

SISTEM KENDALI *FUZZY* UNTUK ROBOT *MOBILE*: STUDI KASUS PELACAKAN OBJEK BERGERAK MENGGUNAKAN SIMULASI *WEBOTS*

Fredy Tri Hartadi¹, Bagus Arya Wicaksana², Hendro Saputro³, Ardy Seto Priambodo⁴

^{1,2,3,4} Universitas Negeri Yogyakarta; Jl. Colombo No.1, Karang Malang, Caturtunggal, Kec. Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55281; (0274) 586168

Received: 22 Juni 2024

Accepted: 31 Juli 2024

Published: 7 Agustus 2024

Keywords:

Fuzzy Logic;
Deteksi Objek;
Kamera;
Vision;
Webots;

Correspondent Email:

fredytri.2022@student.uny.ac.id

Abstrak. Dalam era perkembangan teknologi robotika yang semakin maju, implementasi sistem kendali menjadi sangat penting dalam berbagai aplikasi, termasuk logistik, pengawasan, dan layanan kesehatan. Penelitian ini mengembangkan dan menguji sistem pengendalian robot menggunakan *fuzzy logic* untuk mendeteksi dan mengikuti objek bergerak berwarna biru dalam lingkungan dinamis. Penggunaan *fuzzy logic* dipilih karena kelebihanannya dalam menangani ketidakpastian dan dinamika lingkungan yang kompleks dibandingkan dengan metode pengendalian tradisional seperti PID. Robot *Khepera IV* digunakan sebagai *platform*, dan simulasi dilakukan menggunakan *software Webots*. Sistem ini mengintegrasikan sensor kamera untuk pendeteksian objek dan sensor inframerah untuk penghindaran rintangan. Hasil percobaan menunjukkan bahwa sistem kendali *fuzzy logic* yang dikembangkan mampu mengikuti objek dengan baik, menunjukkan waktu *rise time* yang cepat bernilai 0,384 detik, dan tidak mengalami *overshoot*. Penelitian ini membuktikan bahwa *fuzzy logic* efektif dalam pengendalian robot untuk pelacakan objek berwarna, dengan hasil terbaik menunjukkan rata-rata error yang rendah yaitu -1,687 dan respon kendali yang responsif terhadap perubahan posisi objek.

Abstract. In the era of advanced robotics technology development, the implementation of control systems has become very important in various applications, including logistics, surveillance, and healthcare. This research develops and tests a robot control system using *fuzzy logic* to detect and follow a blue moving object in a dynamic environment. The use of *fuzzy logic* was chosen due to its advantages in handling uncertainty and complex environmental dynamics compared to traditional control methods such as PID. The *Khepera IV* robot was used as the platform, and simulations were performed using *Webots* software. The system integrates camera sensors for object detection and infrared sensors for obstacle avoidance. Experimental results show that the developed *fuzzy logic* control system is able to follow objects well, exhibits a fast rise time of 0.384 seconds, and does not experience overshoot. This research proves that *fuzzy logic* is effective in controlling robots for tracking colored objects, with the best results showing a low average error of -1.687 seconds and a responsive control response to changes in object position.

1. PENDAHULUAN

1.1. Latar Belakang

Dalam beberapa dekade terakhir, perkembangan teknologi robotika telah mengalami kemajuan pesat, terutama dalam aplikasi robot otonom. Robot otonom memiliki kemampuan untuk menjalankan tugas-tugas kompleks tanpa campur tangan manusia secara langsung. Salah satu aspek penting dari robot otonom adalah kemampuan untuk mendeteksi dan mengikuti objek tertentu, yang sangat relevan dalam berbagai aplikasi seperti logistik, pengawasan, dan layanan kesehatan. Pengendalian robot untuk pelacakan objek biasanya dilakukan dengan menggunakan pengendalian proporsional-integral-derivatif (PID) yang memiliki beberapa keterbatasan dalam menangani ketidakpastian dan dinamika lingkungan yang kompleks[1].

Beberapa penelitian telah mengkaji penggunaan metode pengendalian *fuzzy* untuk meningkatkan performa pelacakan objek oleh robot. *Fuzzy logic* menawarkan pendekatan yang lebih fleksibel dibandingkan metode PID tradisional, terutama dalam lingkungan yang dinamis dan tidak terstruktur[2].

Teori *fuzzy logic* pertama kali diperkenalkan oleh Lotfi Zadeh pada tahun 1965, yang memungkinkan pemodelan sistem yang kompleks dan tidak pasti dengan menggunakan aturan linguistik yang menyerupai cara berpikir manusia[3][4]. Dalam konteks robotika, *fuzzy logic* dapat diterapkan untuk mengendalikan kecepatan roda berdasarkan kesalahan posisi antara robot dan objek yang diikuti. Pengolahan citra digunakan untuk mendeteksi objek berdasarkan warna dengan menggunakan transformasi warna HSV, yang lebih *robust* terhadap variasi pencahayaan dibandingkan ruang warna RGB[5].

