

# IMPLEMENTASI FUZZY-WALL FOLLOWING DAN PID-OBJECT TRACKING PADA MOBILE ROBOT

Soni Hartadi<sup>1\*</sup>, Zahidan Akhmad N.<sup>2</sup>, Eko Budi P.<sup>3</sup>, Naufal Farros Z. A.<sup>4</sup>, Muharom Eko S.<sup>5</sup>, Ardy Seto Priambodo.<sup>6</sup>

<sup>1,2,3,4,5,6</sup> Universitas Negeri Yogyakarta; alamat; Jl. Colombo No.1, Karang Malang, Caturtunggal, Kec. Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55281; (0274) 586168

Received: xxxx-xx-xx

Accepted: xx-xx-xx

## Keywords:

Distributed Architecture;  
Wall Following;  
Object Tracking;  
Fuzzy Logic;  
PID;

## Correspondent Email:

[xxxxxxxxxx@xxxx.xxx](mailto:xxxxxxxxxx@xxxx.xxx)



JITET is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

**Abstrak.** Penelitian ini merancang dan mengimplementasikan sistem kendali hibrida pada robot bergerak yang mampu menjalankan dua fungsi sekaligus: mengikuti dinding (*wall-following*) dan melacak objek (*object tracking*). Sistem ini dibangun di atas platform tunggal dengan arsitektur terdistribusi yang mengintegrasikan Raspberry Pi untuk pemrosesan visi dan ESP32 untuk kontrol aktuator. Metode *wall-following* diimplementasikan menggunakan Logika Fuzzy berbasis data sensor jarak, sementara pelacakan objek visual dikendalikan oleh kontroler *Proportional-Integral-Derivative* (PID). Hasil pengujian pada berbagai skenario lintasan menunjukkan performa yang solid. Pada tugas *wall-following* di lintasan lurus, kontroler Fuzzy menunjukkan respons cepat dengan *rise time* 0.61 detik dan *overshoot* rendah sebesar 2.00%. Sementara itu, kontroler PID berhasil melacak dan menggrip target dengan hasil terbaik pada skenario belokan 90 derajat, yang ditunjukkan oleh *overshoot* terkontrol sebesar +14.06% dan *settling time* 6.50 detik. Kesimpulannya, arsitektur hibrida terdistribusi ini terbukti efektif dalam mengintegrasikan dua metode kontrol yang berbeda untuk menjalankan fungsi navigasi dan pelacakan secara simultan pada satu platform robot.

**Abstract.** This research designs and implements a hybrid control system for a mobile robot capable of simultaneously performing two functions: wall-following and object tracking. The system is built on a single platform with a distributed architecture that integrates a Raspberry Pi for vision processing and an ESP32 for actuator control. The wall-following method is implemented using Fuzzy Logic based on distance sensor data, while visual object tracking is controlled by a Proportional-Integral-Derivative (PID) controller. Test results across various trajectory scenarios show solid performance. In the wall-following task on a straight path, the Fuzzy controller demonstrates a fast response with a rise time of 0.61 seconds and a low overshoot of 2.00%. Meanwhile, the PID controller successfully tracks and grips the target with the best performance in the 90-degree turn scenario, indicated by a controlled overshoot of +14.06% and a settling time of 6.50 seconds. In conclusion, this distributed hybrid architecture is proven effective for integrating two different control methods to perform navigation and tracking functions simultaneously on a single robot platform.

## 1. PENDAHULUAN

Seiring pesatnya perkembangan teknologi saat ini, robotika telah menjadi bidang yang sangat berperan dalam berbagai aspek

pekerjaan manusia [2][3]. Pengembangan robot bergerak (mobile robot) menghadapi tantangan fundamental berupa navigasi otonom [8]. Di era kemajuan teknologi ini,

robot yang mampu bergerak secara mandiri telah banyak diterapkan dalam berbagai sendi kehidupan, mulai dari pengelolaan logistik di gudang, pengawasan lingkungan, hingga misi pencarian dan penyelamatan. Salah satu aspek penting dari robot otonom yaitu kemampuan untuk mendeteksi dan mengikuti objek tertentu. Kemampuan utama yang menjadi tantangan adalah navigasi yang andal di lingkungan yang kompleks dan selalu berubah [6][8]. Salah satu tugas navigasi fundamental adalah wall-following (mengikuti dinding) yang memungkinkan robot menjelajahi koridor atau menghindari rintangan, serta *object detection and tracking* (pendeteksian dan pelacakan objek) yang esensial untuk interaksi atau tugas spesifik seperti pengambilan barang [10]. Implementasi sistem kendali yang efektif dan efisien untuk kedua fungsi ini merupakan isu krusial dalam desain robot otonom.

Beberapa permasalahan utama yang sering dihadapi meliputi ketidakpastian data dari sensor (seperti gangguan pada sensor jarak), variasi kondisi lingkungan, kebutuhan akan respons yang cepat (secara real-time), dan keterbatasan sumber daya komputasi pada platform robot itu sendiri. Sistem kendali konvensional seringkali kurang optimal dalam menangani ketidakpastian dan sifat non-linear yang ada pada lingkungan nyata [7]. Untuk mengatasi hal tersebut, sistem kendali fuzzy menawarkan keunggulan karena kemampuannya menangani ketidakpastian dan melakukan penalaran berbasis aturan, menjadikannya pilihan yang baik untuk tugas wall-following yang adaptif [5][8]. Di sisi lain, sistem kendali PID (*Proportional-Integral-Derivative*) telah terbukti sangat efektif untuk tugas pelacakan (tracking) dan pendekatan objek (positioning) yang memerlukan presisi dan responsivitas tinggi. Penggunaannya terbukti dapat membuat robot lebih stabil dalam mengikuti target dan mampu memposisikan robot pada titik pengambilan objek dengan tepat [2].

Penelitian-penelitian sebelumnya telah banyak mengeksplorasi kedua jenis sistem kendali ini secara terpisah. Sebagai contoh, studi yang dilakukan oleh [1] D.I. Schema menunjukkan bahwa sistem kendali fuzzy handal untuk manuver wall-following yang stabil pada berbagai bentuk koridor. Penelitian lain oleh [2] M. N. F. Astoro, T. Winarno, dkk.

berhasil menunjukkan akurasi kendali PID dalam mengarahkan robot menuju target yang telah ditentukan. Meskipun demikian, penggabungan kedua sistem ini ke dalam satu arsitektur robot yang terpadu dan efisien masih menyisakan tantangan dan ruang untuk eksplorasi lebih dalam.

