

## RANCANG BANGUN ROBOT OTONOM PEMINDAH BARANG BERODA DENGAN KONTROL ORIENTASI BERBASIS MPU6050

Harry Budi Saputra<sup>1</sup>, Pratomo Budi Santosa<sup>1</sup>

<sup>1</sup> Universitas Muhammadiyah Surakarta, Jl. A. Yani Tromol Pos 1, Pabelan,  
Kartasura, Sukoharjo, Indonesia

 Email korespondensi: [harrybudisaputra06@gmail.com](mailto:harrybudisaputra06@gmail.com)

**Abstrak.** Robot autonomous (otonom) merupakan robot dengan sistem kompleks yang membutuhkan interaksi dengan perangkat lunak. Robot autonomous dirancang untuk melakukan tugas tingkat tinggi dengan kontrol eksternal yang sangat terbatas. Penelitian ini dilakukan untuk merancang sistem pengendalian robot secara otonom dengan kontrol orientasi sebagai keseimbangan sudut robot dari kemiringan sehingga dapat menyelesaikan misi pada arena yang ada. Dalam rangka itu, penulis merancang dan mengimplementasikan robot otonom untuk kestabilan gerak dengan *Gyro* melalui pembacaan *error* kemiringan saat bergerak dengan roda mekanum. Sistem robot ini menggunakan sensor ultrasonik sebagai penyesuaian titik lokasi dari setiap *plan* yang dirancang antara titik gerak pengambilan dan titik taruh barang. Sistem otonom ini menggunakan Arduino Nano sebagai kontrol roda melalui *driver* L298N dan MPU6050 yang membaca nilai kemiringan pada robot dan dihitung dalam faktor koreksi berbentuk nilai *yaw*, lalu ESP32 sebagai kontrol bantu untuk mengolah hubungan antara dua mikroprosesor melalui *hardware serial* RX TX untuk mengirimkan pesan serial pemanggilan antara *plan* yang disusun pada saat robot bergerak. Penelitian ini mengevaluasi dan menerapkan faktor koreksi *yaw* dalam akurasi gerak roda mekanum yang sering mengalami pergeseran kemiringan yang dibaca dalam bentuk *error* dan perlu dikoreksi melalui perbandingan PID dan kinematic proporsi mekanik untuk mendapatkan gerakan yang halus dan mengurangi *error* pergeseran dilapangan. Penerapan ini menunjukkan akurasi *error* 80% dengan selisih jarak target *yaw* 180 dan 0 derajat, selisih *yaw input* pada nilai 178 dan 2 derajat.

**Kata kunci:** Orientasi, Gyro, Roda Mekanum, yaw, error, Otonom



## PENDAHULUAN

Robot otonom (*autonomous*) adalah robot dengan sistem kompleks yang membutuhkan interaksi dengan perangkat lunak seperti mikrokontroler. Robot otonom dirancang untuk melakukan tugas dengan kontrol otonom mikro sebagai otaknya. Robot otonom beroperasi di lingkungan yang sangat bervariasi, tidak pasti, dan berubah waktu, harus memenuhi interval waktu secara *real-time* untuk bekerja dengan baik [11].

Robot adalah seperangkat alat mekanik yang bisa melakukan kegiatan fisik dengan menggunakan program yang telah di buat oleh manusia, ataupun dengan diawasi atau dikontrol oleh manusia [7]. Kendaraan otonom dan robot bergerak mewakili kemajuan penting dalam bidang teknologi kecerdasan buatan modern, memungkinkan penerapan di banyak industri, termasuk manufaktur, pertanian, dan perawatan kesehatan. Mengembangkan sistem robot bergerak yang sepenuhnya otonom memerlukan integrasi beragam fungsi, mulai dari persepsi dan perencanaan lingkungan hingga pengambilan keputusan, pengendalian perilaku, dan pelaksanaan [12].

Penerapan penelitian ini menggunakan roda mekanum sebagai proporsi gerakan banyak arah untuk kemudahan gerak robot. Robot dengan empat roda mekanum adalah sistem yang digerakkan secara berlebihan dengan lebih dari satu batasan gerakan (kecepatan roda berkorelasi linier) [5].

Di lapangan dimungkinkan robot akan mengalami deformasi bentuk badan maupun roda akibat tekanan atau benturan. Untuk membuat robot tetap mudah dikendalikan maka dikembangkan sistem kemudi otomatis. Sistem kemudian akan dibantu untuk menjaga robot berada pada jalur yang sesuai sehingga memudahkan pengendalian. Sistem didasarkan pada orientasi gerak robot yang dibaca oleh sensor *gyroscope*. Kecepatan motor kanan dan kiri akan disesuaikan sehingga arah gerak robot akan tetap pada jalur lintasan [8]. Sensor *Gyro* memungkinkan robot bergerak dalam garis lurus atau memutar derajat tertentu dengan menghitung *yaw* nilai dari data mentahnya dan kecepatan kendali motor DC menggunakan pengontrol *P* [15].

Robot didesain secara otonom, lalu akan menerapkan sistem otonom dalam menyelesaikan misi di arena dengan menerapkan modeling kestabilan gerakan roda dengan melakukan pembacaan dari *gyroscope* melalui sudut kemiringan robot sehingga pembacaan kecepatan roda mekanum akan dapat dibaca kemiringannya. Kestabilan pergerakan roda mekanum dapat distabilkan dengan penyesuaian kecepatan putaran roda sehingga pada saat robot bergerak meminimalisir terjadinya *error* dari kemiringan robot saat bergerak.



## METODE

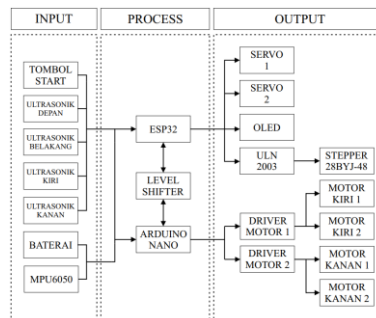
Pembuatan robot dengan penerapan *gyro* sebagai kestabilan dalam gerakan ini peneliti menggunakan susunan seperti yang tertera pada Gambar 1 yang menjelaskan proses pembuatan dan perancangan robot ini membutuhkan waktu serta pengujian yang baik agar didapatkan pengambilan data yang sesuai pada kondisi lapangan atau area simulasi.



Gambar 1. *Flowchart* Tahapan Penelitian

## Perancangan Sistem

Blok sistem menampilkan kategori cara kerja sistem dari *input*, proses dan *output* yang dilakukan pada alat. Dimana ketiga hal ini dibagi kedalam setiap fungsi kerja sistem seperti pada Gambar 2.

