

## Kendali Robot Beroda sebagai *Self Driving Car* Berbasis *Image Processing* dengan Metode Fuzzy Logic

Sang Made Sedana Yoga<sup>\*1</sup>, Porman Pangaribuan<sup>2</sup>, Azam Zamhuri Fuadi<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom, Indonesia  
Email: <sup>1</sup>sedanayoga@student.telkomuniversity.ac.id, <sup>2</sup>pormanpangaribuan@telkomuniversity.ac.id,  
<sup>3</sup>azamzamhurifuadi@telkomuniversity.ac.id

### Abstrak

Seiring dengan perkembangan zaman, teknologi robotika di bidang industri dituntut untuk semakin berkembang, termasuk sektor industri otomotif. Contohnya, sistem kendali pada mobil secara otonom saat ini banyak dikembangkan oleh perusahaan mobil. Pengembangan teknologi ini, diharapkan dapat meminimalisir kecelakaan yang disebabkan oleh kesalahan manusia dalam mengemudikan mobil. Pada penelitian ini, sistem yang dirancang adalah robot beroda empat yang dapat berjalan secara otomatis berdasarkan deteksi objek. Robot beroda ini memungkinkan untuk pengenalan objek berupa garis layaknya garis marka pada jalan beraspal serta mampu mengendalikan mobil untuk melaju secara otomatis dengan metode logika Fuzzy. Selain itu, penelitian ini juga mengembangkan robot beroda untuk mendeteksi rambu lalu lintas dan lampu lalu lintas. Bila terdapat benda halangan, maka robot akan berbelok ke kanan untuk mengambil jalur kanan lalu menyalip halangan tersebut lalu kembali lagi ke jalur kiri. Ketika mendeteksi rambu stop maka, robot akan berhenti selama 5 detik. Kemudian, ketika robot mendeteksi lampu berwarna merah maka robot akan berhenti hingga lampu berubah menjadi warna hijau. Berdasarkan hasil pengujian yang dilakukan, kinerja pengolahan citra sebagai sistem "*Self Driving Car*" berjalan dengan baik dimana robot bisa mendeteksi jalan, rambu 'STOP', replika mobil dan lampu lalu lintas. Pengolahan citra berbasis *Haar-Like Classifiers* dapat mencapai rata-rata akurasi sebesar 92,7%. Pada sistem kendali berbasis Logika Fuzzy, robot dapat berjalan pada jalurnya dengan baik hingga mencapai rata-rata akurasi sebesar 91,94%.

**Kata Kunci:** *Image Processing*, Logika Fuzzy, OpenCV

### Abstract

*Along with the times, robotics technology in the industrial sector is required to grow, including the automotive industry sector. For example, autonomous car control systems are currently being developed by many car companies. The development of this technology is expected to minimize accidents caused by human error in driving a car. In this study, the system designed is a four-wheeled robot that can run automatically based on object detection. This wheeled robot allows for object recognition in the form of lines like markings on paved roads and is able to control the car to drive automatically with the fuzzy logic method. In addition, this research also develops a wheeled robot to detect traffic signs and traffic lights. If there is an obstacle, the robot will turn right to take the right lane and then overtake the obstacle and then return to the left lane. When it detects a stop sign, the robot will stop for 5 seconds. Then, when the robot detects a red light, the robot will stop until the light turns green. Based on the results of the tests carried out, the performance of image processing as a "Self Driving Car" system is running well where the robot can detect roads, 'STOP' signs, replica cars and traffic lights. Image processing based on Haar-Like Classifiers can achieve an average accuracy of 92.7%. In the Fuzzy Logic-based control system, the robot can run on its path well until it reaches an average accuracy of 91.94%.*

**Keywords:** Fuzzy Logic, *Image Processing*, OpenCV

## 1. PENDAHULUAN

Teknologi robot saat ini berkembang secara pesat dan telah masuk ke berbagai bidang. Hal tersebut dapat dilihat dari semakin maraknya pemanfaatan teknologi yang dapat dioperasikan dan dikendalikan secara otonom. Contohnya, semakin maraknya robot yang dapat membantu manusia dalam kegiatan sehari-hari termasuk dalam pekerjaan. Beberapa pekerjaan manusia dalam kategori sukar, berbahaya

dan membutuhkan tingkat ketelitian yang tinggi, saat ini dapat digantikan oleh peran robot. Tidak terkecuali dengan sistem kendali mobil secara otonom atau bisa disebut *Self Driving*. Saat ini, banyak perusahaan yang sedang mengembangkan sistem tersebut agar bisa diterapkan pada mobil. Alasannya, dengan mengembangkan sistem *self driving*, resiko kecelakaan akibat kesalahan manusia dapat diminimalisir.

Di Indonesia sendiri, tingkat kecelakaan lalu lintas dari tahun ke tahun semakin meningkat. Berdasarkan data [1]. Badan Pusat Statistik (BPS), terjadi peningkatan jumlah kecelakaan dari tahun 2015 hingga tahun 2019. Puncaknya terjadi pada tahun 2019, dimana jumlah kecelakaan lalu lintas sebanyak 116.411 atau naik 6,59% bila dibandingkan dengan tahun 2018 sebesar 109.215 kecelakaan. Jumlah kecelakaan tersebut mengakibatkan korban jiwa sebanyak 175.488 jiwa. Adapun persentase korban yakni, korban luka ringan sebanyak 78,26%, korban luka berat sebanyak 7,11% dan korban meninggal sebanyak 14,63% dari total kecelakaan sepanjang tahun 2019 di Indonesia. Total kerugian yang diperkirakan sepanjang tahun 2019 adalah 254.779 juta rupiah. Selain itu, pada grafik tersebut terlihat sepanjang tahun 2015-2019 terdapat peningkatan jumlah korban yang mengalami luka ringan [3]. Namun, jumlah kasus korban yang meninggal dan luka berat mengalami penurunan.

Melihat banyaknya kecelakaan yang terjadi, banyak penelitian yang dilakukan untuk mengembangkan sistem kendali mobil secara otonom. Adapun, minat masyarakat tentang sistem *Self Driving Car* atau kendali mobil secara otonom sebelumnya sudah diteliti oleh D. Howard dkk pada suatu lokasi tepatnya di Museum Sains Barkley [2]. Hasil dari survey menunjukkan, dari 107 Responden, 46% percaya bahwa sistem kendali mobil secara otonom harus beroperasi dengan lalu lintas normal, 38% di jalur terpisah, dan 11% tidak memiliki pendapat. Berdasarkan data tersebut, hal yang paling penting ketika merancang sebuah sistem kendali mobil secara otonom adalah sistem dapat beroperasi pada lalu lintas normal [4].

Lewat implementasi dari robot beroda, penulis menciptakan suatu sistem yang dapat mengendalikan robot beroda agar melaju secara otonom berbasis citra atau gambar. Melalui *image processing*, citra atau gambar berupa garis jalan atau marka jalan bisa diproses dan dapat dijadikan navigasi untuk kendali robot beroda [3]. Sistem yang dibuat memungkinkan robot beroda dapat berjalan dengan kendali otonom. Selain itu, robot beroda dapat membaca marka jalan, lampu lalu lintas dan rambu 'STOP' lalu menjalankan perintah sesuai dengan pembacaan menggunakan logika Fuzzy sebagai metode kendalinya.