Secara spesifik, belum banyak penelitian yang mengintegrasikan fuzzy *wall-following* dan PID *object tracking* dalam satu platform *mobile robot* berbasis arsitektur terdistribusi (Raspberry Pi dan ESP32). Kesenjangan inilah yang menjadi fokus utama penelitian ini. Oleh karena itu, penelitian ini bertujuan untuk merancang, mengimplementasikan, dan menguji kinerja sistem kendali hibrida pada platform robot bergerak tunggal untuk menjalankan fungsi navigasi dan pelacakan objek secara simultan.

## 2. TINJAUAN PUSTAKA

### 2.1. Mobile Robot

Mobile robot merupakan robot yang mempunyai kapabilitas untuk melakukan pergerakan secara bebas pada sebuah lingkungan [6]. Mobile robot memiliki mekanisme yang dapat digunakan untuk berpindah posisi, seperti roda, kaki, atau trek [4][6]. Pengembangan berbagai jenis mobile robot ini diharapkan dapat memberikan kontribusi signifikan bagi manusia, terutama dalam mengotomatiskan sistem transportasi, menyediakan landasan bergerak untuk robot-robot industri, melakukan penjelajahan di area berbahaya tanpa kehadiran manusia, serta berbagai aplikasi lainnya [5].

### 2.2. Fuzzy Logic Controller

Sistem kendali fuzzy memetakan input numerik jarak dinding menjadi variabel linguistik melalui fuzzifikasi. Aturan IF-THEN diproses dalam inferensi, lalu diubah kembali ke nilai numerik melalui defuzzifikasi. Keunggulan utamanya adalah kemampuan menangani ketidakpastian sensor tanpa model matematis kompleks [1][2].

Implementasi dasar ini terus dikembangkan untuk meningkatkan performa dan adaptabilitas. Beberapa penelitian berfokus pada optimasi FLC. Sebagai contoh, Chen dkk. [6] mengintegrasikan reinforcement learning untuk menyempurnakan aturan fuzzy, sementara Štefek dkk. [12] menggunakan algoritma optimasi untuk menemukan

parameter keanggotaan fuzzy yang paling efektif untuk robot beroda diferensial.

### 2.3. Wall Follower

Metode navigasi wall follower memungkinkan robot bergerak dengan mengikuti kontur dinding pada jarak yang telah ditentukan tanpa menyentuhnya [1][9]. Dengan menggunakan sensor jarak seperti sensor laser VL53L0X di bagian sisi dan depan, robot mengumpulkan data tentang posisi dinding dan rintangan [4][10][14]. Data ini kemudian digunakan untuk mengendalikan kecepatan dan arah roda, dengan tujuan untuk mempertahankan robot pada jarak optimal dari dinding [11].

### 2.4. PID Controller

Sebuah sistem kontrol PID melibatkan tiga jenis pengaturan, yaitu kontrol Proportional (P), Integral (I), dan Derivative (D). Ketiganya dapat diimplementasikan secara individual maupun sebagai kombinasi [15]. Kendali PID menjadi pilihan yang dominan untuk tugas yang menuntut presisi tinggi seperti pelacakan dan penargetan objek karena responnya yang cepat dan kemampuannya untuk meminimalkan error secara sistematis [15]. Penelitian oleh Astoro dkk. [2] berhasil mengimplementasikan kendali PID untuk memposisikan robot secara akurat pada titik pengambilan objek berdasarkan data dari sensor kamera.

Efektivitas PID tidak terbatas pada positioning statis. Chu dkk. [15] menerapkan variasi kendali PID dalam kerangka Model Predictive Control (MPC) untuk pelacakan trajektori pada kendaraan otonom, yang membuktikan relevansinya dalam skenario dinamis. Menariknya, PID juga pernah dieksplorasi untuk tugas wall-following, seperti yang ditunjukkan oleh Sihombing dkk. [13]. Meskipun berhasil, pendekatan ini cenderung lebih kaku dan memerlukan tuning parameter yang cermat. Kegagalan dalam tuning dapat menyebabkan osilasi atau respons yang lambat.

### 2.5. Object Tracking

Object tracking melibatkan deteksi target dan estimasi posisi relatifnya terhadap robot. Pada robot mobile, sistem ini memerlukan fusi data vision sensor (koordinat piksel objek) dan kinematik robot (kecepatan roda) untuk menghasilkan trajectory pursuit yang smooth.

Tantangan utama dari object tracking adalah latensi pemrosesan citra dan perubahan iluminasi [2].

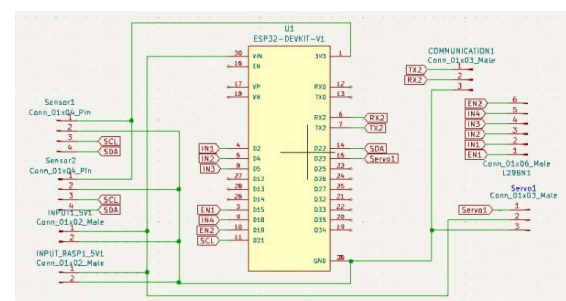
## 3. METODOLOGI PENGEMBANGAN DAN PENGUJIAN ROBOT

Proses yang digunakan untuk pengembangan sistem dalam proyek ini meliputi tahap desain sistem, implementasi perangkat lunak dan perangkat keras, integrasi modul fungsional, dan pengujian di seluruh sistem untuk memastikan seberapa efektif kinerja sistem.