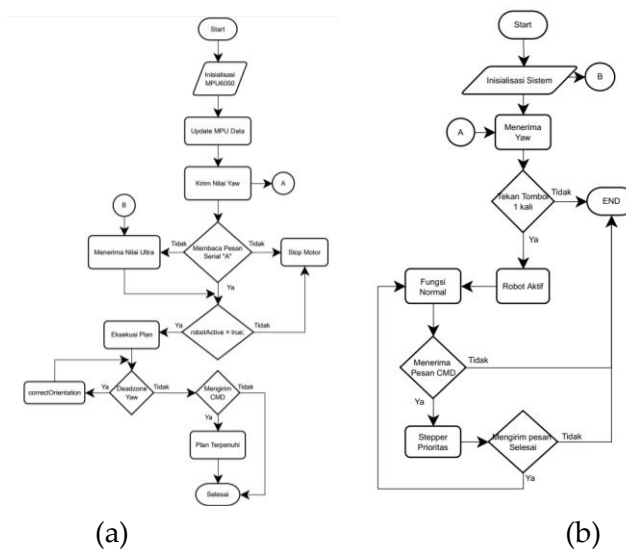


Gambar 2. Blok Diagram Sistem

Pada *Flowchart* sistem menampilkan alur kerja sistem yang berjalan pada robot. Proses dimulai dengan inisialisasi mikrokontroler dan sensor inputan yang digunakan, selanjutnya MPU mulai membaca nilai *yaw* untuk nilai orientasi yang kemudian dikirimkan ke ESP32 dan nilai ultrasonik akan dikirimkan ke Nano juga agar bisa dimonitoring melalui OLED. Setelahnya sistem akan menunggu serial menerima pesan "A" sebagai pesan mulai robot bergerak, saat ESP mengirimkan pesan "A" dan Nano menerima pesan akan masuk kedalam sistem robot berjalan, apabila menerima pesan lain



fungsi robot bergerak tidak bernilai benar sehingga hanya pembacaan sensor yang berjalan. Saat menerima pesan "A" robot aktif dan memulai eksekusi *plan* dari Nano dan ESP beralih pada fungsi normal, bersamaan dengan sistem bekerja apabila terdapat *deadzone yaw* maka robot akan koreksi orientasi secara otomatis dan jika tidak maka *plan* akan terus berjalan hingga mengirimkan pesan CMD ke ESP. Jika ESP menerima pesan CMD maka akan beralih ke fungsi stepper prioritas dan jika tidak maka langsung selesai. Setelah prioritas selesai kembali mengirim pesan selesai ke Nano dan kembali ke fungsi normal, jika tidak maka langsung selesai. Saat nano menerima pesan dari esp dari prioritas maka *plan* terpenuhi. Alur kerja sistem dapat dilihat pada Gambar 3 (a) dan Gambar 3 (b).

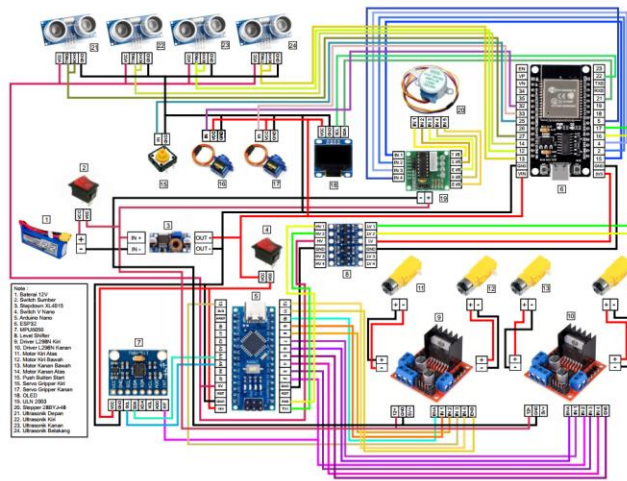


Gambar 3. Flowchart Sistem (a) Arduino Nano (b) ESP32

### Perancangan Elektrikal

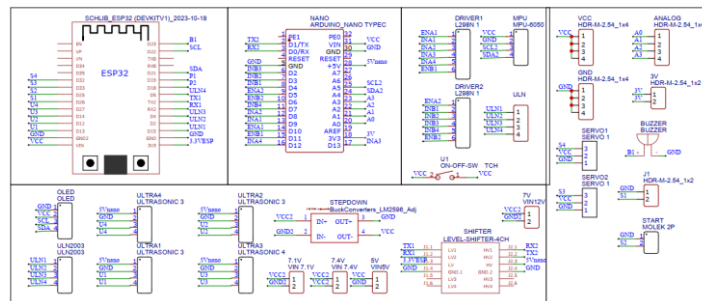
Pada tahap ini rancangan wiring diagram sistem pada Gambar 4 menampilkan komponen yang dijalurkan ke bagian-bagiannya. Dimana Arduino Nano dan ESP32 dilakukan hubungan *hardware serial* dengan menambahkan *level shifter* sebagai pengubah level tegangan dari RX TX. Sedangkan inputan posisi akan ditentukan melalui pembacaan ultrasonik berdasarkan jarak dalam jarak (cm) yang kemudian MPU6050 akan membaca orientasi kemiringan dari pergerakan yang dilakukan.





Gambar 4. Wiring Diagram Sistem

Elektrikal dibuat dalam bentuk PCB (Papan Circuit Cetak) yang didesain menggunakan *software* EasyEDA seperti yang ditampilkan pada Gambar 5 dengan merancang desain skematik untuk memberikan keterangan jalur hubungan antara komponen yang akan digunakan, setelah itu akan dibuat dalam bentuk *layout* untuk memberikan jalur pada cetakan tembaga PCB sehingga jalur garis dapat menghubungkan antara komponen satu dengan komponen yang lain.

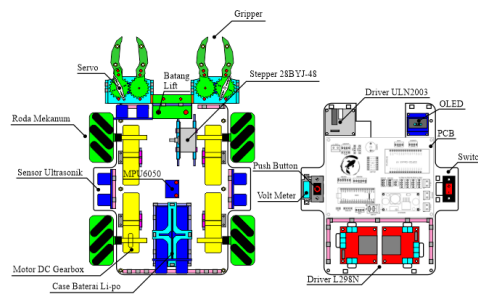


Gambar 5. Skematik PCB

### Perancangan Mekanikal

Tahapan ini merancang bentuk fisik robot dalam bentuk dua dimensi. Rancangan gambar akan memberikan kemudahan dalam merancang bangun robot untuk meminimalisir ketidaksesuaian ukuran dan untuk keperluan mencetak kerangka dengan akrilik yang dibuat seperti susunan per potong. Desain ini memiliki 2 lapisan, yaitu lapisan atas untuk kerangka utama penempatan roda dan lapisan atas untuk kerangka penempatan PCB dan komponen modul seperti pada Gambar 6.





Gambar 6. Desain 2D

## Pengujian Sistem

Setelah proses perancangan dan pembuatan robot selesai, tahap yang diperlukan selanjutnya adalah pengujian sistem robot. Tahap ini merupakan fase krusial dimana uji coba komponen, fitur, fisik dan program robot akan diuji secara keseluruhan operasional untuk memastikan bahwa robot sudah bekerja sesuai dengan spesifikasi dan perencanaan yang diharapkan kedalam konteks kestabilan roda mekanum saat bergerak. Pengujian meliputi observasi orientasi keseimbangan dan kemiringan robot saat bergerak maupun saat diam dengan mengamati nilai yang dikeluarkan oleh sensor *gyroscope* MPU6050 berupa *yaw*. Gerakan dilakukan berdasarkan pola gerak roda mekanum yang diintegrasikan dengan sensor orientasi *gyro* dengan pola maju, mundur, geser kanan, geser kiri, rotasi kiri, rotasi kanan, serong kiri maju, serong kiri mundur, serong kanan maju, dan serong kanan mundur lalu diambil berapa *error* yang dihasilkan dari pembacaan *yaw* target dengan *yaw input* apakah menampilkan perbedaan toleransi 0-3 derajat *yaw*. Pengujian selanjutnya dilakukan dengan perbandingan tanpa sensor *gyro* orientasi dengan pola gerakan yang sama sehingga akan diamati *error* yang dihasilkan. Pengujian ini dilakukan untuk penyempurnaan robot agar memiliki mekanisme orientasi otomatis saat nilai *yaw* target di set sehingga jika terdapat penyimpangan inputan akan dilakukan perbaikan otomatis oleh robot agar menghasilkan gerakan yang lebih stabil dan terarah.