Meskipun penelitian sebelumnya menunjukkan bahwa penggunaan *fuzzy logic* dalam pengendalian robot memberikan hasil yang lebih baik dibandingkan metode tradisional, masih terdapat beberapa kekurangan yang perlu diatasi. Banyak penelitian yang belum mengeksplorasi

penggunaan *fuzzy logic* untuk pelacakan objek bergerak dengan warna selain merah[6][7]. Sementara itu dalam keadaan simulasi, memungkinkan untuk bisa didapatkan kontroler *fuzzy* paling optimal. Oleh karena itu, penelitian ini berupaya memberikan hasil dengan mengembangkan sistem pengendalian robot menggunakan *fuzzy logic* untuk mengikuti objek tujuan sesuai target objeknya yaitu berwarna biru dan mengintegrasikan sensor inframerah untuk penghindaran hambatan ketika tidak terdapat objek yang dideteksi oleh kamera robot.

1.2. Tujuan

Penelitian ini bertujuan untuk mengembangkan dan menguji sistem pengendalian robot menggunakan *fuzzy logic* untuk mendeteksi dan mengikuti objek bergerak berwarna biru dalam lingkungan dinamis [8]. Dipilih warna biru karena warna pada objek terkecil adalah biru, digunakan untuk mengukur seberapa sensitif kamera mendeteksi warna pada objek. Persoalan utama dalam penelitian adalah bagaimana efektivitas penggunaan *fuzzy logic* dalam mengendalikan robot untuk mengikuti objek berwarna biru antara objek yang diam dan bergerak.

Penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam bidang robotika, khususnya dalam pengembangan sistem robot otonom yang lebih adaptif dan efektif dalam mengikuti objek bergerak di lingkungan yang kompleks.

2. TINJAUAN PUSTAKA

2.1. Robot Mobile

Robot mobile adalah robot yang memiliki kemampuan untuk bergerak secara bebas di lingkungan fisik. Tidak seperti robot stasioner yang tetap di satu tempat, robot *mobile* dapat berpindah tempat untuk melaksanakan tugas-tugasnya.

Robot *mobile* dilengkapi dengan mekanisme yang memungkinkan mereka bergerak, seperti roda, kaki, atau trek. Robot *mobile* biasanya dilengkapi dengan berbagai sensor untuk navigasi dan deteksi lingkungan, seperti sensor jarak, kamera,

lidar, atau sensor inframerah. Terdapat sistem navigasi pada robot untuk menentukan dan mengikuti jalur, menghindari rintangan, dan mencapai tujuan tertentu. Robot *mobile* memiliki sistem kontrol yang mengatur gerakannya berdasarkan input dari sensor dan instruksi yang telah diprogram. Kontrol ini bisa bersifat otonom atau teleoperasi (dikendalikan dari jarak jauh)[8].

2.2. Robot Vision

Robot *vision*, atau visi robot, adalah teknologi yang memungkinkan robot untuk memahami dan menafsirkan informasi visual dari lingkungannya. Teknologi ini menggabungkan berbagai disiplin ilmu, termasuk pemrosesan citra, analisis video, kecerdasan buatan, dan pembelajaran mesin, untuk memberi robot kemampuan untuk "melihat" dan membuat keputusan berdasarkan apa yang dilihatnya.

Penggunaan *vision* pada robot memiliki proses pengubahan data mentah dari kamera menjadi informasi yang berguna (*Image Processing*). Macam macam pemrosesan gambar seperti penghapusan noise, peningkatan kontras, segmentasi gambar. *Vision* robot melibatkan teknik untuk mendeteksi, mengenali, dan melacak objek dalam gambar mencakup deteksi tepi, pengenalan pola, dan pengenalan objek[9].

2.3. Dynamic Object Following

Dynamic object following merujuk pada kemampuan sebuah sistem untuk mengidentifikasi, melacak, dan mengikuti objek yang bergerak secara dinamis dalam lingkungannya melibatkan penggunaan sensor untuk mendeteksi objek dan algoritma kontrol untuk memandu robot atau perangkat agar tetap dalam jarak tertentu dari objek tersebut [1][10].

2.4. Logika Fuzzy dan Kontrol Fuzzy

Logika *fuzzy* diperkenalkan pada tahun 1965 oleh Lotfi A. Zadeh dalam makalah penelitiannya "Fuzzy Sets". Lotfi A. Zadeh dianggap sebagai bapak logika *fuzzy*. Logika *fuzzy* menyerupai metode pengambilan keputusan manusia. Pada logika *boolean* hanya terdapat dua kondisi 0 atau 1, dengan

logika *fuzzy* suatu nilai dapat dikatakan diantara nilai 0 dan 1[3].

Kontrol *fuzzy* merupakan metode kontrol lanjut yang mengimitasi cara berpikir manusia. Penggunaan kontrol ini dapat memecahkan permasalahan pada kontrol yang disebabkan oleh permodelan dan gangguan yang bersifat acak dan diluar dari perhitungan matematika. Kontroler *fuzzy* dapat menahan gangguan secara efektif, memiliki kecepatan respons yang optimal, dan sanggup menahan perubahan sistem secara intensif[11].

