	454	151	4	1	2,76

Keterangan:

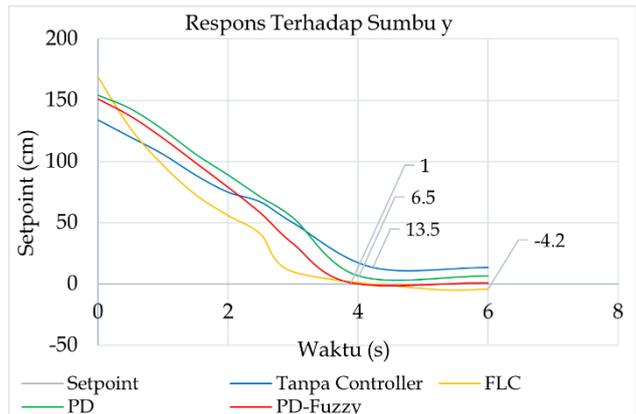
- I = *Positioning Tanpa Controller*
- II = *Positioning Fuzzy Logic Controller*
- III = *Positioning PD-Controller*
- IV = *Positioning PD-Fuzzy Controller*

E. Pengujian *Positioning Point to Home*

Pengujian *positioning* ini bertujuan untuk mengetahui respons sistem *positioning* yang dimulai dari titik *Point* (450,150) menuju *Home* atau posisi awal robot yaitu (0,0). Pengujian dilakukan sebanyak empat kali yaitu (1) *Positioning* tanpa controller, (2) *Positioning* dengan Fuzzy Logic Controller, (3) *Positioning* dengan PD-Controller, (4) *Positioning* dengan PD-Fuzzy Controller. Grafik hasil respons *positining* robot beroda ditunjukkan oleh Gambar 11 terhadap sumbu x dan Gambar 12 terhadap sumbu y. Dan secara detail nilai parameter yang didapatkan disajikan pada Tabel 6.



Gambar 13 Respons Sistem *Positing Point to Home* Terhadap Sumbu x



Gambar 14 Respons Sistem *Positing Point to Home* Terhadap Sumbu y

TABEL 6. HASIL RESPONS SISTEM *POSITIONING HOME TO POINT*

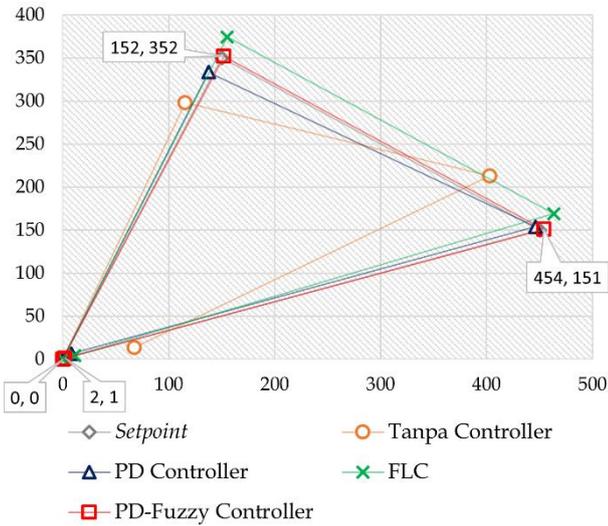
	Koordinat		Eror		Waktu (s)
	x	y	x	y	
I	67.6	13.5	67,6	13,5	4,23
II	8,2	6,5	8,2	6,5	5,13
III	11,3	4,2	11,3	4,2	4,01
IV	2	1	2	1	3,89

Keterangan:

- I = *Positioning Tanpa Controller*
- II = *Positioning Fuzzy Logic Controller*
- III = *Positioning PD-Controller*
- IV = *Positioning PD-Fuzzy Controller*

Berdasarkan Tabel 6, dapat disimpulkan bahwa respons sistem *positioning* terbaik yaitu pada pengujian *positioning* dengan PD-Fuzzy Controller karena memiliki eror paling kecil sebesar 2,33 cm terhadap sumbu x, dan 1,33 cm terhadap sumbu y. Selain itu pada *positioning* dengan PD-Fuzzy Controller menempuh waktu paling cepat yaitu 3,09 detik.

Dari hasil pengujian *positioning* yang telah dilaksanakan, perbandingan hasil titik koordinat *positioning* yang telah ditempuh oleh robot beroda antara tanpa controller, Fuzzy Logic Controller, PD-Controller dan PD-Fuzzy Controller disajikan dalam Gambar 15. Sehingga berdasarkan perbandingan tersebut, didapatkan sistem *positioning* dengan PD-Fuzzy Controller yang paling mendekati titik koordinat *setpoint* dibandingkan *positioning* dengan controller lain.



Gambar 15. Perbandingan Titik Koordinat Positioning

Selain itu, rata-rata error yang dihasilkan oleh tiap pengujian dijelaskan oleh persamaan sebagai berikut.