## Pengambilan Data dan Analisis

Tahapan analisis data adalah langkah krusial dalam proses penelitian. Pada tahap ini, data yang telah dikumpulkan menggunakan berbagai metode teknik pengumpulan akan diproses dan disajikan untuk membantu menjawab pertanyaan penelitian. Analisis data melibatkan pemisahan, perhitungan, dan pengkajian data agar dapat diinterpretasikan menjadi informasi yang relevan. Dalam pengambilan data dan analisis menggunakan beberapa perhitungan sebagai berikut:

### Nilai Error Sudut

$$(\text{Target} - \text{Input}) \leq 180^\circ$$



$$Error = Target - Input \quad (1)$$

Digunakan ketika selisih target dan *input* tidak melewati batas siklik (tidak melintasi  $\pm 180^\circ$ ). Dimana dengan range max sudut adalah  $180^\circ$  dan  $-180^\circ$ .

**(Target – Input) < -180°**

$$Error = (Target - Input) + 360^\circ \quad (2)$$

Digunakan ketika *input* berada di kuadran positif (+), tetapi target di kuadran negatif (-), dan selisihnya melewati batas bawah  $-180^\circ$ .

**(Target – Input) > 180°**

$$Error = (Target - Input) - 360^\circ \quad (3)$$

Digunakan ketika *input* berada di kuadran negatif (-), tetapi target di kuadran positif (+), dan selisihnya melewati batas atas  $+180^\circ$ .

**Nilai Persentase Error**

$$error \% = \frac{error\ yaw}{Rentang\ Maksimal\ yaw} \times 100\% \quad (4)$$

Nilai hasil *error* akan dibuat dalam nilai persentase dengan pembagian rentang maksimal sudut adalah  $180^\circ$ , lalu dapat dikalikan nilai 100% untuk mendapatkan persentase *error yaw*.

**Nilai Rata-Rata Error**

$$Rata - rata\ error = \frac{Penjumlahan\ semua\ data\ uji}{Jumlah\ data\ uji} \quad (5)$$

Didapatkan dari hasil penjumlahan semua data uji yang ada, lalu dibagi dengan berapa banyak jumlah data yang dihitung untuk mendapatkan nilai rata-rata data.

**Nilai Akurasi**

$$Akurasi = 100\% - (rata - rata\ error)\% \quad (6)$$

Didapatkan dari hasil rata-rata (dalam persentase) sehingga akan dikurangi 100% untuk mendapatkan nilai akurasi pembacaan.

## HASIL

### Pengujian Kalibrasi MPU6050

Dilihat dari Tabel 1 menunjukan nilai kalibrasi *error offset* akselerometer dan *gyroscope* untuk lingkungan yang berbeda, nilai ini menentukan akurasi pembacaan *offset* MPU sehingga nilai *yaw*, *pitch* dan *roll* memiliki nilai yang stabil. Nilai ini diperoleh dengan program kalibrasi MPU dari lingkungan yang berbeda rentang nilai yang mana memiliki



ketinggian atau kemiringan yang berbeda-beda. Kalibrasi akselerometer digunakan untuk mengkompensasi bias pada pengukuran akselerometer, nilai *Xaccel* mengoreksi *offset* pada sumbu X, *Yaccel* mengoreksi *offset* pada sumbu Y, dan *Zaccel* *offset* pada sumbu Z. Kalibrasi *gyroscope* digunakan untuk menghilangkan *drift* (pergeseran nilai saat sensor diam) pada *gyroscope* *offset*, nilai *Xgyro* mengoreksi *offset* pada sumbu X, *Ygyro* mengoreksi pada sumbu Y, dan *Zgyro* mengoreksi pada sumbu Z.

Tabel 1. Kalibrasi Offset MPU

Lingkungan	setXAccel	setYAccel	setZAccel	setXGyro	setYGyro	setZGyro
Lantai	-508	-11	1695	32	39	-29
Arena	-498	7	1694	33	40	-28
Karpet	-500	2	1694	33	40	-27

### Pengujian Kompensasi Motor

Berdasarkan karakteristik roda mekanum ini sangat bergantung dengan kondisi lapangan dan spesifikasi motor yang digunakan, karena robot menggunakan motor *gearbox* kuning dengan torsi 0,15 – 0,6 Nm maka perlu dilakukan penyesuaian agar robot dapat lebih maksimal saat bergerak terkhusus geser karena mekanum sangat sensitif dengan gerak geser dikarenakan proporsi roda, maka dilakukan pengujian mengubah nilai kalibrasi kompensasi per roda.

Table 2. Kalibrasi Kompensasi Geser Kiri

No.	FL	FR	RL	RR	Kecepatan (PWM)	Durasi (s)	Jarak Tempuh (m)	Deviasi Yaw (°)	Keterangan
1.	1,0	1,0	1,0	1,0	100	8	2.1	2	Terglincir Mundur
2.	0,9	1,1	0,9	1,2	100	8	2.5	-1	Agak Stabil
3.	1,2	1,1	1,1	1,2	100	8	2.4	-1	Terglincir Mundur Parah
4.	1,0	1,5	1,0	1,3	100	8	2.5	1	Agak Stabil
5.	1,2	1,1	1,2	1,3	100	8	2.3	3	Terglincir Mundur Parah

Pengujian kalibrasi kompensasi geser kiri pada Tabel 2, pengambilan data dilakukan sebanyak 5 kali percobaan yang menampilkan nilai posisi roda untuk melihat kestabilan gerakan bergeser ke kiri dalam lintasan lurus. Untuk percobaan pertama berada pada pengujian 1 sampai dengan 5 mendapatkan hasil saat dilakukan pada kecepatan motor 100 PWM dengan maksimal kecepatan motor 255 PWM yang dijalankan dengan melakukan pewaktuan gerak selama 8 detik. Sehingga didapatkan koreksi sudut *yaw* paling besar adalah 3 derajat dengan jarak tempuh relatif berbeda tergantung pada kestabilan dan terglincirnya roda saat bergerak. Berdasarkan keterangan hasil roda mengalami terglincir mundur untuk gerak geser ke kiri pada percobaan 1, 3, dan 5, tetapi

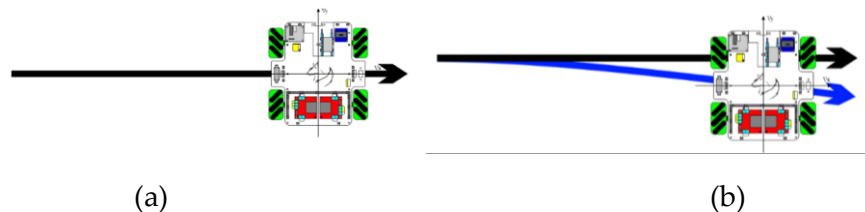


gerakan pada percobaan 3 dan 4 mulai menunjukkan kestabilan dari lintasan lurus dengan roda terglincir mundur hampir kecil.