2.5. Khepera IV

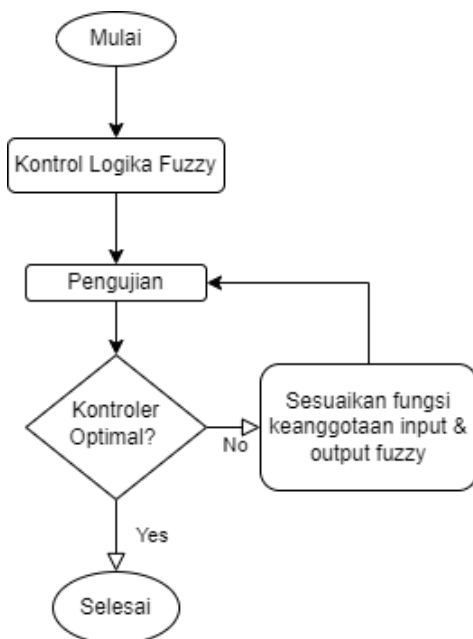
Pada penelitian ini robot yang digunakan adalah *Khepera IV*. Ini adalah sebuah *mobile* robot berukuran mini yang digunakan untuk pengujian algoritma seperti menghindari rintangan, mengikuti dinding, mencari target, navigasi, sepak bola robot, dan lain sebagainya. *Khepera IV* memiliki pustaka aplikasi *on-board* yang digunakan untuk mengontrol robot, mengawasi proses berjalannya percobaan, dan mengeksekusi perintah pengguna secara efisien[12].

2.6. Webots

Webots merupakan software dengan lingkungan virtual yang digunakan untuk mensimulasikan robot dengan perintah yang sudah diinputkan pada kontroler dalam bentuk kode program. Spesifikasi dari robot yang digunakan pada simulasi terdapat pada dokumentasi situs *cyberbotic*[13].

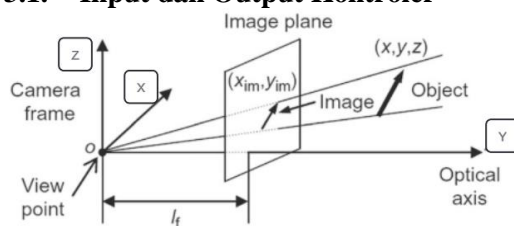
3. METODE PENELITIAN

Dalam penelitian ini, terdapat beberapa tahap untuk menyelesaikannya. Adapun tahap pengerjaan dalam penelitian dapat dilihat pada Gambar 1.



Gambar 1. Kerangka Metode Penelitian

3.1. Input dan Output Kontroler



Gambar 2. Ilustrasi Vision Robot

Ilustrasi objek dalam kamera dapat dilihat pada Gambar 2, dimana target objek harus ada pada titik tengah horizontal *vision* kamera. Untuk parameternya adalah:

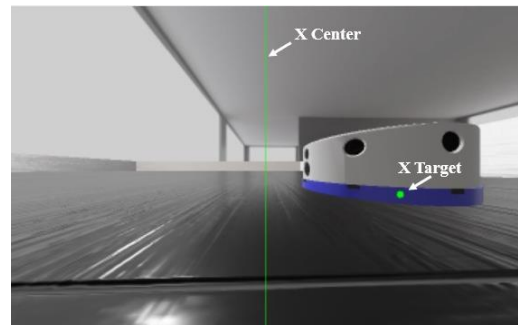
e : Error
 Δe : Perbedaan *error*
 $e(t)$: *Error* saat ini
 $e(t - l)$: *Error* sebelumnya
 X : Nilai lebar gambar (piksel)
 X_{target} : Nilai aktual target (piksel)
 X_{center} : Nilai tengah gambar (piksel)

Berdasarkan parameter yang telah disebutkan dapat ditulis persamaan penghitungan *error*:

$$e = X_{target} - X_{center} \quad (1)$$

Dengan nilai X_{center} dapat diperoleh dari:

$$X_{center} = X/2 \quad (2)$$



Gambar 3. Ilustrasi dalam Simulasi

Sehingga dapat ditarik kesimpulan seperti tabel dibawah ini.

Tabel

1.

Nilai e	Arah Gerak Robot
Negatif	Belok kanan
Positif	Belok kiri
0	Lurus

Keputusan Gerak Robot

Sedangkan untuk mendeteksi adanya perubahan *error* atau pergerakan arah objek pada kamera adalah seperti berikut:

$$\Delta e = e(t) - e(t - l) \quad (3)$$

Dari hasil perhitungan dapat ditarik kesimpulan dalam tabel dibawah ini.

Nilai Δe	Pergerakan Objek
Negatif	Bergerak ke kanan
Positif	Bergerak ke kiri
0	Tidak ada pergerakan

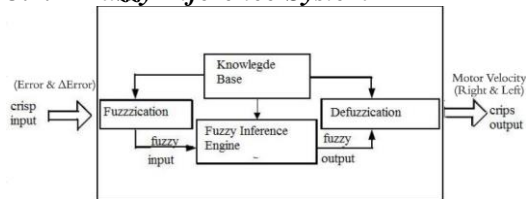
Tabel 2. Deteksi Arah Gerak Objek

Untuk output pada kontrol fuzzy menggunakan kecepatan motor kanan dan kiri sehingga perbedaan kecepatan motor tersebut memberikan pergerakan robot berbelok ke kanan dan kiri menyesuaikan target objek pada titik tengah *vision* kamera. Jika kecepatan motor kiri V_l dan motor kanan V_r , maka:

- $V_l > V_r$ untuk robot berbelok ke kanan,
- $V_l < V_r$ untuk robot berbelok ke kiri,
- $V_l = V_r$ untuk robot bergerak lurus.