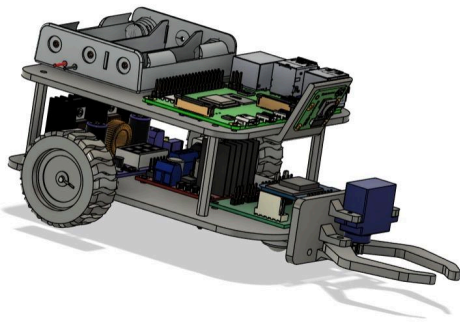
### 3.1. Arsitektur Sistem Robot

Arsitektur sistem robot yang dikembangkan dalam proyek ini menggunakan pendekatan komputasi terdistribusi, dengan membagi peran pemrosesan antara dua unit utama: Raspberry Pi dan ESP32. Pendekatan ini diterapkan untuk meningkatkan efisiensi kinerja, memungkinkan pemrosesan paralel, dan membedakan antara tugas komputasi tingkat tinggi dengan kendali waktu nyata pada perangkat keras.

Raspberry Pi berfungsi sebagai pusat pemrosesan tingkat tinggi. Perangkat ini menangani komputasi intensif, seperti pengolahan citra dari kamera untuk mendeteksi objek, serta perhitungan dan pengiriman parameter kontrol fuzzy dan PID. Sedangkan ESP32 bertindak sebagai pengendali utama untuk fungsi kendali waktu nyata. Modul ini bertugas membaca data dari sensor, seperti sensor jarak laser VL53L0X, serta mengendalikan aktuator seperti motor DC dan gripper melalui driver motor.



Gambar 1. Schematic Diagram



**Gambar 2.** Robot Design

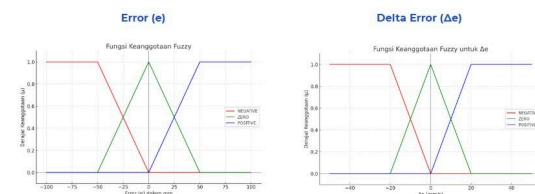
### 3.2. Fuzzy Logic Controller Design for Wall Following

Dalam penelitian ini, sistem pengendalian wall-following dirancang menggunakan pendekatan logika fuzzy, yang dikenal efektif untuk menangani ketidakpastian dan kondisi lingkungan yang dinamis serta non-linier tanpa memerlukan model matematis yang kompleks. Pendekatan ini sangat relevan diterapkan pada robot bergerak, khususnya untuk menjaga jarak konstan terhadap dinding dalam lingkungan nyata yang cenderung tidak ideal dan sering berubah-ubah.

Sistem pengendali fuzzy dirancang dengan dua variabel masukan, yaitu Error ( $e$ ) dan Delta Error ( $\Delta e$ ). Variabel Error mengukur selisih antara jarak aktual robot terhadap dinding dan jarak ideal yang telah ditetapkan. Sementara itu, Delta Error menunjukkan seberapa cepat perubahan error terjadi, yang menggambarkan apakah robot sedang bergerak mendekati atau menjauh dari dinding. Variabel keluaran dari sistem fuzzy ini adalah sinyal kendali yang digunakan untuk menyesuaikan kecepatan motor kiri dan kanan, dengan tujuan mempertahankan jarak ideal dari dinding.

Untuk setiap variabel masukan, ditetapkan tiga himpunan linguistik fuzzy, yaitu Negatif (N), Nol (Z), dan Positif (P). Interpretasi dari himpunan ini adalah sebagai berikut: Negatif berarti robot terlalu dekat dengan dinding, Positif berarti terlalu jauh, dan Nol berarti berada pada posisi yang ideal. Fungsi keanggotaan (membership function) yang digunakan merupakan kombinasi antara bentuk segitiga dan trapesium, seperti yang terlihat pada Gambar 3. Segitiga digunakan pada nilai pusat untuk memberikan sensitivitas tajam terhadap perubahan kecil di sekitar titik setel, sedangkan trapesium digunakan pada nilai

ekstrem untuk menghasilkan respons yang lebih stabil dan tidak terlalu sensitif terhadap noise. Pemilihan bentuk ini mengikuti pendekatan yang direkomendasikan dalam literatur, karena mampu memberikan keseimbangan antara sensitivitas dan stabilitas dalam lingkungan nyata yang dinamis.



**Gambar 3.** Membership Function

Parameter batas untuk fungsi keanggotaan ditentukan secara eksperimental, berdasarkan pengamatan terhadap rentang akurasi sensor jarak laser VL53L0X yang digunakan, yaitu sekitar 5–30 cm. Agar proses inferensi lebih konsisten dan independen dari skala fisik sensor, nilai masukan dinormalisasi ke dalam rentang -1 hingga +1.

Sistem pengendali fuzzy ini dibangun di atas basis aturan (rule base) yang terdiri dari 15 aturan logika IF–THEN, hasil kombinasi dari tiga nilai linguistik pada masing-masing variabel masukan. Contoh aturan yang digunakan antara lain: IF Error = Z AND Delta Error = Z THEN Output = Z, yang berarti jika posisi robot sudah ideal dan tidak ada perubahan, maka tidak perlu ada penyesuaian; IF Error = N AND Delta Error = N THEN Output = P, artinya jika robot terlalu dekat dan terus mendekat, maka kecepatan motor harus disesuaikan agar menjauh; dan IF Error = P AND Delta Error = P THEN Output = N, yang berarti robot terlalu jauh dan makin menjauh, sehingga harus diarahkan lebih dekat ke dinding. Penyusunan aturan ini mengikuti metode inferensi Mamdani klasik sebagaimana digunakan dalam berbagai sistem kendali fuzzy berbasis aturan, yang menunjukkan efektivitas tinggi pada robot dengan penggerak diferensial.

Proses inferensi dilakukan dengan metode Mamdani Min–Max, yaitu dengan mengambil nilai minimum dari derajat keanggotaan pada bagian premis (IF), dan kemudian menggabungkannya (union) pada bagian kesimpulan (THEN). Hal ini memungkinkan sistem mempertimbangkan banyak aturan

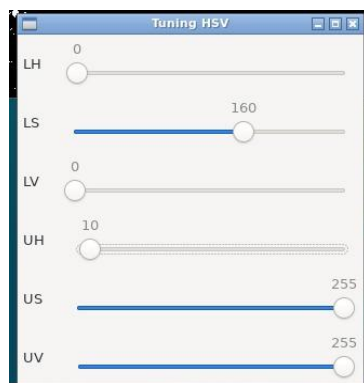


secara bersamaan untuk menghasilkan keluaran yang lebih halus dan logis

Setelah itu, dilakukan proses defuzzifikasi menggunakan metode Centroid of Area (COA), yaitu menghitung titik tengah dari area gabungan fungsi keanggotaan keluaran. COA dipilih karena dapat menghasilkan nilai keluaran yang kontinu, tidak meloncat, dan telah terbukti memberikan kestabilan yang baik dalam aplikasi kontrol robot wall-following.