Table 3. Kalibrasi Kompensasi Geser Kanan

No.	FL	FR	RL	RR	Kecepatan (PWM)	Durasi (s)	Jarak Tempuh (m)	Deviasi Yaw (°)	Keterangan
1.	1,0	1,0	1,0	1,0	100	8	1,8	-2	Terglincir Maju Parah
2.	1,1	1,0	0,9	1,2	100	8	1,8	1	Terglincir Maju
3.	1,2	1,1	1,1	1,3	100	8	2,1	-1	Terglincir Maju Parah
4.	1,0	1,5	1,0	1,3	100	8	2,4	1	Agak Stabil
5.	1,2	1,1	1,2	1,3	100	8	1,3	7	Terglincir Maju Parah

Sedangkan pada Tabel 3 dilakukan pengujian untuk gerak geser kekanan dengan koreksi yaw yang lumayan parah pada nilai 7 dengan jarak tempuh robot 1,3 meter dari yang jauh ada pada nilai 2,4 meter. Pengujian dilakukan dengan kondisi roda terglincir maju saat bergerak geser kekanan, Pada percobaan 1, 3, dan 5 mengalami terglincir yang sangat parah karena bergeser jauh dari lintasan lurus, sedangkan pada percobaan 2 juga mengalami terglincir tetapi roda belakang masih menyentuh garis lintasan, dan pada percobaan 4 didapatkan gerakan yang hanya mengalami terpleset kecil dengan jarak sejauh 2,4 meter.



Gambar 7. Visual Terglincir Saat Geser (a) Normal (b) Terglincir

Gambar 7 ditampilkan visual robot yang mengalami terglincir saat bergerak ke samping dikarenakan kekuatan motor, berat robot yang tidak seimbang dan karakteristik roda mekanum yang memiliki kesulitan di gerakan ini. Pada saat bergeser robot dapat lurus sesuai lintasan apabila mendapatkan nilai yang tepat sesuai dengan alas lintasannya, dan jika belum mendapatkan nilai yang tepat maka robot akan bergerak kesamping sambil agak maju atau mundur yang disebabkan perputaran roda motor yang kekuatannya berbeda-beda sehingga saat bergerak roda tidak memiliki kekuatan yang sama untuk menghasilkan gerakan yang stabil.

### Pengujian Dengan Koreksi Orientasi Robot

Dalam pengujian ini, robot diberikan faktor koreksi sudut robot secara otomatis dalam berbagai pola gerakan. Pengujian pertama dilakukan pada target sudut *yaw* 0 seperti pada



tabel 4, hasil dari pengujian menampilkan nilai *error yaw* dengan nilai tertinggi pada penyimpangan 3 derajat dan persentase 1,66%, lalu yang terendah pada nilai 0 dan persentase 0%. Semakin besar nilai antara target dengan nilai *input* maka semakin besar penyimpangan yang perlu dikoreksi, sedangkan semakin kecil selisih *error* dengan target semakin sedikit koreksinya sehingga robot dapat berjalan dengan meminimalisir koreksi. Nilai *error yaw* didapatkan nilai rata-rata dari semua error adalah 1,5 derajat dengan persentase pada angka 0,97%, maka dari persentase rata-rata *error* didapatkan nilai persentase akurasi 99,03%. Nilai persentase diukur dari nilai range pembatasan *yaw* +180 untuk arah CCW (*Counter-Clockwise*) dan -180 derajat untuk arah CW (*Clockwise*) dan, hal ini dilakukan untuk menentukan arah koreksi yang diperlukan robot untuk mengembalikan keseimbangan kembali ke nilai target.

Tabel 4. Pengujian Dengan Sudut 0 derajat

No.	Gerakan	Sudut Target (°)	Sudut Pembacaan Sensor (°)	Selisih (°)	Error yaw (°)	Error (%)
1	Maju	0	-1	1	1	0,56%
2	Mundur	0	-2	2	2	1,11%
3	Geser Kiri	0	1	1	-1	0,56%
4	Geser Kanan	0	1	1	-1	0,56%
5	Serong Kiri Maju	0	0	0	0	0%
6	Serong Kiri Mundur	0	-2	2	2	1,11%
7	Serong Kanan Maju	0	3	3	-3	1,66%
8	Serong Kanan Mundur	0	-2	2	2	1,11%
Rata-rata error					1.5	0,97%
Akurasi						99,03%

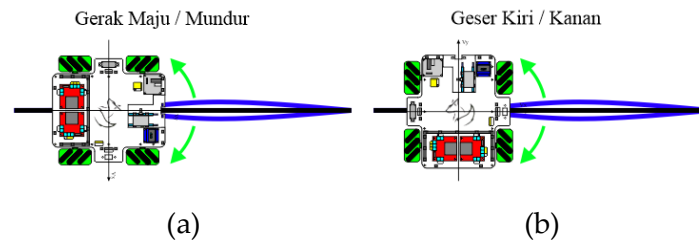
Sedangkan tabel 5 pada pengujian keempat mendapatkan selisih sudut paling besar pada nilai 3 derajat untuk rotasi kanan dan kiri dengan nilai persentase 1,66%, dan nilai terkecil pada sudut error 1 derajat dengan persentase *error* 0,56%. Hasil semua pengujian keempat ini didapatkan nilai rata-rata *error* pada 2 derajat dengan persentase *error* 1,11%, maka didapatkan tingkat akurasi sebesar 98,89% untuk koreksi robot saat berputar.

Tabel 5. Pengujian Rotasi Putar

No.	Gerakan	Sudut Target (°)	Sudut Pembacaan Sensor (°)	Selisih (°)	Error yaw (°)	Error (%)
1	Rotasi Kiri	-90	-88	2	-2	1,11%
2	Rotasi Kiri	-180	177	3	-3	1,66%
3	Rotasi Kiri	0	2	2	-2	1,11%
4	Rotasi Kanan	180	-177	1	1	0,56%
5	Rotasi Kanan	90	87	3	3	1,66%
6	Rotasi Kanan	0	-1	1	1	0,56%
Rata-rata error					2	1,11%
Akurasi						98,89%

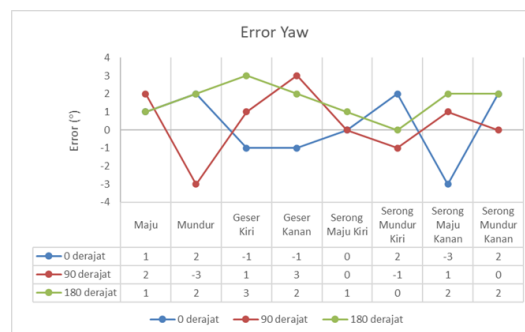


Pengujian setiap sudut dapat diamati pada Gambar 8 untuk potret saat pengujian dilapangan, pada Gambar 8 (a) adalah potret penyimpangan *yaw* dari pengujian sudut dalam gerak maju atau mundur dengan lintasan lurus saat terjadi penyimpangan maka orientasi akan memperbaiki robot agar kembali ke sudut awal. Untuk Gambar 8 (b) adalah potret penyimpangan *yaw* dari pengujian sudut dalam gerakan geser kanan atau kiri dengan kontrol orientasi memperbaiki kemiringan robot.



