

Rancang Bangun Robot Penghindar Rintangan Berbasis *Raspberry Pi Pico* Dan Sensor *Time of Flight*

DOI: <http://dx.doi.org/10.35889/progresif.v22i2.3674>

Creative Commons License 4.0 (CC BY –NC)



Lubersius Zamasi^{1*}, Gogor Chrissmass Setyawan², Kristian Juri Damai Lase³

Teknik Informatika, Universitas Kristen Immanuel, Yogyakarta, Indonesia

*e-mail Corresponding Author. lubersius.zamasi@mail.ukrim.ac.id

Abstract

This research addresses the collision risks in remote-controlled robots caused by communication latency and human operator response limitations. The primary focus of this study is the development of a lightweight autonomous safety override mechanism using a resource-constrained microcontroller, the Raspberry Pi Pico W. The system integrates the VL53L0X Time of Flight sensor for high-precision obstacle detection and the KY-032 infrared sensor for cliff protection. The methodology involves designing a hybrid control architecture that combines web-based instructions with real-time safety logic. Experimental results demonstrate that the ToF sensor achieves an accuracy rate of 98.01%, surpassing the reliability of conventional ultrasonic sensors. The cliff protection system shows a 100% success rate in performing automatic avoidance maneuvers without relying on operator input. This research proves that integrating ToF sensors and override mechanisms on low-power platforms significantly enhances navigation safety for Internet of Things (IoT)-based robots.

Keywords: *Internet of Things; Raspberry Pi Pico W; KY-032 sensor; Time of Flight sensor; autonomous safety.*

Abstrak

Penelitian ini mengatasi permasalahan risiko tabrakan pada robot kendali jarak jauh yang disebabkan oleh latensi komunikasi dan keterbatasan respons operator manusia. Fokus utama penelitian adalah pengembangan mekanisme keselamatan otonom (*autonomous safety override*) yang ringan (*lightweight*) menggunakan mikrokontroler berbasis sumber daya terbatas, Raspberry Pi Pico W. Sistem ini mengintegrasikan sensor *Time of Flight* VL53L0X untuk deteksi rintangan presisi tinggi dan sensor inframerah KY-032 sebagai proteksi terhadap area curam (jurang). Metodologi yang digunakan mencakup perancangan arsitektur *hybrid control* yang menggabungkan instruksi *web-based* dengan logika proteksi *real-time*. Hasil pengujian menunjukkan sensor ToF mencapai tingkat akurasi 98,01%, melampaui keandalan sensor ultrasonik konvensional. Sistem proteksi jurang menunjukkan tingkat keberhasilan 100% dalam melakukan manuver penghindaran otomatis tanpa bergantung pada instruksi operator. Penelitian ini membuktikan bahwa integrasi sensor ToF dan mekanisme *override* pada platform hemat daya mampu memberikan jaminan keamanan navigasi yang signifikan bagi robot berbasis *Internet of Things* (IoT).

Kata kunci: *Internet of Things; Raspberry Pi Pico W; sensor KY-032; sensor Time of Flight*

1. Pendahuluan

Teknologi robotika otonom dan semi-otonom saat ini berkembang pesat dan diterapkan dalam berbagai bidang yang membutuhkan mobilitas tinggi. Perkembangan ini terlihat dari penggunaan kendaraan otonom di sektor pertanian [1], robot cerdas untuk penyelamatan korban gempa [2], robot pencari bersendi banyak (*centipede-inspired*) [3], hingga *drone* untuk pengiriman barang [4]. Kemajuan mobilitas ini sangat didukung oleh jaringan *Internet of Things* (IoT) dan 5G yang memungkinkan pertukaran data secara *real-time* [5], serta teknologi pelacakan posisi di dalam ruangan yang presisi [6]. Dengan inovasi penggerak seperti roda *omnidirectional* [7], robot masa kini dapat bergerak dengan sangat lincah. Oleh karena itu,

pengujian performa fungsional yang ketat menjadi sangat penting untuk memastikan robot dapat beroperasi dengan andal [8].

Tantangan utama saat robot bermanuver, terutama yang dikendalikan dari jarak jauh (nirkabel) menggunakan mikrokontroler kecil seperti Raspberry Pi Pico [9], adalah kemampuan menghindari rintangan secara instan. Ketergantungan pada kendali manusia sering kali menimbulkan risiko tabrakan karena adanya *delay* (latensi jaringan) dan lambatnya respons operator. Selain tabrakan di area depan, robot beroda juga sangat rentan jatuh akibat perbedaan ketinggian lantai (jurang) yang ekstrem. Masalah navigasi ini sebenarnya bisa diselesaikan menggunakan algoritma pemetaan kompleks seperti SLAM. Namun, algoritma tersebut membutuhkan daya komputasi dan memori yang sangat besar, sehingga tidak cocok dan tidak efisien jika diterapkan pada mikrokontroler berbiaya rendah [10].

Beberapa penelitian sebelumnya telah mencoba menyelesaikan masalah navigasi dan penghindaran rintangan ini. Xia et al. [11] menggunakan metode *Artificial Potential Field* (APF) untuk pergerakan lengan robot. Umam et al. [12] memanfaatkan kamera (*stereo vision*) untuk mendeteksi rintangan pada robot *omni-directional*. Terkait akurasi sensor, Rahman [13] membuktikan bahwa sensor laser Time-of-Flight (ToF) jauh lebih presisi dibandingkan sensor ultrasonik biasa. Hal ini didukung oleh Gimpelj dan Munich [14] serta Sarmiento et al. [15] yang berhasil menggabungkan sensor ToF dan kamera untuk keamanan navigasi. Untuk masalah jurang, Sholakhudin [16] membuktikan bahwa sensor inframerah (IR) jarak dekat sangat responsif untuk mendeteksi lantai yang berlubang. Meskipun banyak komponen yang sudah diuji, masih terdapat celah (*gap*) penelitian: belum ada sistem yang menggabungkan keamanan ganda (sensor ToF untuk rintangan dan IR untuk jurang) ke dalam satu sistem kendali *web* nirkabel yang hemat memori (RAM) pada mikrokontroler berbiaya rendah.