### 3.3. Deteksi Objek menggunakan Computer Vision

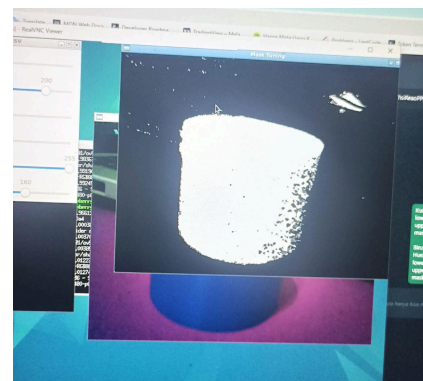
Untuk memudahkan interaksi dengan lingkungan, robot bergerak yang digunakan dalam penelitian ini dilengkapi dengan sistem deteksi objek berbasis visi komputer. Sistem ini dikembangkan dengan menggabungkan kamera yang dipasang pada robot yang memungkinkan pengenalan objek target berdasarkan atribut visual utama, yaitu warna dan, jika berlaku, bentuk. Gambar 4 menyajikan keseluruhan alur proses sistem deteksi objek yang diimplementasikan.



Gambar 4. Tuning HSV

Tahap awal dari proses deteksi melibatkan pengambilan gambar dari lingkungan robot secara langsung menggunakan kamera. Gambar ini kemudian dapat diambil melalui tahap pra-pemrosesan awal jika diperlukan, termasuk proses seperti penghalusan untuk menghilangkan gangguan visual. Metode segmentasi objek berdasarkan warna yang paling banyak digunakan adalah transformasi gambar dari model warna RGB (*Red, Green, Blue*) ke model warna HSV (*Hue, Saturation, Value*). Alasan mengapa model warna HSV dipilih adalah karena model ini memiliki

kemampuan untuk memisahkan intensitas cahaya (*Value*) dan saturasi warna (*Saturation*) dari komponen warna (*Hue*), sehingga memudahkan definisi warna yang dapat dipisahkan secara mudah. Untuk mengetahui warna objek target yang tepat, yaitu biru, batas bawah (LH, LS, LV) dan batas atas (UH, US, UV) ditetapkan untuk masing-masing dari tiga komponen HSV. Proses pengaturan atau kalibrasi nilai-nilai HSV ini, seperti yang terlihat pada Gambar 5, memungkinkan sistem untuk menghasilkan topeng biner yang benar. Pada topeng biner yang diberikan, piksel yang memiliki nilai HSV dalam rentang yang diberikan akan berwarna putih, menunjukkan wilayah minat yang menandakan objek yang diminati, dan semua piksel lainnya akan berwarna hitam, dengan demikian memisahkan objek dari latar belakang.



Gambar 5. Tracking object

### 3.4. PID Control untuk melacak objek.

Setelah objek target berhasil dideteksi oleh sistem visi komputer, tahap selanjutnya dalam alur kerja robot adalah melacak dan mendekati objek tersebut secara akurat. Untuk mencapai tujuan ini, proyek ini mengimplementasikan PID (Proportional-Integral-Derivative) Controller yang dirancang untuk mengatur kecepatan gerak robot, baik dari sisi arah maupun jarak terhadap objek.

#### 3.4.1. Definisi Kesalahan dan Tujuan Kontrol

Sistem pelacakan berbasis PID bekerja dengan menghitung dua jenis error utama berdasarkan data visual dari kamera:

**Error sudut ( $e_\theta$ ):** dihitung dari perbedaan posisi horizontal centroid objek terhadap titik tengah citra kamera. Tujuannya adalah menjaga agar robot selalu menghadap objek.

**Error linier ( $e_v$ ):** dihitung berdasarkan ukuran

objek dalam citra dibandingkan dengan ukuran target atau posisi vertikal objek. Kesalahan ini digunakan untuk mengatur jarak robot terhadap objek.

Kedua error tersebut diolah menggunakan rumus PID:

$$u(t) = K_p \cdot e(t) + K_i \int e(t) dt + K_d \cdot (de(t))/dt \quad (1)$$

Keluaran dari masing-masing kontrol PID berupa Kecepatan sudut ( $\omega$ ) dan Kecepatan linier ( $v$ ). Kedua output tersebut dikonversi menjadi perintah kecepatan untuk motor kiri dan kanan pada platform robot berbasis penggerak diferensial.

#### 3.4.2. Tuning Parameter PID (Metode Trial-and-Error)

Parameter PID yaitu  $K_p$ ,  $K_i$ , dan  $K_d$  ditentukan melalui proses trial-and-error, yakni eksperimen langsung berdasarkan pengamatan respons sistem terhadap perubahan error. Pendekatan ini digunakan karena karakteristik dinamis sistem robotik sulit dimodelkan secara matematis secara akurat. Langkah-langkah tuning dilakukan sebagai berikut:

##### 3.4.3. Inisialisasi Awal

Dimulai dengan nilai awal:  $K_p$  kecil (misalnya 0.1–0.3),  $K_i$  dan  $K_d$  diatur ke nol. Tujuannya adalah mengamati reaksi sistem hanya terhadap komponen proporsional.

##### 3.4.4. Tuning $K_p$ (Proporsional)

$K_p$  dinaikkan secara bertahap sampai robot menunjukkan respons yang cepat terhadap error, tetapi belum menimbulkan osilasi atau overshoot berlebih.

##### 3.4.5. Tuning $K_d$ (Differential)

Setelah  $K_p$  cukup baik, nilai  $K_d$  ditambahkan untuk meredam osilasi dan mempercepat stabilisasi robot setelah melakukan koreksi arah.

##### 3.4.6. Tuning $K_i$ (Integral)

$K_i$  ditambahkan secara perlahan jika ditemukan error steady-state (robot berhenti terlalu jauh atau terlalu dekat dari objek), namun nilainya harus dijaga kecil agar tidak menyebabkan osilasi lambat.