Gambar 8. Visual Lintasan Koreksi (a) Maju/Mundur (b) Geser Kiri/Kanan

Berdasarkan grafik *error yaw* yang ditampilkan pada Gambar 9 menunjukkan garis nilai *error* untuk setiap pola gerakan dengan 3 garis sebagai acuan untuk perbandingan pengujian setiap sudutnya. Garis biru untuk sudut 0 derajat, garis merah untuk sudut 90 derajat dan garis hijau untuk sudut 180 derajat. Grafik menunjukkan garis menurun mengindikasikan sumbu negatif yang ada dititik terendah ada pada nilai -3. Sedangkan sumbu positif tertinggi pada +3 derajat. Titik-titik kenaikan dan penurunan sudut *error yaw* masih dalam jarak antara 0-3 derajat, hal ini mengindikasikan bahwa koreksi robot saat melakukan gerakan sangat presisi dan memiliki akurasi kestabilan yang lumayan baik karena implementasi *gyro* sebagai pembacaan kemiringan robot.

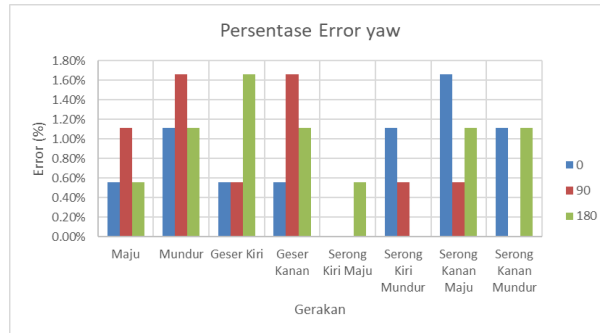


Gambar 9. Grafik Sudut *Error*

Untuk grafik persentase *error yaw* pada Gambar 10 menampilkan persentase dari tinggi atau rendahnya *error* dalam setiap gerakan yang dilakukan robot saat bergerak. Nilai persentase *error* tertinggi pada gerakan mundur, geser kanan, geser kiri dan serong kanan maju dengan persentase diatas 1,60%, sedangkan untuk grafik terendah pada gerakan serong kiri maju dan serong kanan mundur. Rata-rata presentase *error* robot sangat rendah



dari nilai maksimal 100% dan hanya didapatkan nilai tertinggi 1,66%, grafik ini menjadikan hasil bahwa koreksi orientasi *gyro* untuk robot sudah terimplementasikan dengan baik sebagai keseimbangan saat bergerak untuk mengatasi kemiringan dan ketidakseimbangan saat melakukan gerakan yang kompleks.



Gambar 10. Grafik Persentase Error Yaw

### Pengujian Tanpa Koreksi Orientasi Robot

Dalam pengujian ini, robot tanpa faktor koreksi sudut robot secara otomatis dalam berbagai pola gerakan, sehingga saat terjadi penyimpangan robot tidak dapat mengkoreksi lintasan lurus. Pengujian pertama dilakukan pada target sudut *yaw* 0 seperti pada tabel 6, hasil dari pengujian menampilkan nilai *error yaw* dengan nilai tertinggi pada penyimpangan 83 derajat dan persentase 48,11%, lalu yang terendah pada nilai 11 dan persentase 6,11%. Semakin besar nilai antara target dengan nilai *input* maka semakin besar penyimpangan yang perlu dikoreksi, sedangkan semakin kecil selisih *error* dengan target semakin sedikit koreksinya sehingga robot dapat berjalan dengan meminimalisir koreksi. Maka, nilai *error yaw* didapatkan nilai rata-rata dari semua *error* adalah 34,75 derajat dengan persentase pada angka 19,30%, dan persentase rata-rata *error* didapatkan nilai persentase akurasi 80,70%.

Tabel 6. Pengujian Dengan Sudut 0 derajat

No.	Gerakan	Sudut Target (°)	Sudut Pembacaan Sensor (°)	Selisih (°)	Error yaw (°)	Error (%)
1	Maju	0	11	11	-11	6,11%
2	Mundur	0	20	20	-20	11,11%
3	Geser Kiri	0	-40	40	40	22,22%
4	Geser Kanan	0	28	28	-28	15,55%
5	Serong Kiri Maju	0	-83	83	83	46,11%
6	Serong Kiri Mundur	0	-43	43	43	23,88%
7	Serong Kanan Maju	0	12	12	-12	6,66%
8	Serong Kanan Mundur	0	41	41	-41	22,77%
Rata-rata error					34,75	19,30%
Akurasi						80,70%

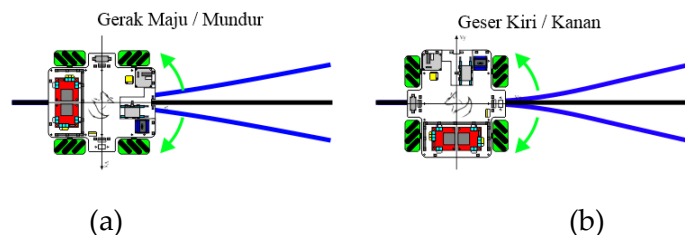


Sedangkan pada table 7 pengujian keempat dilakukan untuk pembacaan gerakan rotasi robot dalam sudut yang dilakukan dengan pewaktuan untuk menentukan durasi berputar tepat mendekati nilai sudut target dan kecepatan diatur pada 100 PWM. Pengujian ini mendapatkan selisih sudut paling besar pada nilai 107 derajat untuk rotasi kiri dengan nilai persentase 59,44%, dan nilai terkecil pada sudut *error* 4 derajat dengan persentase *error* 2,22%. Pada rotasi kanan hanya selisih 5 derajat, hal ini dapat dipengaruhi kecepatan roda yang berbeda dan tidak adanya inputan untuk robot berputar sampai sudut yang dituju. Hasil semua pengujian keempat ini didapatkan nilai rata-rata *error* pada 26,37 derajat dengan persentase *error* 19,53%, didapatkan tingkat akurasi 80,46%.

Tabel 7. Pengujian Rotasi Putar

No.	Gerakan	Waktu Rotasi (s)	Sudut Target (°)	Sudut Pembacaan Sensor (°)	Selisih (°)	Error yaw (°)	Error (%)
1	Rotasi Kiri	1150	-180°	-73	107	107	59,44%
2	Rotasi Kiri	550	-90°	-174	84	-84	46,66%
3	Rotasi Kiri	1150	0°	-4	4	-4	2,22%
4	Rotasi Kanan	1150	180°	175	5	-5	2,77%
5	Rotasi Kanan	550	90°	83	7	-7	3,88%
6	Rotasi Kanan	1150	0°	4	4	4	2,22%
Rata-rata <i>error</i>						26,37	19,53%
Akurasi							80,46%

Pengujian setiap sudut dapat diamati pada Gambar 11 untuk potret saat pengujian dilapangan, pada Gambar 11 (a) adalah potret penyimpangan *yaw* dari pengujian sudut dalam gerak maju atau mundur dengan lintasan lurus saat terjadi penyimpangan maka orientasi tidak dapat memperbaiki robot agar kembali ke sudut awal karena tidak menerapkan pembacaan orientasi *gyro*. Untuk Gambar 11 (b) adalah potret penyimpangan *yaw* dari pengujian sudut dalam gerakan geser kanan atau kiri tanpa kontrol orientasi, sehingga robot tidak dapat memperbaiki kemiringan dan melenceng dari lintasan.