Tujuan dari penelitian ini adalah merancang robot beroda empat yang berjalan sesuai Deteksi marka jalan yang sudah ditentukan, dengan akurasi minimal 80%. Lalu, merancang robot beroda empat yang dapat membaca traffic light, rambu lalu lintas dan halangan di depannya lalu menjalankan perintah berdasarkan objek yang terdeteksi, dengan akurasi minimal 80%.

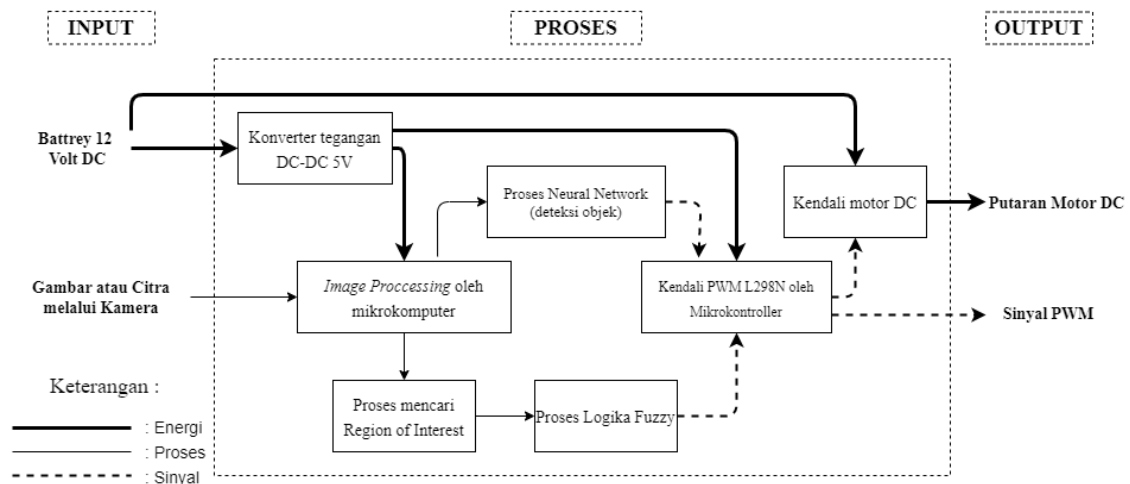
## 2. METODE PENELITIAN

### 2.1. Desain Konsep Solusi

Robot beroda empat yang dibuat menggunakan sistem pemrosesan secara *On Board* memakai Raspberry Pi 4[4]. Mikrokomputer ini berfungsi sebagai pemrosesan gambar atau citra yang menjadi input pada robot agar dapat berjalan sesuai pada jalurnya. Selain itu, mikrokomputer dipakai dalam pemrosesan neural network untuk mendeteksi beberapa objek, antara lain traffic light, rambu lalu lintas dan halangan yang berada di depan robot beroda. Citra atau gambar berasal dari kamera 5MP yang dipasang pada Raspberry Pi [5].

Komponen lain yang dipakai yaitu Motor Driver L298N dan Motor DC [6]. Raspberry Pi juga berfungsi sebagai mikrokontroler Motor Driver. Dengan memberikan sinyal PWM, Raspberry Pi dapat mengendalikan motor driver L298N untuk menggerakkan motor DC sesuai dengan hasil dari pemrosesan citra atau gambar. Sumber tegangan yang dipakai berasal dari baterai yang terpasang pada robot. Baterai yang terpasang juga terhubung dengan konverter tegangan. Fungsinya untuk mengalirkan tegangan yang sesuai dengan Raspberry Pi. Semua komponen nantinya akan terpasang pada sasis robot dan terhubung menggunakan kabel.

Metode kendali yang dipakai adalah metode Logika Fuzzy. Perhitungan metode kendali berbasis logika fuzzy dilakukan secara *on board* pada Raspberry Pi. Dimana, metode kendali ini diterapkan untuk mengatur nilai PWM pada motor DC. Motor DC yang digunakan sebanyak 4 buah, berfungsi sebagai aktuator pada robot [7]. Proses dan gambaran secara umum mengenai konsep solusi, dapat dilihat pada Gambar 1 berikut.



Gambar 1. Diagram Fungsi

Gambar 1 merupakan desain konsep solusi dari robot beroda sebagai *Self Driving Car*. Dimana, Sumber energi listrik dari sistem adalah baterai dengan tegangan 12 Volt. Konverter DC-DC 5V berfungsi untuk mengalirkan daya ke mikrokomputer agar sesuai dengan tegangan inputnya. Proses pengolahan citra pada mikrokomputer dibagi menjadi 2 macam, yaitu deteksi jalur dan deteksi benda. Jalur dapat terdeteksi dengan cara menentukan titik tengahnya menggunakan logika *Region of Interest* (ROI). Logika ini dapat menentukan garis tengah antara 2 marka di depannya sebagai acuan arah robot bergerak. Adapun deteksi benda menggunakan metode *Neural Network*. Metode ini memakai sampel Positif dan sampel Negatif sebagai data *Training* pada Mikrokomputer. Benda yang akan dideteksi adalah *traffic light*, rambu lalu lintas dan *Obstacle* atau halangan yang berada di depan robot beroda. Hasil *image processing* akan memberikan sinyal perintah berupa sinyal PWM ke *motor driver*. *Motor driver* menggerakkan motor DC sesuai sinyal PWM yang diterima dari mikrokomputer. Metode kendali yang digunakan dalam menggerakkan motor DC adalah Logika Fuzzy.

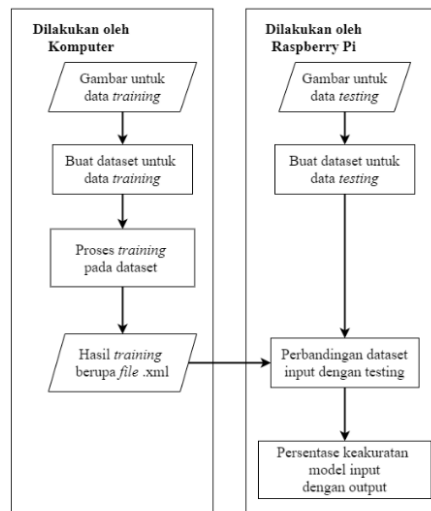
## 2.2. Image Processing

*Image Processing* pada robot beroda dibagi menjadi 2 macam, yakni deteksi objek dengan metode *neural network* dan deteksi jalan dengan menggunakan metode *region of interest*.

### a. Deteksi Benda dengan metode *Neural Network*

Pada metode *Neural Network*, terdapat input dataset yang melalui proses *training*. Data *training* tersebut, nantinya akan menjadi pembanding antara dataset dengan data citra atau gambar yang di tangkap oleh kamera. Sehingga, keakuratan dalam mendeteksi suatu objek dapat tercapai. Adapun struktur dan urutan *training* untuk *haar-like feature* dan *training* pada *backpropagation* dapat dilihat pada Gambar 2.