Seberapa tajam robot berbelok ditentukan oleh besarnya perbedaan antara V_l dan V_r yang akan ditentukan dalam *rule base* kontroler *fuzzy*.

3.2. Fuzzy Inference System



Gambar 4. Fuzzy Inference System

Fuzzy Inference System adalah sistem untuk menghitung *fuzzy logic* sebagai kontroler pada penelitian ini. Terdiri dari Fuzzifikasi, Mesin Inferensi *Fuzzy*, dan Defuzzifikasi. Dimana ketiga hal tersebut mengkonversi dari *input* bilangan crisp yaitu nilai *error* dan $\Delta error$ menjadi nilai keanggotaan *fuzzy* yang akan dihitung dan pemilihan keputusan berdasarkan basis aturan, sehingga didapat *output* dalam bentuk bilangan *fuzzy*. Langkah berikutnya dilakukan pengembalian nilai *output fuzzy* menjadi bilangan crisp yang akan diberikan ke aktuator robot yaitu kecepatan motor kanan (V_r) dan kiri (V_l), langkah ini disebut defuzzifikasi.

3.2.1. Fuzzifikasi

Fuzzifikasi adalah proses mengubah input data yang bersifat tegas (*crisp*) menjadi nilai derajat keanggotaan dalam satu atau lebih himpunan fuzzy [14]. Pada fuzzifikasi ini terdapat dua hal yang harus diperhatikan yaitu nilai masukan (*Input*) dan nilai keluaran (*output*). Proses ini adalah langkah pertama dalam sistem kontrol *fuzzy*. Dimana nilai input *crisp* variabel input *error* dan $\Delta error$ dikonversi menjadi nilai keanggotaan untuk *fuzzy* input.

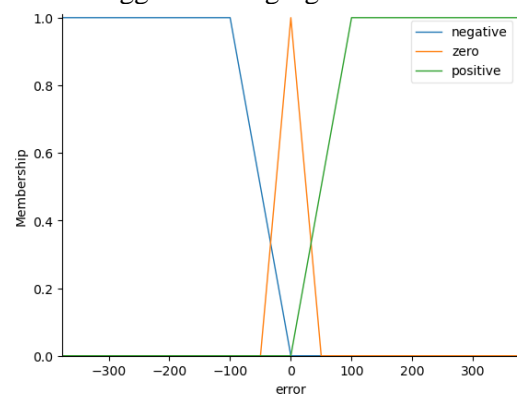
Kemudian rentang nilai linguistik *error* dapat dilihat seperti Tabel 3 dibawah.

Tabel 3. Nilai linguistik *error*

Rentang <i>Error</i>	Nilai Linguistik
----------------------	------------------

-376 s/d 0	<i>Negative</i>
-50 s/d 50	<i>Zero</i>
0 s/d 376	<i>Positive</i>

Berdasarkan Tabel 3 fungsi keanggotaan dari nilai linguistik input *error* dapat dilihat pada Gambar 5, dimana untuk nilai linguistik *Negative* dan *Positive* menggunakan fungsi keanggotaan trapesium, sedangkan untuk nilai linguistik *Zero* menggunakan segitiga.



Gambar 5. Fungsi keanggotaan *error*

Untuk proses fuzzifikasi pada variabel input *error* seperti nilai linguistik *negative*, *zero*, dan *positive* didapatkan dengan persamaan fungsi keanggotaan sebagai berikut.

$$\mu_{negative}(x) \begin{cases} 1; x < -100 \\ \frac{0-x}{0-(-100)}; -100 \leq x \leq 0 \\ 0; x > 0 \end{cases} \quad (4)$$

$$\mu_{zero}(x) \begin{cases} 0; x < -50 \text{ dan } x > 50 \\ \frac{x-(-50)}{0-(-50)}; -50 \leq x < 0 \\ \frac{50-x}{50-0}; 0 < x \leq 50 \\ 1; x = 0 \end{cases} \quad (5)$$

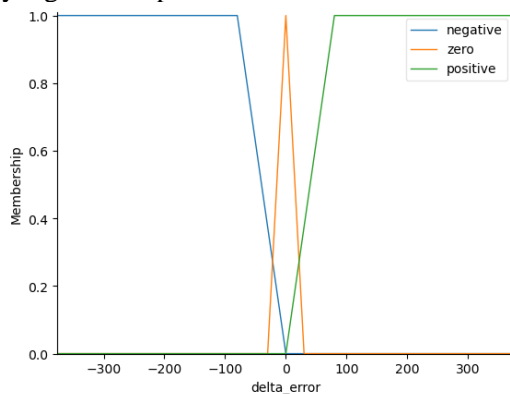
$$\mu_{positive}(x) \begin{cases} 0; x < 0 \\ \frac{100-x}{100-0}; 0 \leq x \leq 100 \\ 1; x > 100 \end{cases} \quad (6)$$

Kemudian, untuk rentang nilai linguistik pada variabel input $\Delta error$ terlihat pada Tabel 4. Untuk nilai linguistiknya yaitu *negative*, *zero*, dan *positive*.