##### 3.4.7. Evaluasi dan Revisi

Respons sistem diamati pada berbagai posisi objek dan kondisi pencahayaan. Parameter dianggap optimal ketika robot mampu Menghadapi objek secara stabil, Bergerak menuju objek tanpa overshoot besar, dan Melakukan pen objek secara akurat setelah

berhenti.

### 3.5. Metode Pengujian

Pengujian dilakukan untuk mengevaluasi fungsionalitas dan kinerja sistem robot dalam menjalankan tiga fungsi utama: mengikuti dinding (wall-following), mendeteksi dan melacak objek (object tracking), serta menjepit objek menggunakan gripper. Pengujian difokuskan pada bagaimana sistem kendali fuzzy dan PID bekerja pada kondisi lintasan yang berbeda, serta bagaimana sistem berpindah antar mode secara efektif.

#### 3.5.1 Parameter Pengujian

Untuk menjaga konsistensi dan validitas data, setiap skenario pengujian dilakukan sebanyak 5 kali pengulangan, dengan kondisi awal robot yang sama dan lingkungan yang dikontrol. Pengamatan dilakukan secara langsung dan melalui pencatatan data dari sensor serta hasil visual dari kamera. Parameter yang digunakan dalam pengujian Response Transien Sistem Kendali PID dan Fuzzy, seperti Rise time, overshoot, dan steady-stated error untuk setiap skenario pengujian.

#### 3.5.2 Skenario Pengujian Lintasan Lurus

Robot bergerak di sepanjang lintasan lurus dengan dinding di satu sisi (misalnya sisi kiri), sepanjang 80 cm. Tujuan dari skenario ini adalah mengevaluasi kemampuan fuzzy controller untuk menjaga jarak tetap dari dinding secara konstan.

#### 3.5.3 Skenario Pengujian Lintasan Sudut 90°

Lintasan memiliki satu tikungan tajam 90 derajat (sudut sudut luar) sepanjang 90 cm. Tujuan skenario ini adalah menguji ketanggapan dan kemampuan manuver robot saat terjadi perubahan mendadak arah lintasan.

#### 3.5.4 Skenario Pengujian Lintasan Melengkung

Robot mengikuti dinding pada lintasan melengkung sepanjang 100 cm. Skenario ini menilai kemampuan fuzzy controller beradaptasi dengan perubahan jarak yang bertahap namun berkelanjutan.

#### 3.5.5 Evaluasi Pelacakan dan Penjepitan Objek

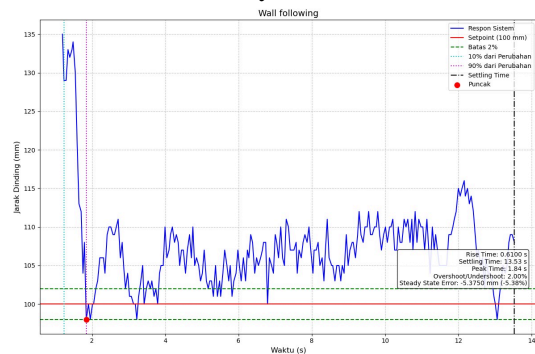
Setelah objek visual terdeteksi, sistem beralih dari wall-following ke mode pelacakan berbasis PID. Pengujian dilanjutkan hingga robot menjangkau objek dan melakukan penjepitan.

Kriteria evaluasi yaitu Keakuratan posisi saat menjepit (objek berada di tengah bingkai kamera), Waktu dari deteksi hingga penjepitan, dan Jumlah penjepitan berhasil dari 5 percobaan

## 4. HASIL DAN PEMBAHASAN

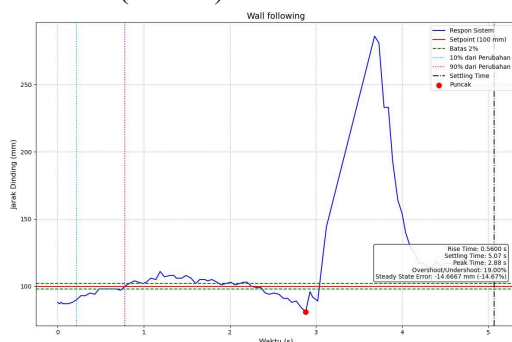
### 4.1. Hasil Pengujian

#### 4.1.1. Kendali Fuzzy



**Gambar 6.** Respon Fuzzy pada tembok lurus

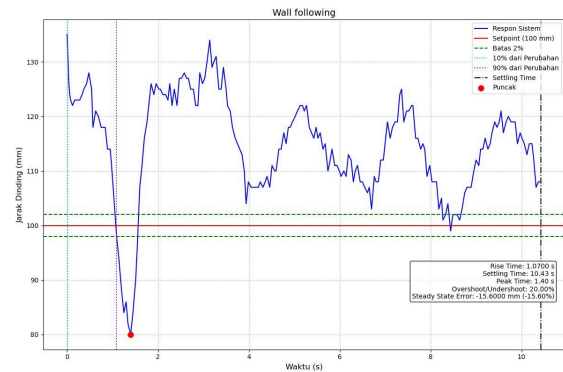
Pada pengujian lintasan lurus, respons sistem ditunjukkan pada Gambar 6. Sistem memberikan respons awal yang cepat dengan rise time 0.61 detik. *Overshoot* yang dihasilkan sangat rendah, yaitu hanya 2%, yang menandakan perilaku kontrol yang stabil dan tidak agresif. Meskipun demikian, sistem memerlukan waktu yang relatif lama untuk mencapai kestabilan penuh, yang ditunjukkan oleh settling time sebesar 13.53 detik. Steady-state error akhir tercatat sebesar -5.375 mm (-5.38%).



**Gambar 7.** Respon Fuzzy pada tembok 90 derajat

Gambar 7 menunjukkan respons sistem saat menghadapi perubahan lintasan yang tajam. Setelah gangguan, sistem merespons dengan cepat, yang terlihat dari rise time 0.56 detik. Namun, perubahan mendadak

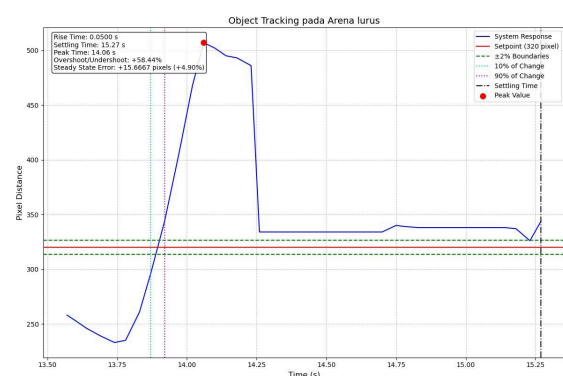
ini menyebabkan overshoot yang signifikan sebesar 19%. Sistem kemudian berhasil mencapai kondisi stabil dalam waktu 5.07 detik.