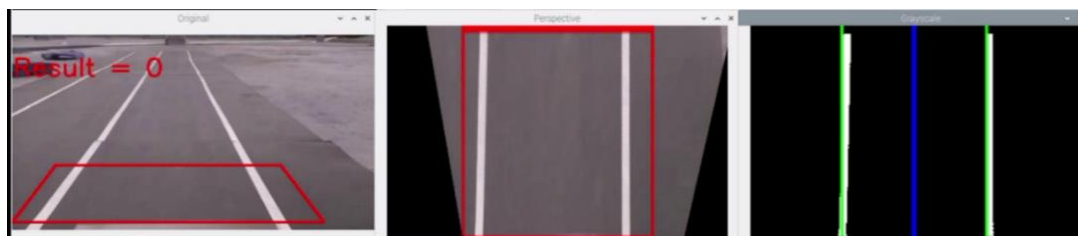
Pada Gambar 2, adalah struktur pembuatan data set yang akan diujikan ke dalam robot beroda. Setelah data di *training* pada komputer dan sudah mencapai tingkat keakuratan maksimal maka, akan dipindahkan ke dalam Raspberry-pi[8]. Setelah itu, data akan dijalankan pada robot beroda. Proses *training* menghasilkan file .xml, dimana file ini merupakan kumpulan *training* dari *data preparation* yang sudah disiapkan sebelumnya.



Gambar 2. Training dataset pada robot beroda

#### b. Deteksi Jalan Dengan Metode *Region of Interest (ROI)*

Pada deteksi jalan dengan metode *Region of Interest* garis jalan di deteksi dengan cara mengubah citra RGB menjadi citra *grayscale* terlebih dahulu. Citra *grayscale* selanjutnya diubah menjadi gambar *threshold* agar gambar *grayscale* berubah menjadi warna hitam (0) dan putih (255) saja tanpa adanya gradasi. Lalu, gambar *threshold* diproses supaya dapat mendeteksi bagian tepi dengan cara memasukkan fungsi *canny* pada OpenCV. Selanjutnya citra diblur dengan *gaussian blur*, agar garis-garis tepi menjadi lebih halus. Setelah deteksi tepi dengan Fungsi *Canny* di atur agar maksimal, maka warna pada deteksi diubah jadi putih layaknya garis marka jalan. Adapun hasil akhir dari deteksi dengan fungsi *Canny* dapat dilihat pada Gambar 3 berikut.



Gambar 3. Hasil Pengolahan Citra Dengan Metode ROI

Pada Gambar 3, *Region of Interest* digunakan untuk membagi citra yang ditangkap oleh kamera menjadi beberapa bagian. Pembagian pada citra berfungsi agar citra yang diolah tidak terlalu banyak, sehingga software dapat memproses data yang lebih sedikit. Dalam mengambil citra, digunakan fungsi `cap = cv2.VideoCapture(0)`. ROI dibuat menggunakan koordinat x dan y (pixel) untuk menentukan titik yang akan digunakan dalam membatasi area yang akan dilakukan pemrosesan [9].

### 2.3. Perancangan Kontroler Logika Fuzzy (FLC)

Untuk merancang pengendali dalam penelitian ini, maka diterapkan logika fuzzy pada pengendali motor DC dengan menggunakan metode Mamdani[10]. Sistem kendali dirancang dengan dua buah masukan yang berupa Error yang merupakan selisih antara set point dengan pembacaan sensor kamera dan Delta Error yang merupakan turunan pertama dari nilai Error. Kedua input Error dan Delta Error akan diproses oleh *fuzzy logic controller (FLC)* untuk mendapatkan nilai output berupa sinyal PWM untuk mengontrol kecepatan putar motor DC [11].

#### 1. Pembentukan himpunan fuzzy

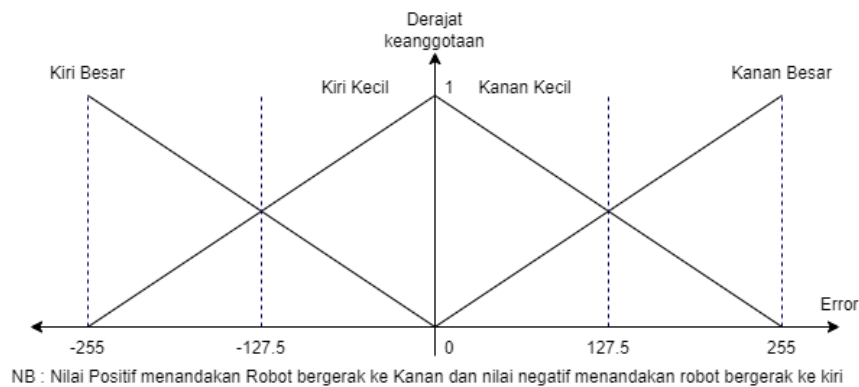
Pengolahan data dilakukan dengan menentukan variabel dan semesta pembicaraan, dilanjutkan

dengan pembentukan himpunan fuzzy. Penentuan variabel dan semesta pembicaraan dapat dilihat pada Tabel 1 berikut.

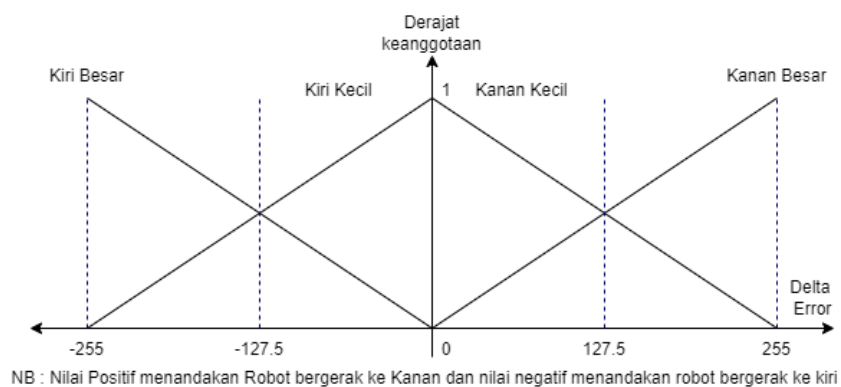
Tabel 1. Penentuan Variabel dan Semesta Pembicaraan

Fungsi	Nama Variabel	Semesta Pembicaraan	Keterangan	Himpunan Fuzzy
Input	Error	[0-255]	Error Jalan	Kecil, Besar
	Delta error	[0-255]	Turunan pertama dari Error	Kecil, Besar
Output	Kecepatan motor DC	[0-100]	Kecepatan Motor	Lambat, Cepat

Tabel 1 merupakan penentuan variabel dan semesta pembicaraan. Tujuannya adalah untuk mengelompokkan nilai tegas (crisp) menjadi himpunan fuzzy. Pada input error dan delta error, himpunan fuzzy yang dipakai adalah kecil dan besar. Sedangkan pada output nilai kecepatan motor DC, himpunan fuzzy yang dipakai adalah lambat dan cepat [12]. Adapun fungsi keanggotaan Input dan Output Fuzzy dapat dilihat pada Gambar 4 di bawah ini.