Tabel 4. Nilai linguistik $\Delta error$

Rentang $\Delta error$	Nilai linguistik
-376 s/d 0	<i>Negative</i>
-30 s/d 30	<i>Zero</i>
0 s/d 376	<i>Positive</i>

Dari Tabel 4 didapatkan fungsi keanggotaan $\Delta error$, untuk nilai linguistik *Negative* dan *Positive* menggunakan fungsi keanggotaan trapesium, sedangkan untuk nilai linguistik *Zero* menggunakan segitiga yang terlihat pada Gambar 6.



Gambar 6. Fungsi keanggotaan $\Delta error$

Pada proses fuzzifikasi variabel input $\Delta error$ seperti nilai linguistik *negative*, *zero*, dan *positive* didapatkan dengan persamaan fungsi keanggotaan sebagai berikut.

$$\mu_{negative}(y) \begin{cases} 1; y < -80 \\ \frac{0-y}{0-(-80)}; -80 \leq y \leq 0 \\ 0; y > 0 \end{cases} \quad (7)$$

$$\mu_{zero}(y) \begin{cases} 0; y < -30 \text{ dan } y > 30 \\ \frac{y-(-30)}{0-(-30)}; -30 \leq y < 0 \\ \frac{30-y}{30-0}; 0 < y \leq 30 \\ 1; y = 0 \end{cases} \quad (8)$$

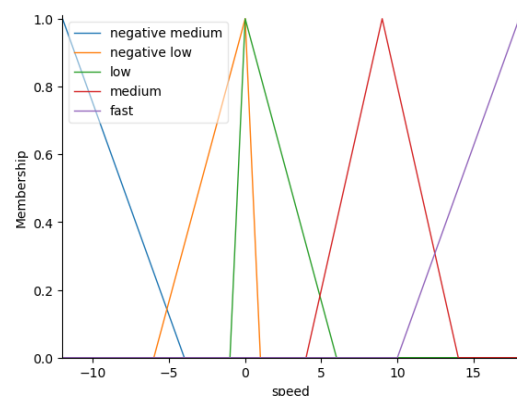
$$\mu_{positive}(y) \begin{cases} 0; y < 0 \\ \frac{80-y}{80-0}; 0 \leq y \leq 80 \\ 1; y > 80 \end{cases} \quad (9)$$

Selanjutnya, variabel output kecepatan motor terdapat 5 kategori nilai linguistik untuk mengatur kecepatan motor terlihat pada Tabel 5.

Tabel 5. Nilai Linguistik Output

Rentang Kecepatan Motor	Nilai Linguistik
-12 s/d -4	<i>Negative Medium</i>
-6 s/d 0	<i>Negative Low</i>
0 s/d 6	<i>Low</i>
4 s/d 14	<i>Medium</i>
10 s/d 18	<i>Fast</i>

Pada Gambar 7 nilai linguistik output kecepatan motor menggunakan fungsi keanggotaan segitiga pada masing-masing kategori, untuk fungsi keanggotaan dan nilai linguistik kecepatan motor kanan dan kecepatan motor kiri sama.



Gambar 7. Fungsi keanggotaan kecepatan motor

Untuk perhitungan fuzzifikasi pada variabel output menggunakan lima fungsi keanggotaan, dengan persamaan sebagai berikut.

$$\mu_{n\ medium}(z) \begin{cases} 1; z = -12 \\ \frac{(-4)-z}{(-4)-(-12)}; -12 < z \leq -4 \\ 0; z > -4 \end{cases} \quad (10)$$

$$\mu_{negative\ low}(z) \begin{cases} 1; z = 0 \\ \frac{z-(-6)}{0-(-6)}; -6 \leq z < 0 \\ 0; z > 0 \text{ dan } z < -6 \end{cases} \quad (11)$$

$$\mu_{low}(z) \begin{cases} 1; z = 0 \\ \frac{6-z}{6-0}; 0 < z \leq 6 \\ 0; z < 0 \text{ dan } z > 6 \end{cases} \quad (12)$$

$$\mu_{medium}(z) \begin{cases} 1; z = 9 \\ \frac{z-4}{9-4}; 4 \leq z < 9 \\ \frac{14-z}{14-9}; 9 < z \leq 14 \\ 0; z > 14 \text{ dan } z < 4 \end{cases} \quad (13)$$

$$\mu_{fast}(z) \begin{cases} 1; z = 18 \\ \frac{z-10}{18-10}; 10 \leq z < 18 \\ 0; z < 10 \end{cases} \quad (14)$$

3.2.2. Mesin Inferensi Fuzzy

Setelah mengetahui fungsi keanggotaan, langkah berikutnya adalah menentukan aturan dasar yang akan digunakan untuk menghasilkan output atau keputusan yang diambil berdasarkan nilai input yang didapat, sesuai dengan aturan yang terdapat pada Tabel 6.

Tabel 6. Basis Aturan

Input		Output	
Error	Δ Error	Left speed	Right speed
Negative	Negative	Fast	Negative medium
Negative	Zero	Fast	Negative low
Negative	Positive	Fast	Low
Zero	Negative	Fast	Medium
Zero	Zero	Medium	Medium
Zero	Positive	Medium	Fast
Positive	Negative	Low	Fast
Positive	Zero	Negative low	Fast
Positive	Positive	Negative medium	Fast

Keterangan:

- *Negative medium*: motor berputar mundur/CCW dengan kecepatan sedang.
- *Negative low*: motor berputar mundur/CCW dengan kecepatan pelan.