Rata-rata error Positioning Tanpa Controller

$$RMSE_x = \frac{34,5+47+67,6}{3} = \frac{149,1}{3} = 49,7 \quad (22)$$

$$RMSE_y = \frac{52+63+13,5}{3} = \frac{128,5}{3} = 42,83 \quad (23)$$

Rata-rata error Positioning Fuzzy Logic Controller

$$RMSE_x = \frac{12,4+4+8,2}{3} = \frac{24,6}{3} = 8,2 \quad (24)$$

$$RMSE_y = \frac{16,5+4+6,5}{3} = \frac{27}{3} = 9 \quad (25)$$

Rata-rata error Positioning PD Controller

$$RMSE_x = \frac{5+13,5+11,3}{3} = \frac{29,8}{3} = 9,93 \quad (26)$$

$$RMSE_y = \frac{24,5+19+4,2}{3} = \frac{47,7}{3} = 15,9 \quad (27)$$

Rata-rata error Positioning PD-Fuzzy Controller

$$RMSE_x = \frac{1+4+2}{3} = \frac{7}{3} = 2,33 \quad (28)$$

$$RMSE_y = \frac{2+1+1}{3} = \frac{4}{3} = 1,33 \quad (29)$$

Sedangkan rata-rata waktu yang ditempuh dalam satuan detik adalah sebagai berikut.

Rata-rata waktu yang ditempuh Positioning tanpa controller

$$\text{Rata-rata waktu} = \frac{3,33+4,13+4,23}{3} = \frac{11,69}{3} = 3,90 \quad (30)$$

Rata-rata waktu yang ditempuh Positioning Fuzzy Logic Controller

$$\text{Rata-rata waktu} = \frac{5,06+4,51+5,13}{3} = \frac{14,70}{3} = 4,9 \quad (31)$$

Rata-rata waktu yang ditempuh Positioning PD Controller

$$\text{Rata-rata waktu} = \frac{3,30+2,88+4,01}{3} = \frac{10,19}{3} = 3,40 \quad (32)$$

Rata-rata waktu yang ditempuh Positioning PD-Fuzzy Controller

$$\text{Rata-rata waktu} = \frac{3,09+2,76+3,89}{3} = \frac{9,74}{3} = 3,25 \quad (33)$$

IV. KESIMPULAN

Berdasarkan hasil pengujian respons sistem positioning robot beroda yang telah dilakukan, maka dapat disimpulkan bahwa sistem positioning robot beroda menggunakan sensor MPU6050 dan rotary encoder berbasis PD-Fuzzy Controller berjalan dengan baik serta robot dapat berpindah posisi dan mencapai setpoint yang telah ditentukan. Pada pengujian kalibrasi sensor MPU6050 yang berfungsi sebagai pembaca orientasi hadap robot, didapatkan rata-rata error sebesar $1,05^\circ$. Serta pengujian kalibrasi pembacaan posisi robot oleh rotary encoder diperoleh rata-rata error sebesar 3,25 cm terhadap sumbu x dan 3,25 cm terhadap sumbu y. Konfigurasi nilai PD controller yaitu $K_p = 1,5$ dan nilai $K_d = 5$ serta dikombinasikan dengan fuzzy logic controller dapat memberikan pengaruh positif dan menghasilkan positioning yang lebih baik dengan algoritma switching yang diolah pada instrumen perangkat lunak STM32CubeIDE. Hasil respons sistem positioning robot beroda dilakukan dengan pengujian pergerakan home to point, point to point dan point to home. Diperoleh positioning dengan PD-Fuzzy controller menjadi respons sistem terbaik setelah dibandingkan dengan respons positioning tanpa controller, Positioning Fuzzy Logic Controller, dan Positioning PD Controller. Positioning dengan PD-Fuzzy controller menghasilkan rata-rata error 2,33 cm terhadap sumbu x dan 1,33 cm terhadap sumbu y, serta membutuhkan waktu 3,25 detik untuk mencapai setpoint. Penelitian ini menghasilkan error yang lebih baik. Namun karena masih terjadinya error pada positioning maka penelitian ini masih belum bisa dijadikan solusi yang absolut. Maka dari itu, pengembangan metode positioning tetap dibutuhkan baik dengan menggunakan controller lain dan/atau mengombinasikan controller seperti Neural Network, Artificial Intelligence (AI) dan sebagainya. Selain itu, penggunaan sensor yang lebih banyak dan andal juga diharapkan dapat memperbaiki hasil positioning.

UCAPAN TERIMA KASIH

Terima kasih untuk keluarga besar tim DEWO ROBOTIC FT UNESA khususnya divisi BRATASENA TEAM Robot Sepak Bola Beroda yang telah menyediakan sarana prasarana serta keilmuan untuk pelaksanaan penelitian sistem positioning robot beroda ini.