(a)



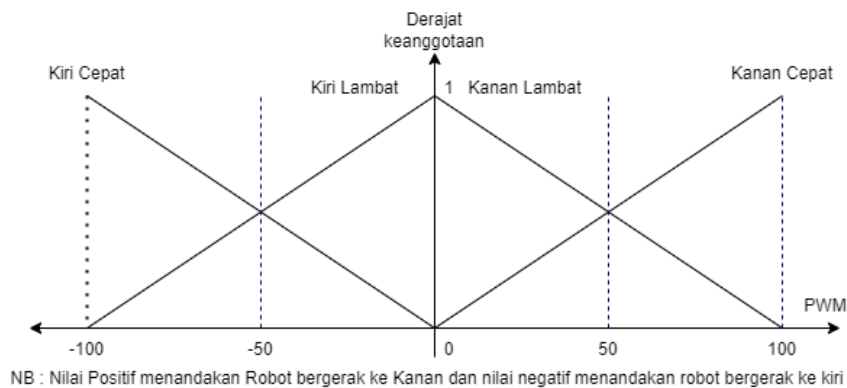
(b)

Gambar 4. Fungsi Keanggotaan Input Logika Fuzzy

Pada Gambar 4, fungsi keanggotaan input fuzzy ada 2 yakni nilai error dan delta error. Nilai input tersebut berupa nilai tegas (crisp) 0 sampai 255. Nilai tersebut diperoleh dari nilai minimum dan nilai maksimum pin PWM Raspberry Pi. Nilai tegas tersebut diubah menjadi himpunan-himpunan input fuzzy kecil dan besar. Sedangkan, untuk Fungsi keanggotaan Output nilai PWM dapat dilihat pada Gambar 5.

Gambar 5 adalah fungsi keanggotaan output fuzzy, dimana sinyal Output PWM berupa nilai tegas (crisp) 0 sampai 100. Nilai tersebut diperoleh dari maksimum library pin PWM Raspberry Pi. Nilai tegas

tersebut diubah menjadi himpunan-himpunan input fuzzy pelan dan cepat. Nilai PWM nantinya akan menggerakkan Motor DC untuk maju, belok secara lambat atau belok secara cepat.



Gambar 5. Fungsi Keanggotaan Output Logika Fuzzy

## 2. Aplikasi fungsi implikasi

Menyusun basis aturan, yaitu aturan-aturan berupa implikasi fuzzy yang menyatakan relasi antara variabel input dengan dengan variabel output. Pada penelitian ini menggunakan metode Mamdani, fungsi implikasi yang digunakan adalah dengan bentuk sebagai berikut:

- [R1] IF Error KECIL And Delta Error KECIL THEN Output BELOK PELAN.
- [R2] IF Error KECIL And Delta Error BESAR THEN Output BELOK PELAN.
- [R3] IF Error BESAR And Delta Error KECIL THEN Output BELOK CEPAT.
- [R4] IF Error BESAR And Delta Error BESAR THEN Output BELOK CEPAT.

## 3. Komposisi Aturan

Apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Metode yang digunakan dalam melakukan inferensi sistem fuzzy, yaitu Metode max (maximum). Pada metode ini, solusi himpunan fuzzy diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakan nilai tersebut untuk memodifikasi daerah fuzzy dan mengaplikasikannya ke output.

## 4. Penegasan (defuzzy)

Pada metode ini, solusi tegas diperoleh dengan cara mengambil titik pusat atau titik berat dari daerah fuzzy. Adapun persamaan matematis untuk mencari nilai output defuzzifikasi dapat dilihat pada persamaan 1.

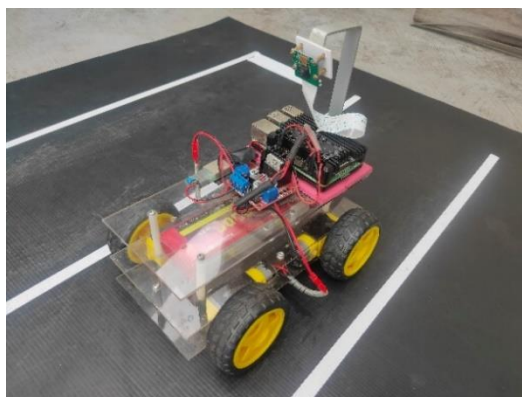
$$Z^* = \frac{\int \mu(z) \cdot z \, dz}{\int \mu(z) \, dz} \quad (1)$$

Pada persamaan 1, cara menghitung output defuzzifikasinya adalah dengan menghitung nilai momen lalu dibagi dengan luas areanya. Nilai momen didapat dengan mengalikan tiap persamaan garis dengan nilai 'z'. Sedangkan, luas area dihitung cara mencari luas masing-masing bangun datar yang terbentuk dengan rumus bangun datar atau menggunakan rumus integral.

## 2.4. Implementasi Alat

Pada sistem robot ini, perangkat keras yang dipakai adalah Raspberry Pi sebagai Mikrokomputer sekaligus sebagai mikrokontroler, Motor Driver L298N sebagai pengontrol motor DC dan motor DC sebagai penggerak roda pada robot beroda yang berjumlah 4 buah. Adapun hasil implementasi desain robot beroda dapat dilihat pada Gambar 6.

Gambar 6 merupakan implementasi dari desain robot beroda. Selain itu, terdapat replika jalan berwarna hitam dengan marka berwarna putih sebagai jalur robot beroda. Adapun perangkat mekanik lain yang digunakan yaitu tiruan atau replika rambu STOP, traffic light, mobil mainan yang ukurannya menyesuaikan dengan kebutuhan object detection [13].



Gambar 6. Implementasi Desain Robot Beroda

Adapun skema jalur yang akan dilalui oleh robot beroda dapat dilihat pada Gambar 7 berikut.



Gambar 7. Gambaran Fungsi Robot Beroda

Pada Gambar 7, terdapat perintah yang diberikan ketika robot mendeteksi objek. Saat robot mendeteksi rambu 'STOP', robot akan berhenti selama 5 detik. Lalu, ketika mendeteksi lampu lalu lintas, robot akan berhenti ketika lampu LED warna merah menyala dan tetap berjalan ketika lampu LED warna hijau menyala. Bila terdapat benda halangan, maka robot akan berbelok ke kanan untuk mengambil jalur kanan lalu menyalip halangan tersebut lalu kembali lagi ke jalur kiri. Pada ujung jalan, garis putih pada ujung memberikan tanda untuk robot beroda berbelok dan masuk ke jalur di sampingnya.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Pengujian Deteksi Objek Dengan Metode Haar-like Classifiers

Pada tahap ini akan dilakukan pengujian mendeteksi objek terhadap sistem robot beroda. Lewat

pengujian ini, kita dapat melihat seberapa respon yang diberikan system ketika mendeteksi objek. Adapun cara pengujiannya adalah objek deteksi diletakkan pada area depan robot beroda. Posisi objek deteksi menyesuaikan dengan frame kamera supaya objek dapat terdeteksi. Lalu lewat image processing, robot dapat menghitung jarak objek deteksi berdasarkan area deteksinya. Bila area deteksinya semakin besar maka jarak objek semakin dekat, dan sebaliknya. Selain itu, pengujian ini menggunakan penggaris agar menjadi perbandingan pada pengujian lewat deteksi objek. Hasil pengukuran deteksi objek dapat dilihat pada Gambar 8 berikut.