3.2.3. Defuzzifikasi

Defuzzifikasi adalah proses pengubahan nilai dari output *fuzzy* yang dihasilkan menjadi nilai *crisp* [15]. Tahap ini merupakan langkah terakhir sistem kendali *fuzzy*. Proses ini menggabungkan nilai-nilai *fuzzy* yang dihasilkan oleh mesin inferensi *fuzzy* dan mengkonversinya menjadi satu nilai numerik output. Penelitian ini menggunakan defuzzifikasi metode Mamdani agar hasil output lebih presisi, yaitu pada agregasi output untuk setiap aturan, dilakukan pemotongan kurva fungsi keanggotaan output pada nilai keanggotaan input yang relevan. Sedangkan penghitungan output *crisp* menggunakan jenis *centroid* yaitu menghitung titik tengah (*centroid*) dari area di bawah kurva fungsi keanggotaan *fuzzy*, dengan persamaan untuk perhitungan *centroid* seperti berikut,

$$z = \frac{\int \mu(z)z \, dz}{\int \mu(z) \, dz} \quad (15)$$

Dimana:

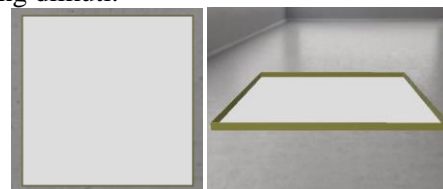
- z adalah variabel output.
- $\mu(z)$ adalah nilai keanggotaan dari z .

3.3. Penyiapan Simulasi

Sebelum melakukan simulasi di Webots, diperlukan beberapa persiapan alat, termasuk pemasangan perangkat lunak Webots dan instalasi pustaka *skfuzzy*. Pemasangan pustaka *skfuzzy* sangat penting untuk memastikan kontroler *fuzzy* berjalan dengan baik.

3.3.1. Penyiapan Arena dan Objek

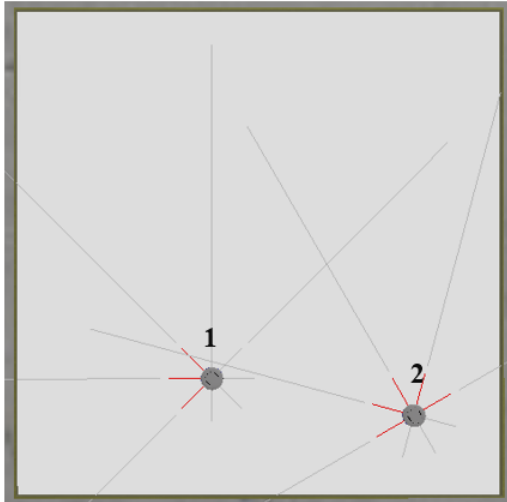
Arena yang akan digunakan berbentuk persegi, dengan meminimalkan warna biru dalam lingkungan simulasi supaya tidak adanya gangguan dalam pendeteksian objek yang diikuti.



Gambar 8. Arena dari Atas dan Samping

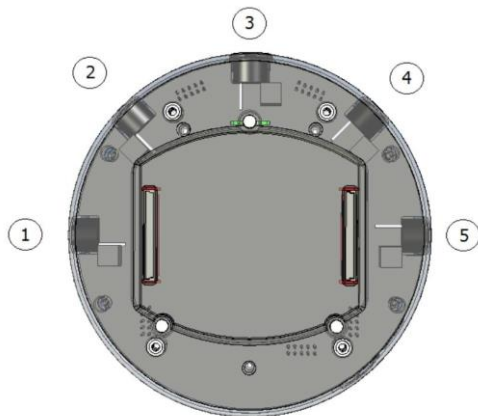
Untuk objek sebagai target dan robot pengikut objek menggunakan robot *Khepera IV*. Robot bernomor 1, dengan algoritma *obstacle avoidance* (menghindari rintangan)

sebagai target objek dan robot nomor 2 adalah robot pendeteksi objek menggunakan kontrol logika *fuzzy* yang terlihat pada Gambar 9.



Gambar 9. Posisi Objek dan Robot Pendeteksi

Robot juga menggunakan sensor inframerah untuk menghindari rintangan yang ada ketika tidak ada target yang terdeteksi oleh kamera, sehingga robot bergerak bebas secara *autonomous*.



Gambar 10. Posisi Sensor Inframerah pada robot *Khepera IV*

Pemilihan sensor inframerah pada robot *Khepera IV*, karena sensor tersebut lebih akurat dalam membaca jarak yang berada didekat robot secara detail.

4. HASIL DAN PEMBAHASAN

Setelah penyiapan simulasi dan pembuatan kontrol *fuzzy* pada robot di aplikasi webots, maka perlu adanya pengujian untuk mendapatkan data respon transien yang dapat dianalisis. Percobaan

dilakukan dengan menempatkan objek pada titik horizontal piksel ke 0 dalam kamera, dimana posisi objek 0 adalah sebelah kiri dengan pergerakan objek ke kiri. Berikut hasil data yang telah didapatkan dari tiga percobaan dalam Tabel 7.