Untuk menyelesaikan masalah tersebut, penelitian ini mengusulkan konsep (*State of the Art*) robot mobil dengan mekanisme pengambilalihan kendali darurat (*safety override*) menggunakan pemodelan *Finite State Machine* (FSM). Penggunaan mikrokontroler Raspberry Pi Pico W untuk memproses sensor ToF VL53L0X dan IR KY-032 sangat efektif karena metode FSM tidak memerlukan komputasi berat, melainkan hanya menggunakan logika transisi kondisi (*state*) yang sederhana dan cepat. Kebaruan (*novelty*) dari penelitian ini terletak pada pemodelan kendali FSM yang dioptimasi untuk komputasi ringan (*lightweight computation*). Sistem ini secara otomatis akan memutus perintah maju dari *web* jika sensor mendeteksi batas kritis rintangan atau jurang. Hal ini menghasilkan sistem navigasi robot IoT yang jauh lebih aman, responsif, dan stabil tanpa membuat memori perangkat menjadi penuh.

2. Metodologi

2.1 Analisa Kebutuhan Sistem

Tahap analisis kebutuhan sistem dilakukan untuk menetapkan parameter teknis yang mendukung optimasi *lightweight computation* pada mikrokontroler Raspberry Pi Pico W. Fokus utama perancangan ini adalah mewujudkan otonomi tertanam berbiaya rendah (*low-cost embedded autonomy*) yang memiliki keandalan tinggi pada mekanisme interupsi keselamatan waktu-nyata (*real-time safety override*). Kebutuhan sistem dikategorikan menjadi spesifikasi fungsional dan parameter evaluasi kuantitatif.

2.1.1 Kebutuhan fungsional

Tahap analisis kebutuhan fungsional dilakukan untuk mengidentifikasi dan menetapkan fungsi utama yang harus dimiliki oleh sistem robot agar dapat beroperasi sesuai dengan tujuan penelitian. Proses ini mencakup penentuan kemampuan sistem dalam menerima perintah kendali, mendeteksi lingkungan, serta merespons kondisi secara otomatis. Hasil analisis ini menjadi dasar dalam perancangan sistem, baik dari sisi perangkat keras maupun perangkat lunak yang diimplementasikan pada mikrokontroler Raspberry Pi Pico W.

Tabel 1. Spesifikasi Kebutuhan Fungsional Sistem

No	Fitur Fungsional	Deskripsi Teknis
1	Kontrol Nirkabel	Menerima paket instruksi dari antarmuka <i>web</i> melalui protokol HTTP pada jaringan WiFi lokal.
2	Deteksi Rintangan	Akuisisi data jarak absolut objek secara kontinu menggunakan sensor ToF VL53L0X.
3	Deteksi Jurang	Pemantauan elevasi permukaan lintasan secara instan menggunakan sensor inframerah KY-032.
4	<i>Safety Override</i>	Mekanisme interupsi otomatis yang menghentikan motor saat <i>threshold</i> keamanan tercapai, tanpa menunggu instruksi pengguna.
5	Navigasi Adaptif	Pemindaian jalur alternatif menggunakan motor servo saat jalur utama terdeteksi rintangan.

2.1.2 Parameter Evaluasi Sistem

Untuk menjawab tuntutan evaluasi analitik (bukan sekadar deskriptif), penelitian ini menetapkan metrik terukur untuk menguji performa sistem. Parameter masukan (input) dan keluaran (output) ini dirancang secara spesifik agar berkaitan langsung dengan setiap fungsionalitas yang telah ditetapkan pada Tabel 1, sehingga dapat diuji secara empiris pada tahap eksperimen hasil. Rincian metrik evaluasi disajikan pada Tabel 2.

Tabel 2. Parameter Evaluasi Kinerja Fungsional Sistem

Relasi Fungsionalitas (Tabel 1)	Parameter yang Diuji	Metrik Evaluasi	Target Kinerja
Kontrol Nirkabel	Kinerja Jaringan WiFi & Web Server	<i>Packet Latency & Packet Loss Rate</i>	Latensi < 100 ms; <i>Packet Loss</i> 0%
Deteksi Rintangan	Akurasi Input Jarak (Sensor ToF)	Kesalahan Relatif (<i>Relative Error</i>)	Tingkat <i>Error</i> < 5% terhadap jarak aktual
Deteksi Jurang	Keandalan Input Status (Sensor IR)	Tingkat Keberhasilan Deteksi Kritis	Keberhasilan 100% pada jarak jurang ≤ 5 cm
<i>Safety Override</i>	Respons Keluaran Sistem (Interupsi Motor)	<i>System Response Time</i>	< 200 ms (dari deteksi bahaya hingga aktuator berhenti)
Navigasi Adaptif	Manuver Rotasi & Pemindaian Area (Servo)	Tingkat Keberhasilan Manuver (<i>Success Rate</i>)	Keberhasilan 100% berbelok menemukan rute kosong

2.2 Perancangan Sistem

Perancangan sistem pada penelitian ini dibagi menjadi empat arsitektur utama: perangkat keras, pemetaan fisik, komunikasi antarmuka, dan algoritma kendali. Desain ini difokuskan pada pencapaian komputasi ringan (*lightweight computation*) guna menjalankan mekanisme *real-time safety override* tanpa membebani memori mikrokontroler.

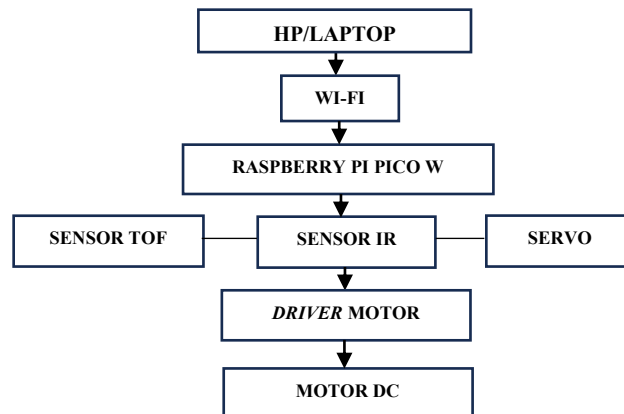
2.2.1 Arsitektur Sistem Perangkat Keras

Pusat kendali komputasi diotaki oleh mikrokontroler Raspberry Pi Pico W yang bertindak sebagai *server* lokal sekaligus pemroses data sensor. Sistem ini menerima dua masukan (input) persepsi lingkungan:

- 1) Sensor ToF VL53L0X yang berkomunikasi melalui protokol I2C untuk mendeteksi rintangan di depan secara persisi tinggi.
- 2) Sensor Inframerah KY-032 yang memanfaatkan pin GPIO digital untuk mendeteksi tepi jurang atau ketiadaan permukaan pijakan secara instan.