REFERENSI

- [1] F. Umam, "Pengembangan Sistem Kendali Pergerakan Autonomous Mobile Robot Untuk Mendapatkan Jalur Bebas Hambatan Menggunakan Fuzzy Logic Controller," *J. Ilm. Mikrotek*, vol. 1, no. 1, pp. 35–42, 2013.
- [2] K. V. Ignatiev, M. M. Kopichev, and A. V. Putov, "Autonomous Omni-Wheeled Mobile Robots," *IEEE*, pp. 0–3, 2016.
- [3] S. G. Tzafestas, *Introduction to Mobile Robot Control*, 1st ed. Athens: National Technical University of Athens, 2013. doi: 10.1016/b978-0-12-417049-0.00020-1.
- [4] M. Rahman and H. Aprilianto, "Penerapan Metode Fuzzy Pada Robot Beroda Menggunakan Omni-Directional Wheels," *Jutisi*, vol. 5, no. 2, pp. 1075–1082, 2017.
- [5] G. Fracapane, R. de Koster, F. Sgarbossa, and J. O. Strandhagen, "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda," *Eur. J. Oper. Res.*, vol. 294, no. 2, pp. 405–426, 2021, doi: 10.1016/j.ejor.2021.01.019.
- [6] A. Rachmawan, "Penentuan Posisi Robot Sepak Bola Beroda Menggunakan Rotary Encoder dan Kamera," *Undergrad. thesis, Jur. Tek. Elektro, Fak. Teknol. Ind. Inst. Teknol. Sepuluh Nopember, Surabaya*, 2017.
- [7] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile Robot Positioning - Sensors and techniques," *J. Robot. Syst.*, vol. 14, no. 4, pp. 231–249, 1997.

- [8] R. H. Abiyev, I. S. Günsel, N. Akkaya, E. Aytac, A. Çağman, and S. Abizada, "Fuzzy control of omnidirectional robot," *Procedia Comput. Sci.*, vol. 120, pp. 608–616, 2017, doi: 10.1016/j.procs.2017.11.286.
- [9] K. Nurariyanto, "Sistem Positioning Pada Four-Wheeled Omnidirectional Mobile Robot Menggunakan Metode Gyrodometry Berbasis PI- Fuzzy Controller," *Tek. Elektro*, 2022.
- [10] M. Lamatenggo, I. Wiranto, and W. Ridwan, "Perancangan Balancing Robot Beroda Dua Dengan Metode Pengendali PID Berbasis Arduino Nano," *Jambura J. Electr. Electron. Eng.*, vol. 2, no. 2, pp. 39–43, 2020, doi: 10.37905/jjee.v2i2.6906.
- [11] I. H. Lahay, H. Hasanuddin, J. D. Giu, and M. G. Bawole, "Penentuan Grade Kopra Dengan Penerapan Metode Logika Fuzzy," *Jambura J. Electr. Electron. Eng.*, vol. 5, no. 1, pp. 122–129, 2023, doi: 10.37905/jjee.v5i1.17073.
- [12] W. Ridwan, I. Wiranto, L. Azzahra, and F. Lakoro, "Penentuan Uang Kuliah Tunggal Mahasiswa Universitas Negeri Gorontalo Berbasis Logika Fuzzy," *Jambura J. Electr. Electron. Eng.*, vol. 3, no. 2, pp. 62–65, 2021, doi: 10.37905/jjee.v3i2.10617.
- [13] J. Yun, J. Park, H. Choi, and J. Lee, "Absolute Positioning System for Mobile Robot Navigation in an Indoor Inveronment," *ICCAS*, no. Iccas, pp. 1448–1451, 2004.
- [14] S. E. Oltean, M. Dulau, and R. Puskas, "Position Control of Robotino Mobile Robot Using Fuzzy Logic," *IEEE Int. Conf. Autom. Qual. Testing, Robot. AQTR 2010 - Proc.*, vol. 1, pp. 366–371, 2010, doi: 10.1109/AQTR.2010.5520855.
- [15] B. Firman, "Implementasi Sensor IMU MPU6050 Berbasis Serial I2C Pada Self-Balancing Robot," *Juena Teknol. Technoscientia*, vol. 9, no. 1, pp. 18–24, 2016.
- [16] S. Setiawan, Firdaus, B. Rahmadya, and Derisma, "Penerapan Invers Kinematika Untuk Pergerakan Kaki Robot Biped," no. November, pp. 1–9, 2015.
- [17] K. Ogata, *Modern control engineering*, 5th ed., vol. 17, no. 3. Tokyo: Prentice Hall, 2010. doi: 10.1109/tac.1972.1100013.
- [18] N. T. Pasaribu and E. M. Sartika, "Wheeled Robot Based for Indonesian Intelligent Robot Contest Robot Beroda Untuk Kontes Robot Cerdas," no. 220148, 2012.