Tabel 7. Hasil Percobaan

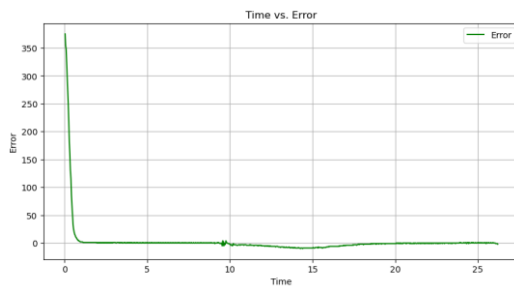
Percobaan ke-	Rata-rata <i>error</i>	<i>Rise time</i>
Percobaan ke-1	-9.258	0.896 s
Percobaan ke-2	-3.804	0.608 s
Percobaan ke-3	-1.687	0.384s

Percobaan pertama menghasilkan nilai rata-rata *error* yang sangat tinggi dan nilai *rise time* yang paling tinggi diantara kedua percobaan lain. Nilai *rise time* yang dihasilkan juga lambat menandakan respon kendali kurang responsif. Percobaan ketiga untuk hasil yang terbaik didapatkan respon transien yang terdapat pada tabel 8.

Tabel 8. Hasil Percobaan ke-3

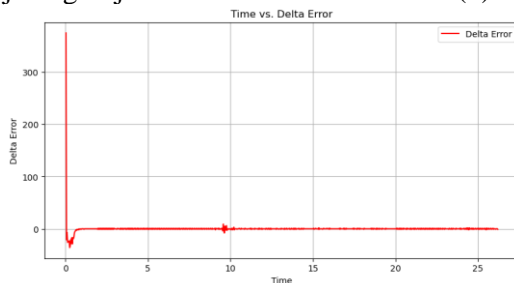
Percobaan ke-3	
<i>Rise Time</i>	0,384 detik
<i>Settling Time</i>	0.576 detik
<i>Steady State-Error</i>	1.77

Grafik percobaan ke-3 terlihat pada Gambar 11 dengan rata-rata *error* -1.687. Percobaan tersebut menghasilkan nilai rata-rata *error* terkecil dari kedua percobaan sebelumnya. Hal tersebut dapat terjadi karena penyesuaian nilai *membership function* sehingga respon yang diberikan lebih baik.



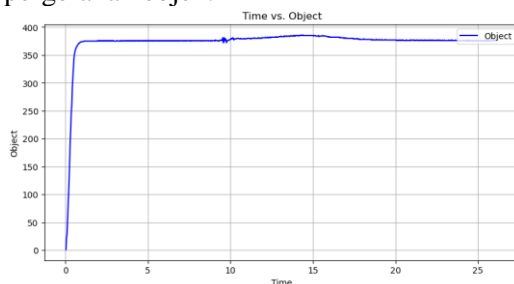
Gambar 11. Grafik Nilai *Error* Terhadap Waktu

Nilai *error* yang muncul saat awal sangat tinggi karena target berada di sebelah kiri garis bantu pada tampilan kamera. Namun pada gambar terlihat bahwa *error* sudah jarang terjadi atau mendekati nilai nol (0).



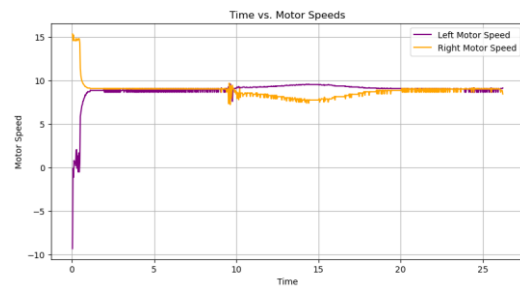
Gambar 12. Grafik Δ error Terhadap Waktu

Gambar 12 merupakan grafik dari nilai Δ error, terlihat bahwa nilai Δ error yang muncul rata rata mendekati nilai nol(0), menandakan pengendali responsif terhadap pergerakan objek.



Gambar 13. Grafik Objek Terhadap Waktu

Dengan *setpoint* letak objek 376, angka tersebut merupakan titik tengah dari letak nilai horizontal kamera. Dalam waktu yang singkat, nilai objek meningkat tajam dari 0 mendekati 376. Setelah peningkatan awal yang cepat, nilai objek mencapai sekitar 380 dan kemudian berfluktuasi di sekitar nilai ini hingga akhir waktu pengamatan.



Gambar 14. Grafik Kecepatan Motor Kanan dan Motor Kiri

Target terletak di sebelah kiri, maka robot berbelok ke kiri dengan kecepatan motor $V_l < V_r$. Pada kisaran waktu 10 sampai 20 detik, target berbelok ke kanan, sehingga kecepatan motor pada robot berubah untuk mengikuti target dengan kecepatan $V_r < V_l$.