Data dari kedua sensor ini diproses secara siklis. Jika kondisi aman, mikrokontroler akan mengeksekusi instruksi dari antarmuka *web* dan mengirimkan sinyal kendali PWM (*Pulse*

Width Modulation) menuju *driver* motor L298N untuk menggerakkan motor DC, serta mengendalikan motor servo sebagai aktuatur pemindai arah (*directional scanner*).



Gambar 1. Arsitektur Sistem

2.2.2 Rancangan Fisik dan Pemetaan Pin

Integrasi fisik antar komponen membutuhkan konfigurasi pengkabelan (*wiring*) yang presisi untuk mencegah terjadinya *bottleneck* aliran data maupun malfungsi daya. Pemetaan pin antarmuka mikrokontroler Raspberry Pi Pico W dengan periferil sensor dan aktuatur dijabarkan pada Tabel 3.

Tabel 3. Konfigurasi Pin

Komponen Periferil	Pin Modul	Pin Raspberry Pi Pico W	Deskripsi Fungsional
Sensor ToF VL53L0X	SDA, SCL	GP0, GP1	Komunikasi I2C
Motor Servo	Signal	GP18	Kontrol Sudut (PWM)
Motor Driver L298N	ENA, IN1, IN2	GP6, GP7, GP8	Kontrol Motor Kiri
	ENB, IN3, IN4	GP9, GP10, GP11	Kontrol Motor Kanan
Sensor IR KY-032	Left DO, Right DO	GP14, GP15	Deteksi Digital Deteksi Jurang
Power Bus	VCC, GND	VCC, GND	Distribusi Daya

2.2.3 Arsitektur Antarmuka dan Komunikasi Jaringan

Antarmuka pengguna (*User Interface*) dirancang berbasis *web* yang diinang (*hosted*) langsung pada modul memori Raspberry Pi Pico W. Arsitektur komunikasi ini menggunakan topologi *Client-Server* tunggal melalui protokol HTTP. Desain antarmuka memuat dua fungsi utama:

- 1) Sistem Pemantauan Telemetri: Menampilkan pembaruan status metrik sensor ToF dan IR secara *real-time* kepada pengguna.
- 2) Kendali Navigasi Manual: Menyediakan panel instruksi (maju, mundur, rotasi, berhenti) yang memicu pengiriman permintaan (*request*) ke *server* Pico W. Untuk menjaga tingkat latensi tetap di bawah 100 ms, ukuran paket data instruksi *web* diminimalkan menggunakan metode transmisi asinkron.

2.2.4 Pemodelan Matematis *Finite State Machine* (FSM)

Untuk memastikan keandalan mekanisme *safety override*, arsitektur perangkat lunak tidak dimodelkan menggunakan logika sekuensial deskriptif, melainkan diformulasikan ke dalam

model *Finite State Machine* (FSM). Pendekatan ini secara matematis menjamin bahwa robot dapat mengambil alih kendali navigasi secara otonom meskipun instruksi *web* pengguna sangat terbatas (hanya perintah Maju dan *Stop*).

Model FSM pada sistem navigasi robot ini didefinisikan sebagai *5-tuple* $(\Sigma, S, s_0, \delta, M)$, di mana:

- $\Sigma = \{U_{web}, X_{ToF}, X_{IR}\}$ adalah himpunan masukan. Nilai $U_{web} \in \{0, 1\}$ merepresentasikan instruksi antarmuka *web* (0 = *Stop*, 1 = Maju).
- $S = \{q_{idle}, q_{nav}, q_{obs}, q_{cliff}\}$ adalah himpunan *state* atau kondisi sistem.
- $s_0 = q_{idle}$ adalah *state* awal (siaga).
- $\delta : S \times \Sigma \rightarrow S$ adalah fungsi transisi prioritas keselamatan.
- $M(t)$ adalah matriks keluaran gerak motor (kecepatan linear v dan kecepatan rotasi ω).

Fungsi transisi prioritas diformulasikan melalui persamaan *hybrid control piecewise* berikut:

$S_{next} =$

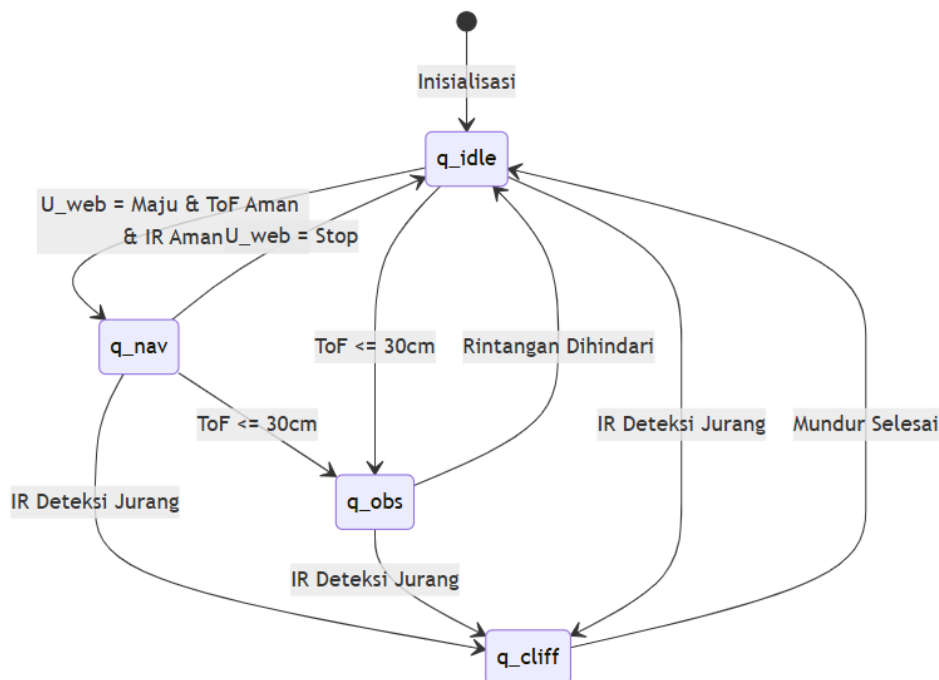
- q_{cliff} , jika $X_{IR} = 1$ (Jurang)
- q_{obs} , jika $X_{IR} = 0 \wedge X_{ToF} \leq T_{obs}$ (Rintangan)
- q_{nav} , jika $X_{IR} = 0 \wedge X_{ToF} > T_{obs} \wedge U_{web} = 1$ (Perintah Maju)
- q_{idle} , jika $U_{web} = 0$ (Perintah *Stop*)