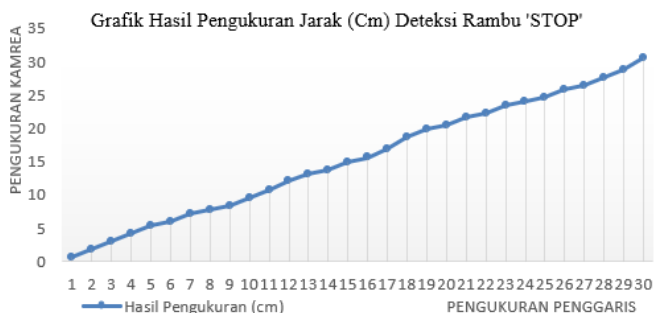
Gambar 8. Hasil Pengukuran Deteksi Rambu 'STOP'

Gambar 8 merupakan hasil pengukuran deteksi rambu 'STOP'. Berdasarkan hasil pengukuran, maksimal jarak yang bisa terukur lewat deteksi rambu 'STOP' adalah 30 Cm. Selain itu, nilai maksimal pixel yang terukur adalah 102 pixel. Lalu, pada pengukuran jarak deteksi replika mobil, didapat nilai perbandingan antara pengukuran menggunakan penggaris dengan pengukuran lewat deteksi objek. Hasil pengukuran Objek deteksi replika mobil bila ditinjau dari robot beroda dapat dilihat pada Gambar 9 berikut.



Gambar 9. Hasil Pengukuran Deteksi Replika Mobil

Gambar 9 merupakan hasil pengukuran deteksi replika mobil. Berdasarkan hasil pengukuran, maksimal jarak yang bisa terukur lewat deteksi replika mobil adalah 20 Cm. Selain itu, nilai maksimal pixel yang terukur adalah 158,86 pixel. Kemudian, pada pengukuran jarak deteksi lampu lalu lintas, didapat nilai perbandingan antara pengukuran menggunakan penggaris dengan pengukuran lewat deteksi objek. Hasil pengukuran objek lampu lalu lintas bila ditinjau dari robot beroda dapat dilihat pada Gambar 10 berikut.



Gambar 10. Hasil Pengukuran Deteksi Lampu Lalu Lintas



Gambar 10 merupakan hasil pengukuran deteksi lampu lalu lintas. Berdasarkan hasil pengukuran, maksimal jarak yang bisa terukur lewat deteksi replika adalah 20 Cm. Selain itu, nilai maksimal pixel yang terukur adalah 88 pixel. Dari besarnya nilai error yang dihasilkan, dapat dihitung berapa persen akurasi dan persen errornya. Nilai error berasal dari selisih antara nilai pengukuran menggunakan penggaris dengan nilai pengukuran menggunakan deteksi objek. Adapun persen akurasi dan persen error pada pengujian deteksi objek dapat dilihat pada Tabel 2 berikut.

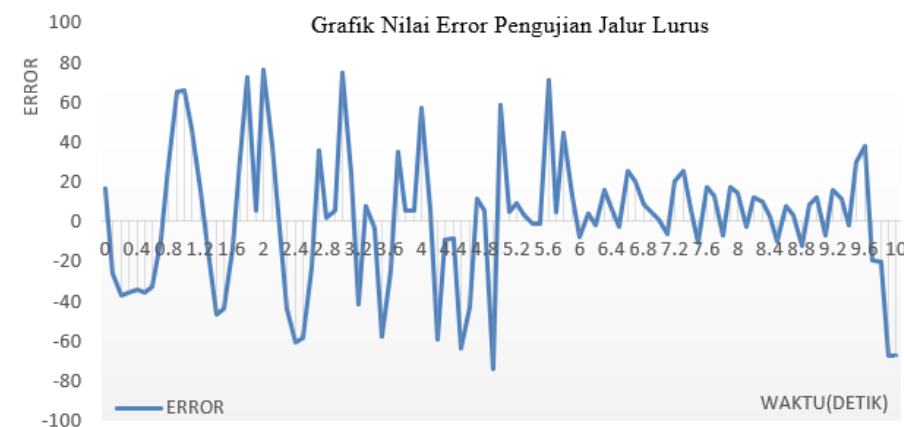
Tabel 2. Hasil Pengujian Deteksi Objek

No	Pengujian	Nilai Akurasi (%)	Nilai Error (%)
1	Jalur lurus	92,77	7,23
2	Jalur belok kiri	92,98	7,02
3	Jalur Belok kanan	91,42	8,58
<b>Rata Rata</b>		<b>91,94</b>	<b>8,06</b>

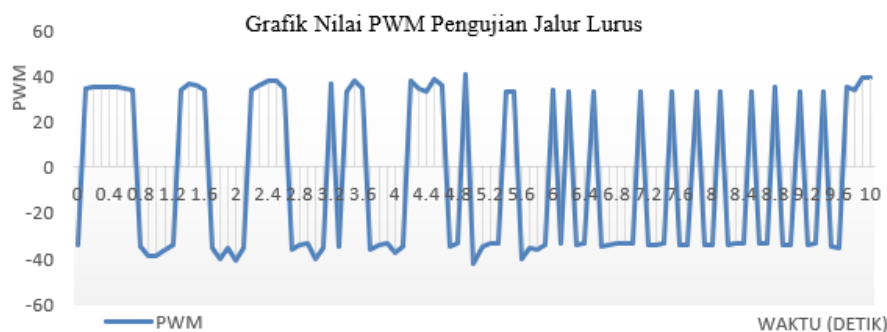
Berdasarkan data pengukuran objek deteksi pada tabel 2, kinerja *image processing* sangat efektif untuk menentukan jarak dari suatu objek. Semakin dekat objek yang terdeteksi, maka nilai Pixel gambar yang terdeteksi semakin besar. Hal tersebut menandakan nilai Pixel gambar berbanding terbalik dengan jarak objek yang terdeteksi Nilai persen akurasi pada deteksi rambu ‘STOP’ mencapai 96,24%. Pada deteksi replika mobil di dapat nilai persen akurasi sebesar 90,49%. Sedangkan, nilai persen akurasi pada deteksi lampu lalu lintas sebesar 91,36%. Sehingga rata-rata total akurasi yang di dapat adalah 92,7%. Nilai pengukuran deteksi benda memiliki rentang yang terbatas yaitu 1-30 untuk rambu ‘STOP’ dan 1-20 untuk replika mobil dan lampu lalu lintas. Hal tersebut dikarenakan ukuran frame yang relatif kecil yakni 210x140p.

### 3.2. Pengujian Kinerja Image Processing Untuk Gerak Robot Berbasis Kendali Logika Fuzzy

Pada pengujian ini, Robot bergerak mengikuti jalur yang sudah ditentukan. Dimana, jalur tersebut terdiri dari 2 garis putih di sisi kiri dan kanan robot pada bidang yang berwarna hitam. Cara pengujian robot beroda adalah menjalankan robot ketika berada pada jalur. Adapun selang waktu pengambilan data adalah 0,1 detik. Pada pengujian ini, robot dijalankan selama 10 detik pada masing-masing kondisi saat lurus, berbelok ke kiri dan berbelok ke kanan. Maka pada pengujian ini, didapat 100 data hasil pengujian di masing-masing kondisi jalur. Nilai error berasal dari perhitungan selisih antara setpoint (0) dengan hasil pembacaan garis tengah oleh robot. Sedangkan, delta error di dapat dari selisih dari nilai error yang di dapat dan nilai error sebelumnya. Lalu, nilai PWM di dapat dari perhitungan logika fuzzy dengan input nilai error dan delta error. Adapun hasil pengujian robot pada jalur lurus dapat dilihat pada Gambar 11 berikut.