5. KESIMPULAN

Setelah melakukan percobaan dan menganalisis terhadap sistem kontrol objek *following* menggunakan logika *fuzzy*, maka dapat ditarik beberapa kesimpulan sebagai berikut:

- Sistem kendali objek *following* menggunakan *fuzzy logic controller* berhasil dilakukan dan berjalan dengan baik.
- Sensor kamera yang digunakan juga berfungsi dengan baik ketika mendeteksi target yang memiliki ukuran kecil.
- Fungsi keanggotaan pada masing-masing *input* dan *output* perlu disesuaikan agar respon robot lebih sensitif terhadap target.
- Respon yang dihasilkan sangat baik, memiliki nilai *rise time* yang tergolong cepat dan tidak mengalami *overshoot*.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Ir. Ardy Seto Priambodo, S.T., M.Eng. selaku Dosen yang membimbing selama proses penyusunan jurnal ini berlangsung. Selanjutnya, penulis ingin berterima kasih kepada pihak yang terlibat dalam memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] D. R. Palupi, D. Syaury, and A. S. Budi, "Object Following Robot berbasis Pembacaan Jarak menggunakan Metode PID Controller," Jul. 2020. [Online]. Available: <http://j-ptiik.ub.ac.id>.
- [2] M. Khairudin, S. Yatmono, A. Nugraha, M. Ikhsani, A. Shah, and M. Hakim, "Object Detection Robot Using Fuzzy Logic Controller Through Image Processing," *Journal of Physics: Conference Series*, vol. 1737, no. 1, p. 012045, Jan. 2021, doi: <https://doi.org/10.1088/1742-6596/1737/1/012045>.
- [3] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965, doi: [https://doi.org/10.1016/s0019-9958\(65\)90241-x](https://doi.org/10.1016/s0019-9958(65)90241-x).
- [4] R. Alamsyah, E. Ryansyah, A. Y. Permana, and R. Mufidah, "SISTEM PENYIRAMAN TANAMAN OTOMATIS MENGGUNAKAN LOGIKA FUZZY DENGAN TEKNOLOGI INTERNET OF THINGS BERBASIS ESP8266 DAN APLIKASI BLYNK," *Jurnal Informatika Dan Teknik Elektro Terapan*, vol. 12, no. 2, Apr. 2024, doi: <https://doi.org/10.23960/jitet.v12i2.4007>.
- [5] M. M. Gulzar, R. P. Singh, and M. Mehra, "HSV Values and OpenCV for Object Tracking," *International Journal of Innovative Research in Computer Science & Technology*, vol. 10, no. 1, pp. 43–48, Jan. 2022, Accessed: Jun. 09, 2024. [Online]. Available: <https://acspublisher.com/journals/index.php/ijircst/article/view/10817>.
- [6] K. D. Irianto, "PENDETEKSI GERAK BERBASIS KAMERA MENGGUNAKAN OPENCV PADA RUANGAN," *publikasiilmiah.ums.ac.id*, Jun. 2010, Accessed: Jan. 01, 2024. [Online]. Available: <https://publikasiilmiah.ums.ac.id/xmlui/handle/11617/1187>.
- [7] Tran Thi Trang and C. Ha, "Irregular moving object detecting and tracking based on color and shape in real-time system," Jan. 2013, doi: <https://doi.org/10.1109/commantel.2013.6482431>.
- [8] S. G. Tzafestas, "Mobile Robot Control and Navigation: A Global Overview," *Journal of Intelligent & Robotic Systems*, vol. 91, no. 1, pp. 35–58, Mar. 2018, doi: <https://doi.org/10.1007/s10846-018-0805-9>.
- [9] M. Sridharan and P. Stone, "Real-time vision on a mobile robot platform," *CiteSeer X (The Pennsylvania State University)*, Jan. 2005, doi: <https://doi.org/10.1109/iro.2005.1545540>.
- [10] S. G. Tzafestas, "Mobile Robot Control and Navigation: A Global Overview," *Journal of Intelligent & Robotic Systems*, vol. 91, no. 1, pp. 35–58, Mar. 2018, doi: <https://doi.org/10.1007/s10846-018-0805-9>.
- [11] S. Lihan, M. Jie and Y. Baoqing, "Fuzzy PID Design of Vehicle Attitude Control Systems," 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 2020, pp. 1826–1830, doi: <https://doi.org/10.1109/CCDC49329.2020.9164275>.
- [12] L. F. Wang, K. C. Tan and V. Prahlad, "Developing Khepera robot applications in a Webots environment," MHS2000. Proceedings of 2000 International Symposium on Micromechatronics and Human Science (Cat. No.00TH8530), Nagoya, Japan, 2000, pp. 71–76, doi: <https://doi.org/10.1109/MHS.2000.903293>.
- [13] A. Ma'arif, A. A. Nuryono and Iswanto, "Vision-Based Line Following Robot in Webots," 2020 FORTEI-International Conference on Electrical Engineering (FORTEI-ICEE), Bandung, Indonesia, 2020, pp. 24–28, doi: <https://doi.org/10.1109/FORTEI-ICEE50915.2020.9249943>.
- [14] M. S. S. Virdaus and E. Ihsanto, "Rancang Bangun Monitoring Dan Kontrol Kualitas Udara Dengan Metode Fuzzy Logic Berbasis Wemos," *Jurnal Teknologi Elektro*, vol. 12, no. 1, p. 22, Jan. 2021, doi: <https://doi.org/10.22441/jte.2021.v12i1.005>.
- [15] S. Neonbeni, G. S. Mada, and F. M. A. Blegur, "Analisis Perbandingan Metode Defuzzifikasi Fuzzy Inference System Mamdani Dalam Penentuan Produksi Tua Kolo (Sopi Timor) 45% Pada Pabrik Sane Up-Ana Kefamenanu," *Jurnal Saintek Lahan Kering*, vol. 5, no. 2, pp. 34–39, Jan. 2023, doi: <https://doi.org/10.32938/slk.v5i2.1994>.