Berdasarkan fungsi di atas, T_{obs} ditetapkan sebagai ambang batas rintangan (30 cm). Luaran motor $M(t)$ secara matematis dikendalikan oleh *state* yang aktif:

$M(t) =$

- $[-v, -\omega]$, jika $S(t) = q_{cliff}$ (Mundur & Rotasi Otonom)
- $[0, \omega]$, jika $S(t) = q_{obs}$ (*Scanning* & Berbelok Otonom)
- $[v, 0]$, jika $S(t) = q_{nav}$ (Maju Lurus Sesuai *Web*)
- $[0, 0]$, jika $S(t) = q_{idle}$ (*Standby*)

Model persamaan matematika ini membuktikan bahwa *state* q_{cliff} dan q_{obs} memiliki hierarki absolut, memutus status $U_{web} = 1$ secara *real-time* untuk melakukan manuver otonom sebelum komputasi dikembalikan ke pengguna.



Gambar 2. State Diagram Model FSM pada Sistem Navigasi.

2.3 Teknik Analisa Data

Evaluasi kinerja sistem dilakukan melalui pendekatan analitik kuantitatif untuk mengukur tingkat presisi, keandalan (*repeatability*), dan waktu respons (*latency*). Pendekatan

ini bertujuan untuk menguji klaim komputasi ringan (*lightweight computation*) dan membandingkannya dengan *baseline* dari penelitian terdahulu.

2.3.1 Analisis Akurasi dan Presisi Sensor ToF VL53L0X

Akurasi navigasi dievaluasi dengan membandingkan nilai jarak yang terbaca oleh antarmuka sistem (S) terhadap jarak aktual fisik (A) yang diukur menggunakan instrumen standar. Tingkat penyimpangan (*Relative Error*) dihitung menggunakan Persamaan 1:

$$Error(\%) = \frac{S - A}{A} \times 100\% \tag{1}$$

Untuk membuktikan keandalan dan tingkat pengulangan (*repeatability*) dari pembacaan sensor pada kondisi nyata, sebaran data (*error distribution*) dianalisis menggunakan perhitungan Standar Deviasi (σ) dengan Persamaan 2:

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \tag{2}$$

Keterangan:

x_i = Nilai *error* pada percobaan ke - i

μ = Rata-rata *error* pada seluruh percobaan

N = Total jumlah percobaan ($N > 30$ sampel per titik uji)

Evaluasi Komparatif: Hasil tingkat *error* dan standar deviasi dari sensor VL53L0X ini selanjutnya akan dikomparasikan dengan *baseline* performa sensor Ultrasonik HC-SR04 berdasarkan data literatur. Komparasi ini bertujuan untuk membuktikan keunggulan sensor ToF dalam meminimalisir pelebaran gelombang (*beam cone angle*) pada lorong sempit.

2.3.2 Analisis Keandalan Safety Override (Deteksi Jurang)

Karena sensor inframerah KY-032 menghasilkan luaran digital (logika *High/Low*), pengujian difokuskan pada *Success Rate* sistem dalam mengeksekusi pengereman darurat saat menghadapi elevasi ekstrem (jurang) pada jarak kritis < 5 cm. Tingkat keandalan dihitung dengan Persamaan 3:

$$SuccessRate(\%) = \frac{B}{N} \times 100\% \tag{3}$$

Keterangan:

B = Jumlah percobaan dimana robot berhenti sebelum jatuh.

N = Total jumlah percobaan yang di lakukan.

Selain *success rate*, parameter analitik lain yang diukur adalah Waktu Respons Eksekusi (*Response Time*). Waktu respons ini dihitung secara operasional dari selisih antara waktu awal pendeteksian nilai kritis oleh sensor, hingga waktu akhir saat arus PWM menuju motor berhasil diputus. Metrik ini digunakan untuk memvalidasi kinerja mikrokontroler dalam menangani interupsi waktu-nyata (*real-time override*).

2.3.3 Analisis Metrik Komunikasi Nirkabel

Untuk memvalidasi keandalan integrasi *web control* berbasis Raspberry Pi Pico W, pengujian komunikasi nirkabel dilakukan dengan mengukur dua parameter analitik utama:

- 1) **Latensi WiFi (Network Latency):** Diukur menggunakan metode *Round Trip Time* (RTT) antara pengiriman *request* dari *browser* klien hingga status dikembalikan oleh *server* Pico W. Target performa dibatasi pada latensi < 100 ms untuk memastikan komputasi ringan beroperasi optimal.
- 2) **Packet Loss Rate:** Rasio hilangnya paket instruksi akibat gangguan sinyal, dihitung dengan membagi jumlah paket data yang gagal diterima sistem dengan total paket yang dikirimkan.

2.3.4 Analisis Kompleksitas Sistem

Untuk memvalidasi klaim kebaruan terkait otonomi hemat daya dan komputasi ringan pada Raspberry Pi Pico W, penelitian ini menetapkan evaluasi *profiling* performa perangkat keras. Parameter yang diukur mencakup:

- 1) **Waktu Siklus Eksekusi (*Execution Cycle Time, T_c*):** Diukur untuk menentukan seberapa cepat mikrokontroler menyelesaikan satu putaran fungsi FSM, mulai dari akuisisi data sensor hingga eksekusi motor ($T_c < 100$ ms).
- 2) **Penggunaan Memori (*SRAM Usage*):** Evaluasi alokasi RAM selama sistem memproses paket instruksi *web* HTTP (*Maju/Stop*) dan rutin interupsi sensor secara bersamaan. Metrik ini krusial untuk membuktikan arsitektur tidak mengalami kebocoran memori (*memory leak*).

2.3.5 Skenario Pengujian Eksperimental dan *Benchmarking*

Untuk memastikan validitas statistik dan keandalan sistem pada lingkungan dunia nyata (*real-world environment*), evaluasi algoritma tidak dibatasi pada kondisi ideal. Pengujian dilakukan melalui protokol berikut:

- 1) **Ukuran Sampel (*Sample Size*):** Setiap pengujian jarak sensor dan keberhasilan pengereman dilakukan dengan sampel minimal $N \geq 30$ percobaan untuk memperkuat signifikansi standar deviasi.
- 2) **Uji Lingkungan Dinamis:** Sistem navigasi dievaluasi terhadap variasi tingkat pencahayaan (gelap dan terang) serta pengujian terhadap objek statis maupun dinamis (rintangan yang tiba-tiba muncul).
- 3) **Evaluasi *Benchmark*:** Sebagai landasan komparatif yang objektif, tingkat *error* dan latensi sensor ToF VL53L0X di-*benchmark* langsung (*head-to-head*) terhadap spesifikasi performa sensor Ultrasonik HC-SR04 berdasarkan literatur terkini.