**Gambar 8.** Respon Fuzzy pada tembok setengah lingkaran

Pada pengujian lintasan melengkung (Gambar 8), sistem menunjukkan kesulitan dalam menjaga kestabilan. Hal ini terlihat dari adanya osilasi yang besar dan berkelanjutan di sepanjang lintasan. Respons sistem memiliki overshoot/undershoot tertinggi, yaitu sebesar 20.00%, dan settling time yang lama, yaitu 10.43 detik. Steady-state error pada skenario ini juga tercatat paling besar, yaitu -15.6 mm (-15.6%).

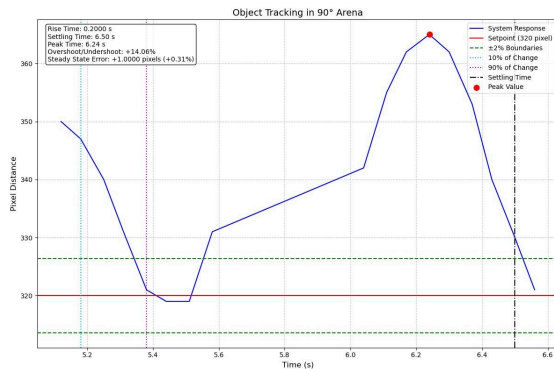
#### 4.1.2. Kendali PID



**Gambar 9.** Respon PID pada tembok lurus

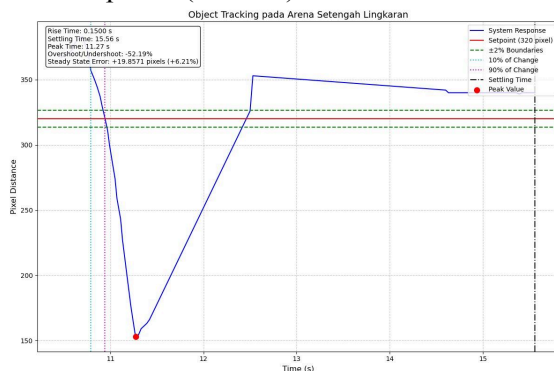
Pada pengujian arena lurus (Gambar 9), kontroler PID menunjukkan respons awal yang sangat cepat dengan rise time hanya 0.05 detik. Namun, kecepatan ini menghasilkan overshoot yang sangat tinggi, mencapai +58.44%, yang menandakan

perilaku kontrol yang sangat agresif. Akibat osilasi besar ini, sistem memerlukan waktu yang sangat lama untuk stabil, dengan settling time tercatat sebesar 15.27 detik.



**Gambar 10.** Respon PID pada tembok 90 derajat

Pengujian arena belokan 90 derajat (Gambar 10) menunjukkan kinerja terbaik dari kontroler PID. Rise time tercatat sebesar 0.2000 detik dengan overshoot yang jauh lebih terkontrol, yaitu +14.06%. Sistem berhasil mencapai kestabilan dengan cepat, ditunjukkan oleh settling time 6.50 detik, dan memiliki steady-state error yang sangat kecil, hanya +1.0000 piksel (+0.31%).



**Gambar 11.** Respon PID pada tembok setengah lingkaran

Pada arena setengah lingkaran (Gambar 11), sistem kembali menunjukkan kesulitan dalam menjaga kestabilan. Grafik menunjukkan adanya undershoot yang sangat signifikan sebesar -52.19%. Settling time pada skenario ini tercatat sangat lama, yaitu 15.56 detik, yang mengindikasikan kesulitan sistem untuk menyesuaikan diri dengan jalur yang melengkung secara halus.

#### 4.1.3. Tabel Respon Transien

**Tabel 1.** Respon Transien PID

Parameter Response	Arena Lurus	Arena 90°	Arena Setengah Lingkaran
Rise Time	0.61s	0.56s	1.07s
Settling Time	13.53s	5.07s	10.43s
Peak Time	1.84s	2.88s	1.4s
Overshoot/ Undershoot	2%	19%	20%
Steady State Error	-5.375 mm	-14.67 mm	-15.6mm

Tabel ini mengukur performa kontroler PID saat robot bertugas melacak objek menggunakan kamera. Tujuannya adalah menjaga objek tetap di tengah pandangan kamera.

**Lintasan Lurus:** Responnya sangat cepat (*rise time* 0.05 detik) namun sangat tidak stabil. Hal ini terlihat dari *overshoot* yang masif sebesar +58.44%, yang berarti robot bergoyang hebat sebelum stabil.

**Lintasan 90°:** Ini adalah performa **terbaik** dari kontroler PID. Robot menunjukkan keseimbangan yang baik antara kecepatan dan stabilitas, dengan *overshoot* yang jauh lebih terkontrol (+14.06%) dan *error* akhir yang sangat kecil (+1.0000 piksel).

**Lintasan Setengah Lingkaran:** Kontroler PID juga sangat kesulitan di sini. Robot merespons dengan lambat (*undershoot* -52.19%) dan butuh waktu paling lama untuk stabil (*settling time* 15.56 detik)

**Tabel 2.** Respon Transien Fuzzy

Parameter Response	Arena Lurus	Arena 90°	Arena Setengah Lingkaran
Rise Time	0.05s	0.2s	0.15s
Settling Time	15.27s	6.5s	15.56s
Peak Time	14.06s	6.24s	11.27s
Overshoot/ Undershoot	58.44%	14%	-52.19%



<b>Steady State Error</b>	15.667 Pixels	1 Pixels	19.857 Pixels
---------------------------	---------------	----------	---------------

Tabel ini mengukur performa Logika Fuzzy saat robot bertugas mengikuti dinding. Tujuannya adalah menjaga jarak yang stabil dari dinding.