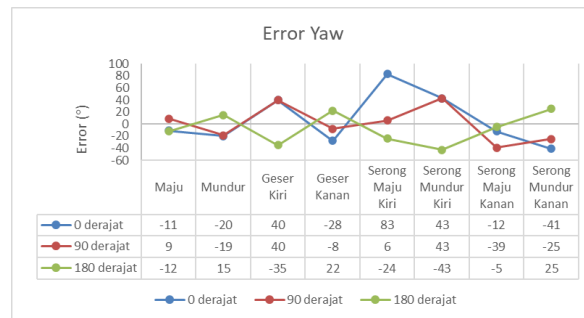


Gambar 11. Visual Tanpa Koreksi (a) Maju/Mundur (b) Geser Kiri/Kanan

Berdasarkan grafik *error yaw* yang ditampilkan pada Gambar 12 menunjukkan garis nilai *error* untuk setiap pola gerakan dengan 3 garis sebagai acuan untuk perbandingan

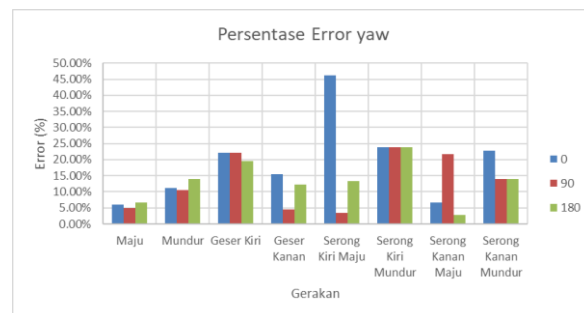


pengujian setiap sudutnya. Garis biru untuk sudut 0 derajat, garis merah untuk sudut 90 derajat dan garis hijau untuk sudut 180 derajat. Grafik menunjukkan nilai positif dan negatif pada setiap titik yang ada pada garis linear, garis menurun mengindikasikan sumbu negatif yang ada dititik terendah ada pada nilai -43 pada linear 180 derajat. Sedangkan sumbu positif tertinggi pada +83 derajat untuk linear 0 derajat. Titik-titik kenaikan dan penurunan sudut *error yaw* masih dalam jarak antara -43 - 83 derajat, hal ini mengindikasikan bahwa koreksi robot saat melakukan gerakan sangat tidak seimbang dan memiliki akurasi kestabilan yang kurang karena tidak menerapkan implementasi *gyro* sebagai pembacaan kemiringan robot.



Gambar 12. Grafik Sudut *Error*

Untuk grafik persentase *error yaw* pada Gambar 13 menampilkan persentase dari tinggi atau rendahnya *error* dalam setiap gerakan yang dilakukan robot saat bergerak. Nilai persentase *error* tertinggi dengan persentase diatas 45%, sedangkan untuk grafik terendah pada persentase dibawah 5%. Rata-rata persentase *error* robot mengalami kenaikan tajam diangka melebihi 10% dikarenakan tanpa ada koreksi kemiringan robot, grafik ini menjadikan hasil bahwa tanpa adanya koreksi orientasi *gyro* untuk robot sebagai keseimbangan saat bergerak tidak dapat mengatasi kemiringan dan ketidak seimbangan saat melakukan gerakan yang kompleks.



Gambar 13. Grafik Persentase *Error*

## Pengujian Tuning Nilai PID

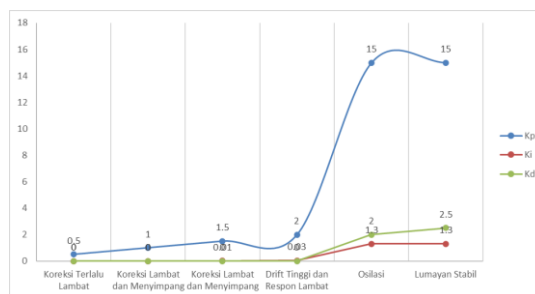


Robot bergerak dalam kondisi tanpa arah dan keseimbangan yang stabil dikarenakan kekuatan motor yang berbeda-beda menyebabkan robot menyimpang dari lintasan lurus, maka dari itu tuning PID diperlukan untuk mengurangi simpangan robot. Pada tabel 8, tuning dilakukan secara manual melalui percobaan mengatur parameter Kp (Proportional), Ki (Integral), dan Kd (Derivative) dengan pengamatan secara langsung respon gerakan robot. Tuning nilai dilakukan dari menaikkan Kp menjadi 0,5 dan respon koreksi robot lambat dan menyimpang sehingga menaikkan nilai nilai Kp untuk respon cepat koreksi, Ki untuk menghilangkan *steady-state error* karena robot tidak mencapai target *yaw* dan Kd untuk getaran atau *overshoot* sesuai dengan respon robot yang terjadi. Didapatkan nilai mendekati stabil ada pada nilai Kp 15,0, Ki 1,3 dan Kd 2,5 dengan kondisi robot bergerak pada jalur lurus secara stabil dan memiliki sedikit osilasi maupun *overshoot*.

Tabel 8. Tuning Nilai PID

No.	Kp	Ki	Kd	Respon Robot	Catatan
1	0.5	0	0	Koreksi Terlalu Lambat	Naikan Kp
2	1	0	0	Koreksi Lambat dan Menyimpang	Naikan Kp dan Ki
3	1.5	0.01	0	Koreksi Lambat dan Menyimpang	Naikan Kp dan Ki
4	2	0.03	0	Drift Tinggi dan Respon Lambat	Naikan Kp dan Ki
5	15	1.3	2	Osilasi	Naikan Kd Sedikit
6	15	1.3	2.5	Lumayan Stabil	Naikan Kd atau Kp

Grafik pada Gambar 14 menunjukkan dimana perubahan respon nilai tergantung pada nilai PID yang di atur dengan garis biru sebagai Kp, garis merah sebagai Ki dan garis hijau sebagai Kd. Grafik menampilkan bahwa nilai Kp mempengaruhi kestabilan robot dalam bergerak karena membutuhkan respon yang cepat sehingga nilai ini lebih besar dari penerapan nilai Ki dan Kd untuk membantu robot agar lebih stabil.



Gambar 14. Grafik Tuning Nilai PID

### Pengujian Performa Robot di Arena

Percobaan pengujian dilakukan kedalam tiga *plan* pergerakan, dimana setiap gerakan diatur dengan menggunakan sensor MPU6050 dan sensor ultrasonik untuk menentukan posisi berdasarkan jarak. Pada Tabel 9 diperlihatkan percobaan yang sudah dilakukan robot saat diterapkan kedalam arena dengan rata-rata barang berhasil disimpan adalah 3



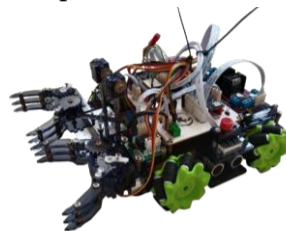
barang dengan waktu 80 detik permainan, hal ini disebabkan karena robot hanya menggunakan sensor ultrasonik sebagai koordinat posisi melalui jarak yang terkadang pada saat robot bergerak secara cepat untuk melakukan aksi pengambilan ataupun menyimpan pada sensor mengalami gangguan keakuratan dalam membaca jaraknya, yang menyebabkan robot tidak memosisikan pada koordinat yang tepat untuk menyimpan barangnya. Keterangan pada tabel menunjukkan bahwa rata-rata permasalahan robot adalah gagal dalam menyimpan barang yang dibawanya, dan barang yang tidak masuk kedalam kotak penyimpanan ataupun terdapat barang yang tergeser karena *error* pembacaan oleh sensor ultrasonik sehingga robot menggeser barang yang sudah ditaruh kedalam kotak dan tergeser.