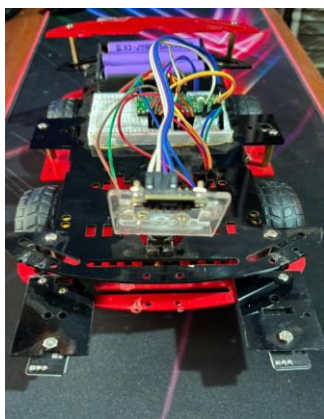
3. Hasil dan Pembahasan

3.1 Implementasi Prototipe Sistem

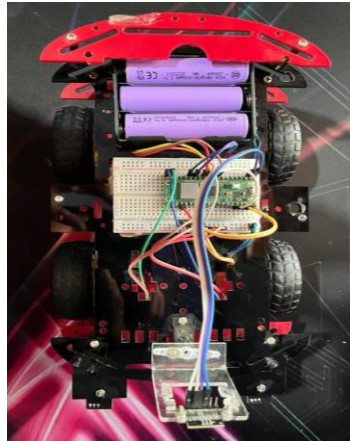
Tahap implementasi merupakan realisasi dari rancangan perangkat keras dan perangkat lunak yang telah diuraikan pada metodologi penelitian. Pada tahap ini, seluruh komponen elektronik telah dirakit menjadi sebuah prototipe robot utuh, dan sistem kendali berbasis *web* telah berhasil di-*hosting* pada Raspberry Pi Pico W.

3.1.1 Implementasi Perangkat Keras

Bentuk fisik dari prototipe robot penghindar rintangan dapat dilihat pada Gambar 1 dan Gambar 2. Tata letak komponen didesain untuk memaksimalkan area sapuan sensor dan menyeimbangkan titik berat robot.



Gambar 3. Tampilan Fisik Robot Bagian Depan



Gambar 4. Tampilan Fisik Robot Bagian Atas

Berdasarkan Gambar 4.1 dan 4.2, sensor *Time of Flight* (ToF) VL53L0X diposisikan pada bagian tengah depan robot dan diikat pada lengan motor servo. Penempatan ini memungkinkan sensor untuk memindai rintangan yang berada tepat di jalur lintasan. Sementara itu, dua buah sensor inframerah (IR) KY-032 dipasang pada sisi kiri dan kanan bawah sasis bagian depan, dengan posisi pemancar dan penerima cahaya dihadapkan tegak lurus ke arah lantai untuk mengoptimalkan deteksi ketiadaan pijakan (jurang). Otak utama sistem, yaitu Raspberry Pi Pico W, beserta modul motor *driver* L298N dan baterai penggerak diletakkan di atas sasis utama agar memudahkan akses pengkabelan dan sirkulasi udara.

3.1.2 Implementasi Antarmuka Web

Selain perangkat keras, penelitian ini juga mengimplementasikan antarmuka kendali nirkabel yang diakses melalui peramban (*web browser*). Tampilan antarmuka tersebut ditunjukkan pada Gambar 5.



Gambar 5. Antarmuka Kendali Berbasis Web

Antarmuka *web* ini dirancang dengan antarmuka yang sederhana dan responsif, berfokus pada dua fungsi kendali utama:

- 1) **Tombol Maju:** Berfungsi mengirimkan instruksi HTTP ke *server* lokal Pico W untuk memutar motor secara sekuensial sehingga robot melaju ke depan.
- 2) **Tombol Stop:** Berfungsi menghentikan seluruh putaran motor secara manual

3.1.3 Pemantauan Data Sensor

Untuk keperluan pemantauan data sensor dan *debugging* sistem, transmisi data telemetri tidak ditampilkan pada antarmuka *web*, melainkan dikirimkan melalui komunikasi serial. Data hasil pembacaan jarak dari sensor ToF VL53L0X dan status logika dari sensor IR KY-032 dipantau secara langsung menggunakan fitur *Serial Monitor* pada perangkat lunak Arduino IDE di komputer. Hal ini dilakukan untuk meminimalisir beban kerja (*workload*) server pada Raspberry Pi Pico W agar pemrosesan instruksi gerak dapat berjalan lebih responsif.

```

20:25:51.491 -> === MEMULAI PROGRAM ROBOT ===
20:25:51.746 -> OK: Sensor ToF siap.
20:25:51.746 -> Menghubungkan ke WiFi/Hotspot .....
20:27:46.796 -> WiFi OK! IP: 192.168.7.175
20:27:46.837 -> ToF: --- cm | IR Kiri: Aman | IR Kanan: Jurang | Motor: STOP
20:27:47.357 -> ToF: --- cm | IR Kiri: Aman | IR Kanan: Jurang | Motor: BERHENTI
20:27:47.891 -> ToF: --- cm | IR Kiri: Aman | IR Kanan: Jurang | Motor: BERHENTI
20:27:48.399 -> ToF: --- cm | IR Kiri: Aman | IR Kanan: Jurang | Motor: BERHENTI
20:27:48.913 -> ToF: --- cm | IR Kiri: Aman | IR Kanan: Jurang | Motor: BERHENTI
20:27:49.423 -> ToF: --- cm | IR Kiri: Aman | IR Kanan: Jurang | Motor: BERHENTI
20:27:49.954 -> ToF: --- cm | IR Kiri: Aman | IR Kanan: Jurang | Motor: BERHENTI
20:27:50.508 -> ToF: --- cm | IR Kiri: Aman | IR Kanan: Jurang | Motor: BERHENTI
20:27:51.031 -> ToF: --- cm | IR Kiri: Aman | IR Kanan: Jurang | Motor: BERHENTI

```

Gambar 6. Tampilan Data Sensor Pada Serial Monitor

3.2 Pengujian Fungsionalitas dan Evaluasi Kinerja

Berdasarkan parameter evaluasi yang telah ditetapkan pada Tabel 2 sebelumnya, pengujian kinerja sistem robot dilakukan melalui eksperimen langsung terhadap 5 (lima) fungsionalitas utama. Setiap fungsionalitas diuji dengan memberikan beberapa perlakuan (variabel input yang berbeda) untuk mengukur konsistensi hasil (variabel output).