**Lintasan Lurus:** Ini adalah performa terbaik dari kontroler Fuzzy. Responnya stabil dengan *overshoot* hanya 2%.

**Lintasan 90°:** Robot berbelok dengan cepat (*settling time* 5.07 detik), namun cukup agresif karena *overshoot*-nya tinggi, yaitu 19%.

**Lintasan Setengah Lingkaran:** Ini adalah performa terburuk. Kontroler kesulitan menjaga kestabilan di tikungan landai, yang ditunjukkan oleh *overshoot* tertinggi sebesar 20% dan *error* akhir paling besar (-15.6mm).

#### 4.2. Pembahasan

##### ANALISIS

Untuk *Wall Following*, kontroler Fuzzy menunjukkan performa paling stabil di lintasan lurus. Hal ini dibuktikan dengan nilai *overshoot* yang sangat rendah, hanya 2.00%, yang menandakan sistem tidak agresif dan mampu menjaga jarak dengan mulus. Meskipun waktu respons awalnya cepat (*rise time* 0.61 detik), sistem memerlukan waktu yang cukup lama untuk mencapai kestabilan penuh (*settling time* 13.53 detik) dengan *steady-state error* akhir sebesar -5.375 mm. Performa menurun drastis saat menghadapi tikungan, di mana *overshoot* melonjak menjadi 19% pada belokan 90 derajat dan 20.00% pada lintasan melengkung. *Steady-state error* juga memburuk secara signifikan pada lintasan melengkung, mencapai -15.6 mm.

*Object tracking*, kontroler PID. Pada lintasan lurus, kontroler ini memiliki waktu respons yang sangat cepat (*rise time* 0.05 detik), namun harus dibayar dengan *overshoot* yang sangat tinggi mencapai +58.44% dan *settling time* yang lama (15.27 detik). Hal ini mengindikasikan tuning parameter yang terlalu agresif untuk skenario pelacakan yang stabil. Performa terbaik PID justru terlihat pada skenario belokan 90 derajat, di mana *overshoot* jauh lebih terkontrol sebesar +14.06% dan *settling time* menjadi yang tercepat (6.50

detik). Selain itu, *steady-state error* pada skenario ini sangat kecil, hanya +1.0000 piksel, membuktikan efektivitasnya dalam melakukan koreksi yang presisi setelah berbelok tajam. Namun, seperti kontroler Fuzzy, PID juga gagal beradaptasi pada lintasan melengkung, yang ditunjukkan dengan adanya *undershoot* signifikan sebesar -52.19% dan *settling time* terlama, yaitu 15.56 detik.

##### KESIMPULAN

Kesimpulan dari proyek ini menunjukkan bahwa arsitektur sistem kontrol terdistribusi dengan pembagian peran antara Raspberry Pi dan ESP32 berhasil meningkatkan efisiensi dan memungkinkan pemrosesan gambar serta kendali. Pengontrol fuzzy menunjukkan kinerja adaptif dalam berbagai skenario lintasan, namun masih memiliki kelemahan dalam hal presisi jangka panjang dan sensitivitas terhadap perubahan lingkungan mendadak. Sistem deteksi objek berbasis HSV mampu mengenali warna target secara andal, sedangkan pengendali PID berhasil menghasilkan gerakan pelacakan objek yang terarah, meskipun memerlukan tuning parameter yang cermat. Pengujian di lapangan menunjukkan performa terbaik pada lintasan lurus, tetapi sistem masih menghadapi kesulitan signifikan pada tikungan tajam dan lintasan melengkung. Secara umum, sistem memiliki keunggulan dari sisi modularitas, fungsionalitas multi-misi, dan integrasi sensor, namun masih perlu peningkatan dalam hal stabilitas kontrol, akurasi *steady-state*, dan kemampuan adaptasi terhadap lingkungan dinamis. Pengembangan selanjutnya diarahkan pada otomatisasi tuning kendali menggunakan algoritma optimasi, penambahan sensor seperti IMU/LIDAR, pengembangan sistem SLAM untuk navigasi otonom, serta peningkatan sistem komunikasi antar modul untuk mendukung skala dan fleksibilitas lebih tinggi.

Kontribusi utama dari penelitian ini terletak pada keberhasilan integrasi kedua metode kontrol ini dalam sebuah arsitektur terdistribusi yang efisien. Sebagian besar penelitian sebelumnya, seperti yang dirujuk, cenderung berfokus pada salah satu metode untuk satu tugas spesifik. Penelitian ini menjawab kesenjangan tersebut dengan mendemonstrasikan bagaimana Raspberry Pi

dapat secara efektif menangani pemrosesan yang intensif untuk *object tracking*, sementara ESP32 secara *real-time* mengelola logika Fuzzy dan kontrol motor untuk navigasi *wall-following*. Dengan demikian, keunggulan sistem yang diusulkan adalah fungsionalitas multi-misi yang berjalan secara bersamaan pada satu platform robot tunggal, yang dicapai melalui pembagian tugas komputasi yang jelas.

Meskipun sistem yang diusulkan berhasil menunjukkan fungsionalitasnya, penelitian ini memiliki beberapa keterbatasan yang membuka peluang untuk pengembangan di masa depan. Keterbatasan utama terletak pada proses *tuning* parameter kontroler (Fuzzy dan PID) yang masih dilakukan secara manual melalui metode *trial-and-error*. Proses ini tidak hanya memakan waktu tetapi juga belum tentu menghasilkan parameter yang paling optimal. Hal ini kemungkinan menjadi penyebab munculnya respons yang kurang ideal, seperti *overshoot* yang tinggi pada kontroler PID dan osilasi pada lintasan melengkung. Selain itu, pengujian sistem hanya dilakukan pada arena lintasan yang terkontrol dan tidak di lingkungan yang lebih kompleks atau dinamis yang mungkin memiliki rintangan tak terduga atau perubahan kondisi pencahayaan.