(a)



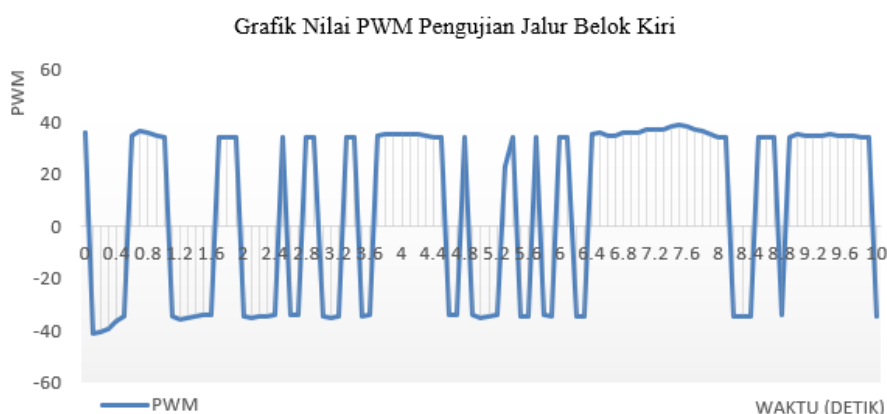
(b)

Gambar 11. Hasil Pengujian Robot di Jalur Lurus

Pada Gambar 11, nilai error yang dihasilkan menjadi perhitungan untuk nilai delta errornya. Pada grafik nilai PWM, nilainya berkebalikan dengan nilai error. Hal tersebut menandakan, jika nilai errornya negatif maka robot akan berbelok ke kiri. Agar dapat berbelok ke kiri, maka nilai motor DC kanan berputar sesuai dengan nilai PWM hasil perhitungan logika fuzzy, dan sebaliknya. Adapun hasil pengujian robot di jalur belok kiri dapat dilihat pada Gambar 12 berikut.



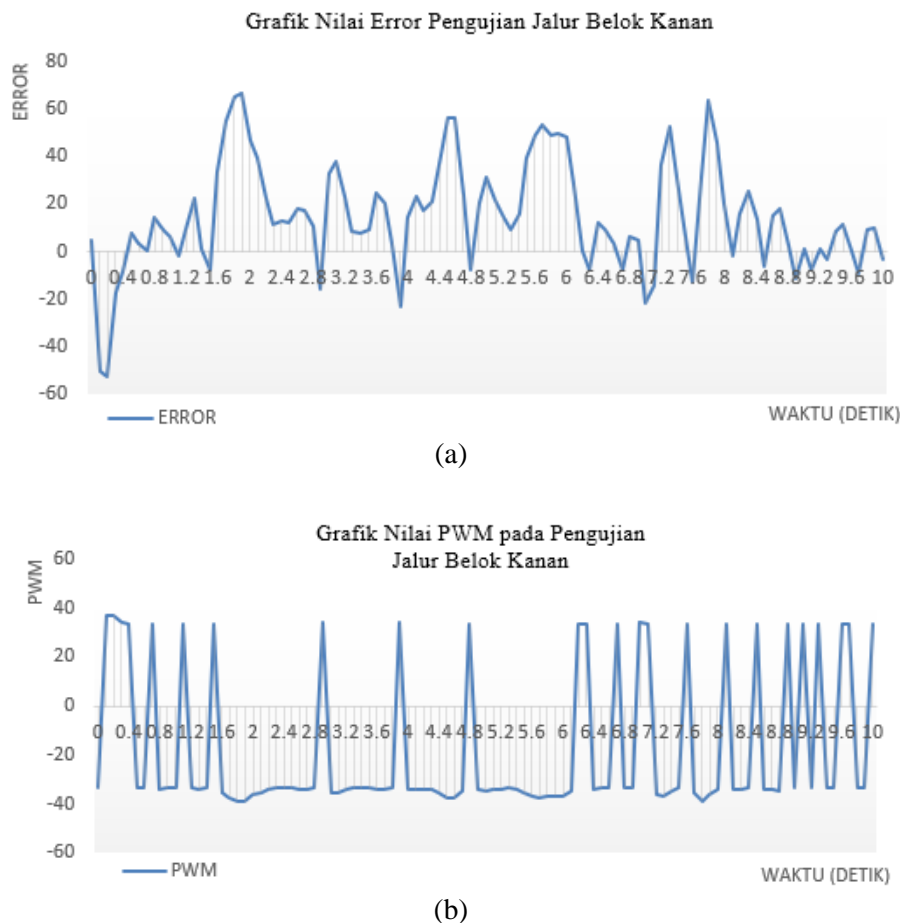
(a)



(b)

Gambar 12. Hasil Pengujian Robot di Jalur Belok Kiri

Pada Gambar 12, nilai error yang dihasilkan juga menjadi perhitungan untuk nilai delta errornya. Namun, ketika robot berbelok ke kiri, nilai errornya cenderung negatif. Hal tersebut menunjukkan jalur yang terdeteksi berbelok ke kiri dan robot harus berbelok ke kiri. Adapun hasil pengujian robot di jalur belok kanan dapat dilihat pada Gambar 13 berikut.



Gambar 13. Hasil Pengujian Robot di Jalur Belok Kanan

Gambar 13, nilai error yang dihasilkan juga menjadi perhitungan untuk nilai delta errornya. Namun, ketika robot berbelok ke kanan, nilai errornya cenderung positif. Hal tersebut menunjukkan jalur yang terdeteksi berbelok ke kanan dan robot harus berbelok ke kanan. Dari hasil pengujian tersebut, besar nilai error yang dihasilkan dapat dihitung berapa persen akurasi dan persen errornya. Perhitungan persen error dan persen akurasi menggunakan nilai acuan 255 (maksimal nilai input logika fuzzy). Adapun persen akurasi dan persen error pada pengujian robot dapat dilihat pada Tabel 3 berikut.