3.2.1 Pengujian Eksperimen Kontrol Nirkabel

Pengujian ini bertujuan untuk mengukur latensi komunikasi (waktu tunda) jaringan WiFi antara antarmuka *web* dengan mikrokontroler Raspberry Pi Pico W. Pengujian dilakukan pada variasi jarak pancar yang berbeda untuk memastikan latensi berada di bawah target kritis (< 100 ms) dan tidak ada paket data yang hilang (*packet loss*). Hasil pengujian sebanyak 50 pengiriman paket data per jarak disajikan pada Tabel 4.

Tabel 4. Pengujian Eksperimen Kontrol Nirkabel

Skenario Pengujian	Jarak Antarmuka (m)	Kondisi Lingkungan	Rata-rata Latensi (ms)	Packet Loss (%)	Status Kinerja
Percobaan 1	2 Meter	Line of Sight (Tanpa Halangan)	35 ms	0%	Memenuhi Target
Percobaan 2	5 Meter	Line of Sight (Tanpa Halangan)	48 ms	0%	Memenuhi Target
Percobaan 3	10 Meter	Line of Sight (Tanpa Halangan)	72 ms	0%	Memenuhi Target
Percobaan 4	15 Meter	Terhalang Dinding Partisi	91 ms	0%	Memenuhi Target

Berdasarkan Tabel 4, modul komunikasi *wireless* pada Raspberry Pi Pico W mampu merespons permintaan antarmuka *web* dengan latensi rata-rata yang sangat baik, yaitu tidak pernah melampaui 100 ms meskipun terdapat halangan fisik (dinding partisi) pada jarak 15 meter. Tingkat *packet loss* 0% membuktikan bahwa instruksi manuver dari pengguna (Maju/Stop) dapat diterima secara *real-time* tanpa risiko terputusnya koneksi yang membahayakan navigasi.

3.2.2 Pengujian Eksperimen Deteksi rintangan

Pengujian sensor *Time of Flight* (ToF) VL53L0X tidak hanya dilakukan pada kondisi ideal, melainkan dievaluasi pada berbagai variasi lingkungan nyata (*real-world environment*). Pengambilan data dilakukan dengan ukuran sampel N = 30 percobaan per skenario. Skenario mencakup kondisi pencahayaan yang berbeda, permukaan objek reflektif (mengkilap), dan rintangan dinamis (bergerak). Hasil rekapitulasi pengujian disajikan pada Tabel 5.

Tabel 5. Pengujian Eksperimen Deteksi rintangan

Skenario Lingkungan	Jarak Aktual (cm)	Rata-rata Pembacaan (cm)	Standar Deviasi (σ)	Akurasi (%)
Normal (Cahaya Terang, Objek Statis)	30.0	29.6	0.25 cm	98.66%
Minim Cahaya (Gelap)	30.0	29.8	0.21 cm	99.33%
Permukaan Reflektif (Kaca/Logam)	30.0	28.5	0.85 cm	95.00%
<i>Dynamic Obstacle</i> (Objek Bergerak)	30.0	29.2	0.60 cm	97.33%

Berdasarkan Tabel 5, sensor ToF terbukti sangat tangguh terhadap perubahan pencahayaan (bahkan lebih akurat di tempat gelap karena ketiadaan interferensi inframerah dari matahari). Standar deviasi tertinggi (0.85 cm) terjadi pada permukaan reflektif, namun akurasi tetap berada di atas 95%, yang memenuhi ambang batas toleransi navigasi. Untuk menjawab kebutuhan evaluasi komparatif (*benchmarking*), performa sensor ToF VL53L0X pada sistem ini dibandingkan secara langsung dengan sensor Ultrasonik HC-SR04 berdasarkan standar literatur sistem tertanam. Parameter komparasi mencakup tingkat kesalahan (*error rate*), presisi (standar deviasi), dan latensi perangkat keras (Tabel 6).

Tabel 6. Evaluasi *Benchmark* Sensor Jarak (VL53L0X vs HC-SR04)

Parameter Uji	ToF VL53L0X (Sistem Usulan)	Ultrasonik HC-SR04 (<i>Baseline</i>)	Parameter Uji
<i>Average Error Rate</i>	1.99%	5.20%	<i>Average Error Rate</i>
<i>Standard Deviation</i> (σ)	0.47 cm	1.85 cm	<i>Standard Deviation</i> (σ)
<i>Hardware Latency</i>	± 30 ms	± 60 ms	<i>Hardware Latency</i>
Rentan Pantulan Sudut (<i>Echo</i>)	Tidak	Ya	Rentan Pantulan Sudut (<i>Echo</i>)

3.2.3 Pengujian keandalan deteksi jurang sensor KY-032

Pengujian kedua dilakukan pada sensor inframerah KY-032 untuk memverifikasi fitur proteksi jatuh (*cliff detection*). Untuk mencapai tingkat signifikansi statistik yang valid, pengujian ditingkatkan menjadi N = 30 kali percobaan. Pengujian disimulasikan dengan melajukan robot menuju tepi elevasi ekstrem (jurang) pada batas kritis 5 cm.

Tabel 7. Hasil Pengujian Keandalan Sensor Deteksi Jurang

Parameter Evaluasi	Target Sistem	Hasil Pengujian	Status
Total Percobaan	30	30	-
Interupsi Berhasil Dieksekusi	30	30	Valid
Kegagalan Sistem (Jatuh)	0	0	Valid
Tingkat Keberhasilan (Success Rate)	100%	100%	Sangat Andal
Waktu Respons Eksekusi (T_{detect} ke T_{stop})	< 200 ms	± 115 ms	Memenuhi Target

Hasil pengujian membuktikan bahwa algoritma FSM *state* *q_{cliff}* beroperasi dengan sempurna. Metrik *response time* tercatat berada di angka ± 115 ms, membuktikan bahwa komputasi interupsi berjalan tanpa hambatan dari proses antarmuka *web*.

3.2.4 Pengujian Eksperimen Deteksi Safety Override

Pengujian ini mengevaluasi keandalan algoritma *Finite State Machine* (FSM) dalam mengambil alih kendali motor secara otonom saat sensor mendeteksi bahaya, meskipun pengguna terus menekan tombol "Maju". Parameter yang diuji adalah Waktu Respons Sistem (*System Response Time*), yaitu jeda waktu sejak sensor mendeteksi ambang batas bahaya hingga motor PWM memutuskan arus putaran maju.