Tabel 9. Pengujian Performa Robot Dengan *gyro*

Uji Coba	Barang Terambil	Barang Tersimpan	Durasi (s)	Persentase Keberhasilan		Keterangan
				Diambil	Disimpan	
1	6	4	96	100%	66,66%	2 Barang Gagal Disimpan
2	6	5	97	100%	83,33%	1 Barang Gagal Disimpan
3	4	2	88	66,66%	33,33%	2 Gagal Terambil dan 2 Gagal Disimpan
4	6	4	86	100%	66,66%	2 Barang Gagal Disimpan
5	6	2	76	100%	33,33%	4 Barang Gagal Disimpan
6	4	1	96	66,66%	16,66%	2 Barang Gagal Terambil, 3 Gagal Disimpan
7	6	3	78	100%	50%	3 Gagal Disimpan
8	6	4	95	100%	66,66%	2 Gagal Disimpan

## PEMBAHASAN

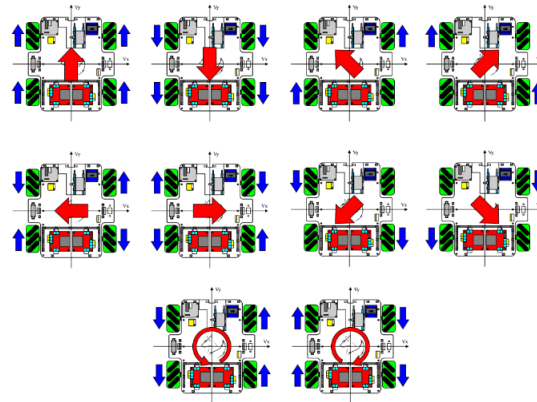
Robot ini adalah prototipe dengan dimensi 30 x 18 cm dengan bahan perakitan menggunakan kerangka akrilik 3mm yang ringan. Robot ini bekerja untuk menyelesaikan misi pada arena atau bergerak dengan banyak pola untuk menguji keseimbangan roda mekanum saat bergerak apakah selalu stabil dan tidak *overshoot*. Tampilan robot dapat dilihat pada Gambar 15 yang menampilkan dari bentuk fisik robot.



Gambar 15. Bentuk Fisik Robot

Pada Gambar 16 merupakan pola gerakan dari roda mekanum, yang mana ditentukan untuk arah maju, mundur, geser kiri, geser kanan, serong kiri maju, serong kiri mundur, serong kanan maju, serong kanan mundur, rotasi ke kiri dan rotasi ke kanan. Untuk pola gerak sendiri ditentukan dari arah putaran roda dan posisinya.





Gambar 16. Pola Gerak Mekanum

Gerakan ditentukan dari arah putaran roda dan posisinya, pada Gambar 17 menampilkan rumus perhitungan nilai *wheelSpeeds* sebagai deklarasi roda FL (*Front Left*), FR (*Front Right*), RL (*Rear Left*), dan RR (*Rear Right*). Penentuan arah berdasarkan nilai  $V_y$  sebagai arah maju atau mundur,  $V_x$  sebagai geser kanan atau kiri, dan  $\omega$  sebagai orientasi rotasi koreksi.

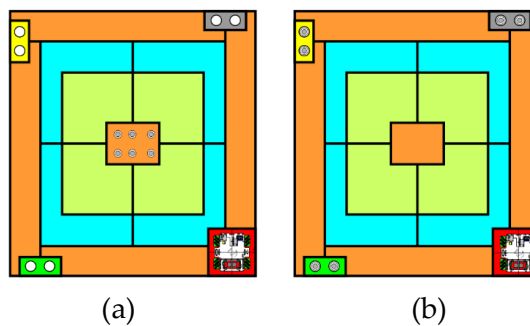
```

wheelSpeeds[0] = (vy - vx - omega * omega_factor) * 100.0f; // FL
wheelSpeeds[1] = (vy + vx + omega * omega_factor) * 100.0f; // FR
wheelSpeeds[2] = (vy + vx - omega * omega_factor) * 100.0f; // RL
wheelSpeeds[3] = (vy - vx + omega * omega_factor) * 100.0f; // RR

```

Gambar 17. Rumus Arah Gerak Roda

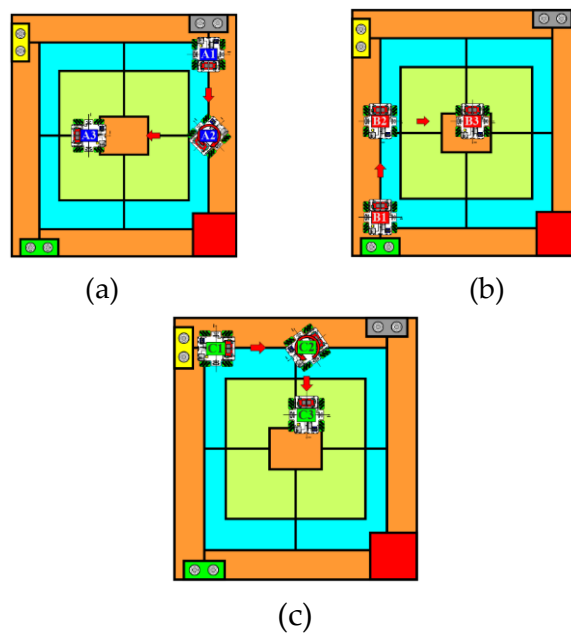
Pada Gambar 18 (b) adalah area pengambilan barang, dimana barang akan diambil oleh robot yang kemudian akan dipindahkan kedalam arena penyimpanan pada Gambar 18 (a). Robot bergerak dengan 3 *plan* utama untuk pengambilan barang, setiap *plan* akan mengambil 2 barang untuk dipindahkan kedalam area penyimpanan dengan total barang berjumlah 6. Setiap gerakan dan posisi koordinat robot dibantu sensor MPU6050 untuk kestabilan dan sensor ultrasonik sebagai pengatur posisi berdasarkan jarak robot untuk setiap titik pergantian gerakan, mulai dari *plan* 1 sampai *plan* 3.



Gambar 18. Arena (a) Penyimpanan (b) Pengambilan



Posisi robot ditentukan oleh sensor ultrasonik yang dipasang pada 4 sisi robot yaitu, depan, belakang, kiri dan kanan. Posisi diukur dalam hitungan jarak cm untuk setiap koordinat yang ditentukan melalui setiap *plan* yang sudah ditentukan gerakannya, setiap gerakan dibagi menjadi beberapa gerakan. Seperti pada Gambar 19 (a) untuk *plan* 1 yang menggunakan 3 posisi pembacaan sensor dari A1 mundur, A2 rotasi kiri dan mundur sampai A3, Gambar 19 (b) untuk *plan* 2 yang juga menggunakan 3 posisi dengan B1 mundur, B2 geser kiri sampai B3, lalu Gambar 19 (c) untuk *plan* 3 yang juga menerapkan 3 posisi gerakan dengan C1 mundur, C2 rotasi kanan dan maju sampai C3. Keseluruhan posisi A1, B1 dan C1 sebagai koordinat pengambilan barang yang akan bergerak sampai posisi A3, B3 dan C3 untuk koordinat penyimpanan barang yang berhasil di ambil.