Tabel 3. Nilai Akurasi Pada Pengujian Robot

No	Pengujian	Nilai Akurasi (%)	Nilai Error (%)
1	Jalur lurus	92,77	7,23
2	Jalur belok kiri	92,98	7,02
3	Jalur Belok kanan	91,42	8,58
	<b>Rata Rata</b>	<b>91,94</b>	<b>8,06</b>

Berdasarkan hasil pengujian pada Tabel 3, dapat dianalisis bahwa robot beroda dapat berjalan secara otonom mengikuti jalur yang telah disediakan. Ketika robot beroda bergerak mengikuti jalan yang lurus maka nilai PWM dan error akan terlihat sama rata (nilai negatif dan positifnya). Hal tersebut dikarenakan robot akan berbelok ke kanan jika nilai errornya positif dan berbelok ke kiri jika nilai errornya negatif secara bergantian. Pada saat robot berbelok ke kiri, nilai error banyak bernilai negatif karena mendeteksi jalan yang berbelok ke arah kiri namun tidak secara signifikan karena jalur yang di buat tidak memiliki belokan tajam ke kiri. Untuk dapat bergerak ke arah kiri, maka nilai PWM motor DC kanan akan berputar sesuai dengan nilai output logika Fuzzynya. Pada saat robot berbelok ke kanan, nilai error akan banyak bernilai positif karena mendeteksi jalan yang berbelok ke arah kanan namun

tidak secara signifikan karena jalur yang di buat tidak memiliki belokan tajam ke kanan. Untuk dapat bergerak ke arah kiri, maka nilai PWM motor DC kiri akan berputar sesuai dengan nilai output logika Fuzzynya. Kendali motor DC dengan menggunakan metode logika Fuzzy dapat membuat kecepatan motor DC Stabil. Hal tersebut dapat dilihat pada grafik PWM nya. Dimana, Ketika nilai Error dan Delta Errornya berubah secara signifikan, PWM motor DC menyesuaikan kecepatan berdasarkan perhitungan logika Fuzzy.

### 3.3. Pengujian Kecepatan Robot Beroda Dan Sudut Belok Berdasarkan Input Nilai PWM

Untuk mengetahui kemampuan robot beroda ketika bergerak sesuai dengan jalur yang sudah ditentukan maka perlu adanya pengujian kecepatan dan sudut belok. Pengujian kecepatan bertujuan untuk mengetahui kecepatan robot beroda secara linear. Kecepatan linear dan kecepatan putaran dipengaruhi oleh nilai PWM yang ditentukan, jumlah motor DC yang dipakai, diameter roda serta berat robot tersebut. Pada pengujian ini, robot beroda memiliki jari-jari roda 3,5 cm dan beratnya 0,91 Kg. Pengujian dilakukan dengan 2 cara yaitu menggunakan Tachometer dan tali meteran. Pada pengujian emnggunakan Tachometer, robot berjalan pada jalur yang sudah ditentukan dan dihitung kecepatan putaran rodanya menggunakan Tachometer. Hasil pengukuran didapat nilai kecepatan rata-rata dalam satuan RPM (revolutions per minute). Adapun hasil pengukuran kecepatan putaran roda robot dapat dilihat pada Tabel 4 berikut.

Tabel 4. Hail Pengukuran Kecepatan Putaran Roda Robot Beroda

No	Percobaan	Kecepatan Putaran (RPM)	Kecepatan Putaran (Rad/s)	Kecepatan Linear (m/s)
1	I	133,2	13,94	0,49
2	II	143,8	15,05	0,53
3	III	151,4	15,85	0,55
4	IV	139,5	14,6	0,51
5	V	136,3	14,27	0,5
<b>Rata-rata</b>		<b>14.84</b>	<b>14,74</b>	<b>0,52</b>

Pada Tabel 4, pengujian kecepatan putaran (dalam satuan RPM) dilakukan sebanyak 5 kali. Dimana, dari hasil pengujian dihitung nilai rata-rata kecepatan putaran dalam satuan Rad/s dan kecepatan linear dalam satuan m/s. Hasil pengujian didapatkan nilai rata-rata kecepatan putarannya yakni 14.74 Rad/s dan kecepatan linearnya yakni 0.52 m/s. Kemudian ketika robot berjalan pada jalur, diukur waktu tempuh berdasarkan Panjang lintasannya. Cara pengujian yang kedua adalah menggunakan meteran tali dan stopwatch. Pengukuran dilakukan dengan menghitung berapa lama robot menempuh jalur yang sudah ditentukan. Pada pengujian ini, Panjang jalurnya adalah 10,75 m. Lewat operasi hitung dan pengujian sebanyak 5 kali, maka didapat rata-rata kecepatan linearnya. Adapun hasil pengukuran kecepatan linear robot dapat dilihat pada Tabel 5 berikut.

Tabel 5. Kecepatan Putaran Roda Robot Beroda

No	Percobaan	Kecepatan Putaran (RPM)	Kecepatan Putaran (Rad/s)	Kecepatan Linear (m/s)
1	I	133,2	13,94	0,49
2	II	143,8	15,05	0,53
3	III	151,4	15,85	0,55
4	IV	139,5	14,6	0,51
5	V	136,3	14,27	0,5
<b>Rata-rata</b>		<b>14.84</b>	<b>14,74</b>	<b>0,52</b>

Pada Tabel 5, pengujian kecepatan linear dilakukan sebanyak 5 kali. Dimana, dari hasil pengujian dihitung nilai rata-rata kecepatan linear dalam satuan m/s. Hasil pengujian didapatkan nilai rata-rata kecepatan linearnya yakni 0.52 m/s. Selanjutnya pada pengujian ini, dicari hubungan antara nilai PWM

dengan kecepatan putaran yang dihasilkan. Adapun hubungan antara nilai PWM dengan kecepatan putaran dalam satuan Radian Per Detik dapat dilihat pada Gambar 14 berikut.



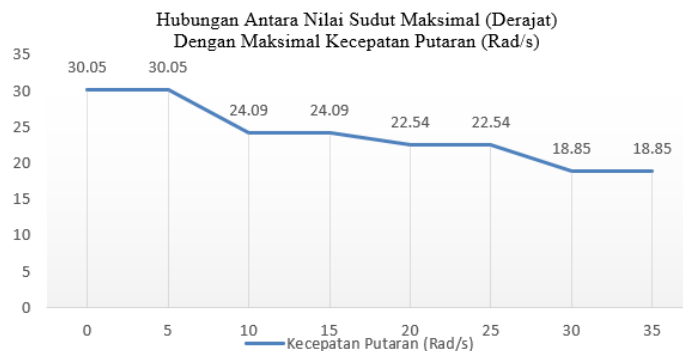
Gambar 14. Grafik Nilai PWM Dan Kecepatan Putaran

Pada Gambar 14, pengujian dilakukan dengan menjalankan robot beroda yang nilai PWM nya sudah di atur. Dimana agar bisa bergerak, nilai PWM minimal yang diperlukan adalah 15. Lalu, kecepatan putaran maksimal yang di dapat (saat nilai PWM 100) adalah 39,67 Rad/s. Adapun hubungan antara nilai PWM dengan kecepatan linear dalam satuan Meter Per Detik dapat dilihat pada Gambar 15 berikut.



Gambar 15. Grafik Nilai PWM Dengan Kecepatan Linear

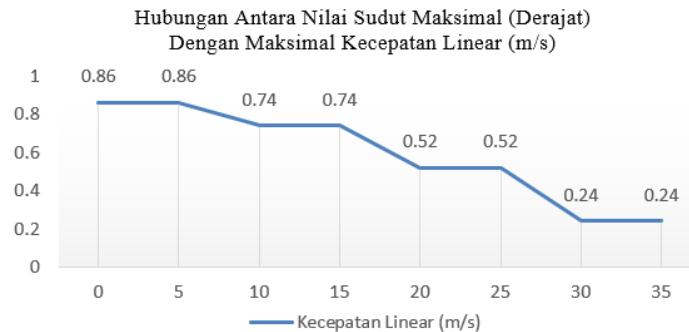
Pada Gambar 15, pengujian dilakukan dengan menjalankan robot beroda yang nilai PWM nya sudah di atur. Dimana, agar bisa bergerak nilai PWM minimal yang diperlukan adalah 15. Lalu, kecepatan linear maksimal yang di dapat (saat nilai PWM 100) adalah 1.21 m/s. Adapun hubungan antara nilai sudut maksimal yang dapat dilalui robot beroda dengan kecepatan putaran dalam satuan Radian Per Detik dapat dilihat pada Gambar 16 berikut.