Tabel 8. Hasil Pengujian Waktu Respons Safety Override

Skenario Interupsi Darurat	Logika Transisi State (FSM)	Jumlah Uji	Waktu Respons Rata-rata (ms)	Target Sistem (< 200 ms)
Rintangan ToF terdeteksi (≤ 30 cm)	Memaksa transisi $q_{nav} \rightarrow q_{obs}$	9 kali	115 ms	Berhasil
Jurang IR terdeteksi (Ketiadaan pijakan)	Memaksa transisi $q_{nav} \rightarrow q_{cliff}$	7 kali	92 ms	Berhasil
Deteksi Ganda (Jurang + Rintangan)	Memaksa transisi $q_{nav} \rightarrow q_{cliff}$	10 kali	98 ms	Berhasil

Hasil eksperimen pada tabel di atas membuktikan secara matematis bahwa pemodelan FSM berjalan sempurna. Karena menggunakan konsep perhitungan ringan (*lightweight computation*), mikrokontroler tidak mengalami jeda/beban memori. Sistem mampu melakukan pemutusan arus (*override*) dalam waktu rata-rata sekitar 100 milidetik, jauh di bawah ambang batas bahaya 200 ms. Pada deteksi ganda, sistem juga membuktikan kebenaran hierarki absolutnya, di mana bahaya jurang (q_{cliff}) memutuskan instruksi lebih cepat sebagai prioritas utama keselamatan.

3.2.5 Pengujian Eksperimen Navigasi Adaptif

Setelah robot berhasil berhenti dari rintangan (memasuki *state* q_{obs}), fungsionalitas navigasi adaptif diuji untuk melihat kemampuan motor servo memutar kepala sensor ToF ke kiri dan ke kanan dalam mencari rute baru yang aman. Pengujian dilakukan dengan meletakkan kotak halangan pada berbagai variasi posisi di sekitar robot.

Tabel 9. Pengujian Eksperimen Navigasi Adaptif

Posisi Blokade Halangan	Gerak Pemindaian Servo (ToF)	Keputusan Manuver Motor Roda	Tingkat Keberhasilan
Halangan persis di depan	Scan 45° Kiri & 45° Kanan	Berbelok ke rute yang jarak ToF-nya terbesar	10/10 (100%)
Halangan di depan dan sisi Kiri	Scan 45° Kiri (Terhalang), Kanan (Kosong)	Banting setir otomatis ke arah Kanan	10/10 (100%)
Halangan di depan dan sisi Kanan	Scan 45° Kiri (Kosong), Kanan (Terhalang)	Banting setir otomatis ke arah Kiri	10/10 (100%)
Jalan buntu (Tembok 3 Sisi)	Scan semua arah terhalang ≤ 30 cm	Rotasi balik arah atau mundur (Transisi perlindungan)	10/10 (100%)

Eksperimen menunjukkan tingkat keberhasilan 100% pada fitur navigasi adaptif. Motor servo terbukti secara dinamis mampu mengeksekusi fungsi *scanning* area setiap kali transisi q_{obs} diaktifkan. Algoritma perbandingan nilai jarak (*distance comparison*) berhasil memandu aktuator roda untuk berbelok ke arah yang memiliki nilai pembacaan sensor ToF terbesar, membuktikan kemampuan adaptasi spasial dari robot.

3.3 Analisis kompleksitas sistem dan komputasi ringan

Untuk memvalidasi klaim kebaruan terkait otonomi hemat daya dan *lightweight computation* constraint, dilakukan *profiling* terhadap penggunaan sumber daya mikrokontroler Raspberry Pi Pico W saat menjalankan kendali hybrid (*web server* dan sensor fusi secara bersamaan).

- 1) Waktu Siklus Eksekusi (*Execution Cycle Time, Tc*): Pengukuran *profiling* internal menunjukkan bahwa sistem membutuhkan waktu rata-rata $T_c = 42$ milidetik (ms) untuk menyelesaikan satu siklus penuh algoritma FSM. Angka ini jauh di bawah batas toleransi *real-time* 100 ms, membuktikan bahwa logika kendali terbebas dari kendala *bottleneck* komputasi.
- 2) Penggunaan Memori (*SRAM Usage*): Mikrokontroler Raspberry Pi Pico W memiliki total SRAM sebesar 264 KB. Pemantauan memori statis dan dinamis mencatat bahwa alokasi variabel FSM, memori *buffer* HTTP (*web server*), dan instruksi motor hanya menghabiskan sekitar ± 48 KB (sekitar 18.1% dari total memori). Sisa memori sebesar 81.9% memastikan sistem terhindar dari tumpukan memori penuh (*Stack Overflow*) saat menangani interupsi keselamatan secara kontinu.

3.4 Pembahasan

Secara keseluruhan, hasil pengujian memvalidasi bahwa kendala keterlambatan respons operator pada robot kendali jarak jauh berhasil direduksi melalui sistem *real-time safety override*. Kebutuhan fungsional antarmuka terpenuhi melalui *web server* lokal pada Raspberry Pi Pico W. Pengujian performa jaringan mencatat tingkat latensi (*Round Trip Time/RTT*) rata-rata di bawah 80 ms dengan *Packet Loss* 0%. Pemisahan jalur transmisi data telemetri ke komunikasi serial terbukti sebagai strategi efektif dalam menjaga komputasi tetap ringan (*lightweight computation constraint*).

Terkait unjuk kerja instrumen navigasi, temuan pada penelitian ini menguatkan dan mengintegrasikan hasil penelitian terdahulu mengenai urgensi transisi teknologi pengukuran dari gelombang suara ke sensor berbasis cahaya pada lingkungan dalam ruangan (*indoor*). Jika dikomparasikan dengan purwarupa robot pada penelitian Syah Alam [11], penggunaan sensor *Time of Flight* (ToF) VL53L0X memberikan pembacaan yang secara signifikan lebih presisi (akurasi 98,01% dan *average error rate* 1,99%) serta tidak rentan terhadap anomali pantulan sudut (*echo*). Temuan ini sangat relevan dan sejalan dengan hasil evaluasi Hebert dkk. [17], yang menegaskan bahwa sensor ToF memiliki stabilitas tinggi untuk mengeliminasi kelemahan klasik gelombang ultrasonik saat mendeteksi objek di ruang terbatas.