Gambar 19. Koordinat (a) Plan 1 (b) Plan 2 (c) Plan 3

## KESIMPULAN

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, dapat disimpulkan bahwa penerapan *gyroscope* dengan MPU6050 sebagai keseimbangan gerakan robot beroda mekanum pada robot otonom ini menunjukkan kinerja yang lumayan optimal dan bekerja dengan baik untuk menyeimbangkan gerakan dalam berbagai sudut. Robot banyak mengalami kemiringan dan gangguan dalam melakukan gerakan apalagi didalam arena atau lantai yang memerlukan gerakan kompleks ataupun terdapat halangan roda untuk berjalan selalu konstan lurus dalam lintasan, maka dari itu pengujian mendapatkan jarak selisih pada saat melakukan gerakan kompleks antara sudut 0 - 4 derajat sebagai toleransi. Motor *gearbox* berbeda-beda kekuatannya sehingga



kalibrasi nilai kompensasi untuk menaikkan atau menurunkan kecepatan PWM motor khusus gerakan geser. Sedangkan untuk penerapan pada arena uji coba perlu dilakukan perubahan dan perbaikan untuk sistem penentuan koordinat posisi robot dengan sensor ultrasonic HC-SR04, karena kurang akurat saat membaca secara serial dalam waktu cepat atau saat terjadi perubahan jarak secara cepat.

### **UCAPAN TERIMA KASIH**

Alhamdulillah. Penulis mengucapkan puji syukur kepada Allah SWT yang telah melimpahkan rahmat serta hidayah-nya sehingga penulis dapat menyelesaikan penulisan Naskah Publikasi ini dengan baik. Shalawat serta salam tidak lupa saya curahkan kepada Nabi besar Muhammad SAW. Dengan terselesaikannya Naskah Publikasi ini, penulis ingin menyampaikan terimakasih juga kepada :

1. Kedua orang tua dan saudara yang telah memberikan dukungan serta doa, sehingga penulis diberikan kelancaran, rahmat dan semangat untuk menyelesaikan penelitian ini dengan baik.
2. Dosen pembimbing penelitian ini, kepada Bapak Ir. Pratomo Budi Santosa, M.T. yang telah memberikan arahan serta bimbingan kepada penulis dalam menyelesaikan penelitian tugas akhir ini.
3. Bapak/Ibu dosen Teknik Elektro yang telah memberikan ilmu dan bimbingan selama perkuliahan, sehingga membantu penulis dalam menyelesaikan penelitian ini tanpa kendala berarti.
4. Komunitas *Robot Research* UMS dan Tim KRI UMS, yang telah menjadi tempat bagi penulis untuk mengembangkan ilmu praktik yang diperoleh selama kuliah.
5. Terima kasih kepada teman-teman Kontrakan Ahmad Mullet dan teman - teman Kos Wisma Pandawa yang telah membantu, menemani dan memberikan tempat bagi penulis dalam mengerjakan penelitian ini.
6. Terima kasih kepada semua pihak yang tidak dapat disebutkan satu per satu atas dukungan yang telah diberikan kepada penulis selama masa akhir kuliah. Penulis berterima kasih atas pelajaran yang telah membuatnya lebih sadar dan terus belajar untuk terus menjadi lebih baik dalam menjalani kehidupan.



## DAFTAR PUSTAKA

- [1] T. Chen, G. Yang, Q. Cai, Z. Wen, and W. Zhang, "A Novel Calibration Method for Gyro-Accelerometer Asynchronous Time in Foot-Mounted Pedestrian," 2022.
- [2] E. Eurobot, "Organization GitHub," 2018.
- [3] F. Fahmizal, A. Mayub, and M. Arrofiq, "Sistem Gerak Robot Mainland Surveillance menggunakan Mecanum Wheel sebagai Militer Robot," *Majalah Ilmiah Teknologi Elektro*, vol. 17, no. 2, pp. 205, 2018. doi: 10.24843/mite.2018.v17i02.p07
- [4] Y. Kaladewa and K. A. Santoso, "Implementasi Sensor Kemiringan Sudut Untuk Alat Bantu (Grab) Gantry Luffing Crane (Glc)," *Jurnal Kajian Teknik Elektro*, vol. 6, no. 2, pp. 62–69, 2022. doi: 10.52447/jkte.v6i2.5726
- [5] Y. Li et al., "Kinematic modeling of a combined system of multiple mecanum-wheeled robots with velocity kompensasi," *Sensors (Switzerland)*, vol. 20, no. 1, 2020. doi: 10.3390/s20010075
- [6] J. Mohd Sultan, N. H. Zani, M. Azuani, S. Z. Ibrahim, and A. Md Yusop, "Analysis of Inertial Measurement Accuracy using Complementary Filter for MPU6050 Sensor," *Jurnal Kejuruteraan*, vol. 34, no. 5, pp. 959–964, 2022. doi: 10.17576/jkukm-2022-34(5)-24
- [7] W. S. Purnomo and A. Soetedjo, "Simulasi Gerak Robot Soccer Beroda Menggunakan Kinematika Omnidireksional Roda Empat," *Magnetika: Jurnal Mahasiswa Teknik Elektro*, vol. 7, no. 2, pp. 229–236, 2021. [Online]. Available: <http://eprints.itn.ac.id/5437/>
- [8] R. G. Putra and B. Artono, "Sistem Kemudi Semi Otomatis Dengan Gyroscope Pada Mobile Robot Berstruktur Origami," *Teknologi Informasi Dan Terapan*, vol. 3, no. 1, pp. 287–293, 2016.
- [9] A. Rahman, I. Setiawan, and D. Darjat, "Perancangan Sistem Stabilisasi Roll Dan Yaw Pada Prototype Hovercraft Menggunakan Sensor Imu 6 Dof Dengan Metode Self Tuning Fuzzy PID," *Transient: Jurnal Ilmiah Teknik Elektro*, vol. 6, no. 4, pp. 524–533, 2017.
- [10] T. Wahyudi, D. Marindani, and A. Elbani, "Rancang Bangun Robot Beroda Pengangkut Barang Menggunakan Sensor Gyroscope," *Journal of Electrical Engineering, Energy and Information Technology*, vol. 10, no. 2, 2022. [Online]. Available: <https://jurnal.untan.ac.id/index.php/j3eituntan/article/view/56713>
- [11] Y. B. Widodo, T. I. Rahman, S. Sibuea, and M. I. Saputro, "Design and Development of Intelligent and Autonomous Control for Wheeled Car Robots," *Jurnal Teknologi Informatika Dan Komputer*, vol. 9, no. 2, pp. 756–771, 2023.



- [12] L. Zheng, C. Hong, H. Song, and R. Chen, "An autonomous mobile robot path planning strategy using an enhanced slime mold algorithm," *Frontiers in Neurobotics*, vol. 17, 2023. doi: 10.3389/fnbot.2023.1270860
- [13] Robotics Stack Exchange, "How to choose closest angle when crossing +/-180 degrees?," Stack Exchange, 2021. [Online]. Available: <https://robotics.stackexchange.com/questions/21911/how-to-choose-closest-angle-when-crossing-180-degrees>
- [14] Wikipedia, "Proportional-integral-derivative controller," Wikipedia, 2023. [Online]. Available: [https://en.wikipedia.org/wiki/Proportional%E2%80%93integral%E2%80%93derivative\\_controller](https://en.wikipedia.org/wiki/Proportional%E2%80%93integral%E2%80%93derivative_controller)
- [15] Ooi, K. R., Rosli, M. A. A., Latiff, A. R. A., Othman, W. A. F. W., Alhady, S. S. N., & Wahab, A. A. A. (2021). Design of hoeckens linkage based walking robot with MPU6050 IMU as navigation sensor. *Journal of Physics: Conference Series*, 1969(1), 0–11. <https://doi.org/10.1088/1742-6596/1969/1/012004>