Gambar 16. Grafik Nilai Sudut Maksimal Dengan Kecepatan Putaran

Pada Gambar 16, pengujian dilakukan dengan menjalankan robot beroda yang nilai PWM nya sudah di atur. Lalu, robot dijalankan pada jalur yang memiliki sudut belok. Hasil pengujian menunjukkan sudut maksimal yang dapat dilalui robot adalah 35 derajat dengan maksimal kecepatan

putarannya adalah 18,85 Rad/s. Adapun hubungan antara nilai sudut maksimal yang dapat dilalui robot beroda dengan nilai kecepatan linear dalam satuan Meter Per Detik dapat dilihat pada Gambar 17 berikut.



Gambar 17. Grafik Nilai Sudut Maksimal Dengan Kecepatan Linear

Berdasarkan hasil pengujian, dapat di analisis bahwa robot beroda memiliki kemampuan untuk berbelok dengan nilai maksimal sudut belok yang berbeda-beda. Hal tersebut dipengaruhi oleh kecepatan dari robot tersebut. Adapun penjelasannya yaitu sebagai berikut. Nilai PWM berbanding lurus dengan kecepatan putaran roda. Pengujian ini dilakukan pada saat robot berjalan dengan nilai PWM yang sudah di atur. Sehingga, kecepatan putarannya lebih lambat dibandingkan putaran roda bila tanpa beban maupun tanpa gesekan pada bidang datar. Nilai Sudut maksimal yang dapat dilalui oleh robot berbanding terbalik dengan maksimal nilai PWM nya. Pada pengujian ini PWM maksimal yang digunakan adalah 60 (rentang 0 – 100). Saat pwm 50, sudut maksimal yang dilalui adalah 5 derajat. Bila lebih besar dari 60, robot tidak bisa berbelok karena terlalu cepat. Adapun untuk kecepatan putaran dan kecepatan linearnya juga berbanding terbalik dengan sudut beloknya.

#### 4. KESIMPULAN

Berdasarkan perancangan dan pengujian kendali Robot Beroda berbasis image processing, maka dapat ditarik 3 buah kesimpulan. Pertama, kinerja pengolahan citra sebagai sistem “Self Driving Car” berjalan dengan baik. Dimana robot bisa mendeteksi rambu ‘STOP’, replika mobil dan lampu lalu lintas. Pengolahan citra berbasis Haar-Like Classifier dapat mencapai rata-rata akurasi sebesar 92,7% pada kecepatan rata-rata 0,52 m/s. Kedua, pengolahan citra dapat mendeteksi jalur yang telah ditentukan dengan baik. Dengan menggunakan sistem kendali berbasis Logika Fuzzy, robot dapat berjalan dengan rata-rata kecepatan 0,52 m/s dan akurasi sebesar 91.94%. Dan ketiga, pada saat robot beroda berbelok pada jalurnya, sudut maksimal yang mampu dilalui adalah 35 derajat dengan kecepatan maksimal saat berbelok adalah 0,24 m/s. Nilai maksimal sudut belok berbanding terbalik dengan kecepatan robot beroda. Dimana, semakin cepat robot beroda maka semakin kecil nilai maksimal sudut beloknya dan sebaliknya.

#### DAFTAR PUSTAKA

- [1] S. S. Transportasi, “Statistik Transportasi Darat 2019,” *BPS RI*, p. 73, 2019, [Online]. Available: <https://www.bps.go.id/>.
- [2] D. Howard, “Public Perceptions of Self-driving Cars: The Case of Berkeley, California,” *MS Transp. Eng.*, vol. 2014, no. 1, p. 21, 2014.
- [3] K. Massidik, E. Susanto, P. Pangaribuan, F. T. Elektro, U. Telkom, and N. Network, “Prototype Autonomous Car Menggunakan Image Processing dan Kontrol Neural Network,” *e-Proceeding Eng.*, vol. 4, no. 2, pp. 1490–1496, 2017.
- [4] Raspberry Pi Team, “Raspberry Pi 4,” *raspberrypi.org*, 2020. <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/>.
- [5] Raspberry Pi Team, “Camera Module,” *raspberrypi.org*, 2020.

- <https://www.raspberrypi.org/documentation/hardware/camera/>.
- [6] R. Rittenberry, "Hands-on technology: L298N Dual H-Bridge Motor Driver," *Occup Heal. Saf*, vol. 74, no. 2, p. 24, 2005.
  - [7] Motion Control Online Marketing Team, "Benefits of DC Motors and Why They're Great for Robotics," 2019. <https://www.motioncontrolonline.org/blog-article.cfm/Benefits-of-DC-Motors-and-Why-They-re-Great-for-Robotics/82>.
  - [8] A. Rusdinar *et al.*, "Automated Ultraviolet C Light Mobile Robot for Room Sterilization and Disinfection," *Int. J. Technol.*, vol. 12, no. 4, p. 854, 2021, doi: 10.14716/ijtech.v12i4.4817.
  - [9] P. Viola, "Rapid Object Detection using a Boosted Cascade of Simple Features," *2017 Int. Conf. Energy, Commun. Data Anal. Soft Comput. ICECDS 2017*, pp. 1193–1197, 2018, doi: 10.1109/ICECDS.2017.8389630.
  - [10] J. T. Elektro and F. Teknik, "Aplikasi Fuzzy Logic untuk Pengendali Motor DC Berbasis Mikrokontroler ATMega8535 dengan Sensor Photodiode," *J. Tek. Elektro*, vol. 7, no. 2, pp. 81–85, 2015, doi: 10.15294/jte.v7i2.8594.
  - [11] A. F. Setiaji, A. Rusdinar, R. Nugraha, F. T. Elektro, and U. Telkom, "KONTROL POSISI ROBOT BERODA DENGAN IMAGE PROCESSING POSITION," pp. 1–8.
  - [12] M. N. Fadhil and A. Rusdinar, "Kontrol Pergerakan Robot Beroda Menggunakan Pengendali Fuzzy-PID Berbasis Line Follower dan Image Processing Wheeled Robot Movement Control Using Fuzzy-PID Based on Line-Follower and Image Processing," vol. 6, no. 2, pp. 2750–2757, 2019.
  - [13] Y. Joko, H. Widodo, J. T. Elektro, F. Sains, D. A. N. Teknologi, and U. S. Dharma, "Tugas Akhir Charger Baterai Li-Po 3 Sel Menggunakan Flyback Konverter Dengan Masukan 220 Vac Final Project Charger Battery Li-Po 3 Cell Using Flyback Converter With 220 Vac Input," 2017.

**Halaman Ini Dikosongkan**