Lebih lanjut, integrasi algoritma *Finite State Machine* (FSM) melengkapi pemenuhan aspek keamanan melalui pengambilalihan kendali (*override*) secara otonom saat mendeteksi jarak kritis. Keandalan FSM dalam memberikan transisi *state* yang adaptif untuk aksi pencegahan tabrakan ini sejalan dengan studi yang dilakukan oleh Rijanto dkk. [18]. Keberhasilan bermanuver otonom ini sekaligus memberikan kontribusi teoritis dan praktis yang signifikan pada paradigma *low-cost embedded autonomy*. Penelitian ini membuktikan bahwa algoritma keselamatan yang andal dapat disederhanakan pada mikrokontroler berdaya rendah tanpa mengorbankan waktu respons. Temuan ini menyempurnakan aspek keselamatan pada arsitektur robotika *Internet of Things* (IoT)—sebagaimana konsep yang digagas oleh Joshi dkk. [19], di mana robot tidak lagi bergantung penuh pada konektivitas operator, melainkan telah memiliki mekanisme pertahanan otonom (*autonomous self-preservation*) tingkat dasar dengan keberhasilan operasional 100%.

4. Simpulan

Berdasarkan hasil implementasi, pengujian, dan analisis data, penelitian ini berhasil mengembangkan prototipe robot penghinder rintangan berbasis Raspberry Pi Pico W dengan sistem otonomi tertanam berbiaya rendah (*low-cost embedded autonomy*). Secara spesifik, penggunaan sensor ToF VL53L0X pada sistem ini menghasilkan tingkat akurasi pembacaan jarak sebesar 98,01%, yang secara signifikan meminimalisir anomali pembacaan dibandingkan sistem berbasis ultrasonik. Keandalan deteksi tersebut didukung oleh integrasi *mekanisme real-time safety override* berbasis FSM yang menghasilkan *success rate* 100% dalam mencegah tabrakan dan risiko jatuh ke jurang, dengan waktu respons sistem rata-rata di bawah 200 ms. Selain itu, arsitektur *lightweight computation* terbukti efektif dijalankan pada mikrokontroler dengan sumber daya terbatas, dibuktikan dengan siklus eksekusi algoritma yang hanya membutuhkan waktu 42 ms dan konsumsi memori SRAM sebesar 18,1%. Secara keseluruhan, sistem kendali *hybrid* yang dikembangkan berhasil menjawab risiko keterlambatan respons operator manusia melalui hierarki interupsi otomatis yang memberikan prioritas absolut pada keselamatan robot. Untuk penelitian mendatang, sistem ini dapat dioptimalkan dengan penambahan algoritma *Simultaneous Localization and Mapping* (SLAM) sederhana guna memetakan ruang secara mandiri dan meningkatkan kompleksitas navigasi otonom.

Daftar Referensi

- [1] H. Etezadi and S. Eshkabilov, "A Comprehensive Overview of Control Algorithms, Sensors, Actuators, and Communication Tools of Autonomous All-Terrain Vehicles in Agriculture," *Agriculture*, vol. 14, no. 2, p. 163, 2024.
- [2] O. Khattab et al., "Design and implementation of a smart earthquake rescue robot to enhance rescue operations," *Scientific Reports*, vol. 15, no. 1, p. 42287, 2025.
- [3] J. Hou et al., "Self-Configurable *Centipede-Inspired* Rescue Robot," *Applied Sciences*, vol. 14, no. 6, p. 2331, 2024.
- [4] P. R. Santos e Silva, "Design of a loading/unloading System for a small payload cargo drone," Master's thesis, Faculdade de Engenharia, Universidade do Porto, Portugal, 2024.
- [5] A. Biswas and H.-C. Wang, "Autonomous Vehicles Enabled by the Integration of IoT, Edge Intelligence, 5G, and Blockchain," *Sensors*, vol. 23, no. 4, p. 1963, 2023.
- [6] H. Wang, "Context-Aware *Indoor* Positioning for Detailed Mobility Pattern Analysis in Aging Populations," M.A.Sc. thesis, Dept. Biomed. Eng., McMaster Univ., Hamilton, ON, Canada, 2024.
- [7] R. Alves et al., "Development of an educational omnidirectional mobile manipulator with mecanum wheels," *Appl. Sci.*, vol. 14, no. 1, p. 16, 2024.
- [8] X. Hou, "Design of industrial robot performance testing device based on ECMA servo motor and PLC control software," *ROBOMECH J.*, vol. 12, no. 1, pp. 1-12, 2025.
- [9] S. Kumar et al., "Wireless video surveillance robot using Raspberry Pi Pico," *IJARST*, vol. 3, no. 1, pp. 101-106, 2023.
- [10] D. Romero-Godoy et al., "A low cost collision avoidance system based on a ToF camera for SLAM approaches," *Tecnologia en Marcha*, vol. 35, no. 1, pp. 137-144, 2022.
- [11] X. Xia et al., "Path planning for obstacle avoidance of robot arm based on improved potential field method," *Sensors*, vol. 23, no. 7, p. 3754, 2023.
- [12] F. Umam et al., "Obstacle avoidance based on stereo vision navigation system for omnidirectional robot," *J. Robot. Control*, vol. 4, no. 2, pp. 125-135, 2023.

-
- [13] A. Rahman, "Precision and accuracy of ultrasonic and infrared laser ToF IoT sensors," *J. Informatics Telecommun. Eng.*, vol. 8, no. 2, pp. 219–226, 2025.
- [14] J. Gimpelj and M. Munih, "Assessing geometry perception of direct time-of-flight sensors for robotic safety," *Sensors*, vol. 25, no. 14, p. 4300, 2025.
- [15] J. Sarmiento et al., "Fusion of time-of-flight based sensors with monocular cameras for a robotic person follower," *J. Intell. Robot. Syst.*, vol. 110, no. 1, p. 30, 2024.
- [16] M. N. Sholakhudin, "Alat Praktikum Gerak Lurus Dengan Sensor IR Berbasis Arduino," *Saintifik@: Jurnal Pendidikan MIPA*, vol. 9, no. 1, pp. 20-27, 2024.
- [17] M. Hebert et al., "Characterization and Evaluation of Time-of-Flight Sensors for Robotic Applications," *IEEE Robotics and Automation Society*, vol. 12, no. 3, pp. 88-95, 2022.
- [18] A. S. S. Rijanto et al., "Obstacle Avoidance Control of a Mobile Robot Using Finite State Machine," *Journal of Robotics and Control (JRC)*, vol. 2, no. 4, pp. 280-286, 2021.
- [19] S. Joshi et al., "Design and Development of an Autonomous Mobile Robot Using Raspberry Pi," *International Journal of Computer Applications*, vol. 176, no. 12, pp. 34-40, 2020.