Untuk mengatasi keterbatasan tersebut, beberapa pengembangan dapat dilakukan. Pertama, untuk menyempurnakan performa kontroler, dapat diimplementasikan sistem *tuning* otomatis, dapat secara sistematis menemukan parameter kontroler yang paling efektif. Kedua, untuk meningkatkan persepsi dan stabilitas navigasi, penelitian selanjutnya dapat berfokus pada integrasi sensor tambahan. Penambahan sensor seperti *Inertial Measurement Unit* (IMU) akan memberikan data orientasi yang akurat untuk menstabilkan pergerakan robot, sementara sensor LiDAR dapat menawarkan kemampuan pemetaan jarak yang lebih superior dibandingkan sensor ultrasonik. Pada akhirnya, dengan data sensor yang lebih kaya.

## DAFTAR PUSTAKA

- [1] D. I. Shema, "Implementasi Metode Logika Fuzzy Sugeno Pada Prototipe Robot Pemadam Api Dengan Kemampuan Navigasi," Jurnal Teknologi Elektro, vol. 13, no. 1, p. 55, Feb. 2022, doi: <https://doi.org/10.22441/jte.2022.v13i1.010>.
- [2] M. N. F. Astoro, T. Winarno, and A. Komarudin, "Implementasi Kontrol PID Positioning Robot Pada Titik Pengambilan Objek Shagai Menggunakan Sensor Kamera," Jurnal Elektronika dan Otomasi Industri, vol. 7, no. 2, pp. 57–57, Apr. 2020, doi: <https://doi.org/10.33795/elkolind.v7i2.195>.
- [3] A. S. Priambodo, F. Arifin, A. Nasuha, and A. Winursito, "Face Tracking for Flying Robot Quadcopter based on Haar Cascade Classifier and PID Controller," Journal of Physics: Conference Series, vol. 2111, no. 1, p. 012046, Nov. 2021, doi: <https://doi.org/10.1088/1742-6596/2111/1/012046>.
- [4] N. Aisyah, A. Rahman, R. Wirawan, Bilgah, S. Rachmawati, A. Medikano, and A. Sebayang, "Pengujian Robot Otomatis Pendeteksi Rintangan Berbasis Mikrokontroler," Jurnal Esensi Infokom Jurnal Esensi Sistem Informasi dan Sistem Komputer, vol. 5, no. 2, pp. 1–6, Feb. 2022, doi: <https://doi.org/10.55886/infokom.v5i2.264>.
- [5] F. T. Hartadi, B. A. Wicaksana, H. Saputro, and A. S. Priambodo, "SISTEM KENDALI FUZZY UNTUK ROBOT MOBILE: STUDI KASUS PELACAKAN OBJEK BERGERAK MENGGUNAKAN SIMULASI WEBOTS," Jurnal Informatika dan Teknik Elektro Terapan, vol. 12, no. 3, Aug. 2024, doi: <https://doi.org/10.23960/jitet.v12i3.4608>.
- [6] C.-H. Chen, S.-Y. Jeng, and C.-J. Lin, "Mobile Robot Wall-Following Control Using Fuzzy Logic Controller with Improved Differential Search and Reinforcement Learning," Mathematics, vol. 8, no. 8, p. 1254, Jul. 2020, doi: <https://doi.org/10.3390/math8081254>.
- [7] S. R. Divanny, Z. Nugraha, M. I. Ghaffar, Muhammad Rizqi Dwi A, S. D. S. Wardana, and A. S. Priambodo, "PENERAPAN OPENCV DAN FUZZY LOGIC CONTROLLER UNTUK LINE FOLLOWER BERBASIS KAMERA PADA SIMULASI ROBOT E-PUCK DI WEBOTS," Jurnal Informatika dan Teknik Elektro Terapan, vol. 12, no. 3, Aug. 2024, doi: <https://doi.org/10.23960/jitet.v12i3.4718>.
- [8] Anugrah Akmal Arista A, K. A. Paradifta, Z. Nugraha, and A. S. Priambodo, "Implementasi Fuzzy Logic Pada Kendali

- Robot E-Puck Wall Following,” Journal of Electrical, Electronic, Control, and Automotive Engineering, vol. 6, no. 1, pp. 16–21, May 2024, doi: <https://doi.org/10.32486/jeecae.v9i1.708>.
- [9] T. H. Prasetyo, I. Siradjuddin, and Sungkono, “Sistem Kendali Wall Following Pada Mobile Robot Dengan Penggerak Mekanum Menggunakan Metode Fuzzy,” *Jurnal Elektronika Otomasi Industri*, vol. 8, no. 1, pp. 214–220, Sep. 2021, doi: <https://doi.org/10.33795/elkolind.v8i3/268>.
- [10] S. Sukardi and Satria Gunawan Zain, “Pengembangan Algoritme Gerak Robot Hexapod Menggunakan Logika Fuzzy,” *Journal of Embedded System Security and Intelligent Systems*, vol. 2, no. 1, pp. 41–48, May 2021.
- [11] A. Priswanto, S. Kadiman, and J. Prasajo, “SISTEM KENDALI HYBRID PID-FLC PADA STUDI KASUS ROBOT PENJEJAK DINDING BERKAKI,” *Jurnal Mahasiswa Teknik Elektro*, vol. 1, no. 1, pp. 91–100, Sep. 2020.
- [12] A. Štefek, V. T. Pham, V. Krivanek, and K. L. Pham, “Optimization of Fuzzy Logic Controller Used for a Differential Drive Wheeled Mobile Robot,” *Applied Sciences*, vol. 11, no. 13, p. 6023, Jan. 2021, doi: <https://doi.org/10.3390/app11136023>.
- [13] H. Sihombing, E. Setiawan, dan S. R. Akbar, “Sistem Kendali Robot Beroda Wall-Follower pada Tembok Lurus dan Berbelok menggunakan Metode PID,” *J-PTIIK*, vol. 6, no. 11, hlm. 5129–5138, Des 2022.
- [14] Moh. Nur Ilhami Fajri Santoso, “Aplikasi Sensor Ultrasonik HC-SR04 Pada Robot Wall Follower Untuk Menentukan Kecepatan Motor,” *SinarFe7*, vol. 4, no. 1, pp. 148–151, Dec. 2021.
- [15] D. Chu, H. Li, C. Zhao, and T. Zhou, “Trajectory Tracking of Autonomous Vehicle Based on Model Predictive Control With PID Feedback,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2022, doi: <https://doi.org/10.1109/tits.2022.3150365>.