

## **PENERAPAN METODE FUZZY LOGIC SUGENO PADA PROTOTPE SISTEM KENDALI Pengereman DENGAN MENGGUNAKAN ARDUINO**

**Muhammad Ibnu Atha'illah<sup>1</sup>, Rizky Pradana<sup>2\*</sup>**

<sup>12</sup>Fakultas Teknologi Informasi, Teknik Informatika, Universitas Budi Luhur, Jakarta, Indonesia

Email: <sup>1</sup>athallah48@gmail.com, <sup>2</sup>rizky.pradana@budiluhur.ac.id

(\* : corresponding author)

**Abstrak**-Penyebab kecelakaan lalu lintas sangat bervariasi, mulai dari jalanan yang rusak, kendaraan yang tidak baik kondisinya dan lalainya pengemudi. Tingginya angka kecelakaan lalu lintas saat ini berkontribusi terhadap kemungkinan terjadinya (human error) yang menjadi salah satu faktor terjadinya kecelakaan bagi pengemudi di dalam mobil. Keamanan sistem kendaraan sangat penting untuk keselamatan pengemudi. Dengan perkembangan ilmu pengetahuan dan teknologi yang sudah ada, manusia mampu menciptakan sistem keamanan pada kendaraan. Perancangan prototype dengan adanya Sistem Kendali Pengereman bertujuan agar dapat meminimalisir terjadinya kecelakaan lalu lintas. Sistem kontrol pengereman menggunakan logika fuzzy untuk menentukan seberapa besar gaya yang digunakan untuk menghentikan mobil. Fuzzy logic control (FLC) adalah sistem yang membantu mengontrol suatu perangkat atau sistem. Hal ini sering digunakan dalam studi dan sangat berguna dalam sistem pengendalian. Pada logika fuzzy juga kontrol dapat sesuai dengan keinginan pembuat sistem kontrol. Dengan menggunakan metode FLC untuk sistem kendali pengereman secara dalam bentuk prototype dengan dua masukan yaitu nilai kecepatan dan nilai jarak, adapun keluarannya yaitu tingkat pengereman. Pengujian ini dilakukan dengan simulasi Fuzzy Logic Toolbox yang tersedia pada MATLAB. Hasilnya prototype dapat mengendalikan pengereman dengan rata – rata kesalahan 3.45 % jika dibandingkan dengan simulasi MATLAB.

**Kata Kunci:** sistem kendali pengereman, FLC, pengereman, MATLAB

### **APPLICATION OF SUGENO FUZZY LOGIC METHOD FOR PROTOTYPE SYSTEM BRAKING CONTROL USING ARDUINO**

**Abstract**-The causes of traffic accidents vary widely, ranging from damaged roads, vehicles that are not in good condition and negligent drivers. The current high number of traffic accidents contributes to the possibility of occurrence (human error) which is one of the factors in the occurrence of accidents for drivers in cars. The safety of the vehicle system is very important for the safety of the driver. With the development of existing science and technology, humans are able to create security systems on vehicles. The design of the prototype with the Braking Control System aims to minimize the occurrence of traffic accidents. The braking control system uses fuzzy logic to determine how much force is used to stop the car. Fuzzy logic control (FLC) is a system that helps control a device or system. It is often used in studies and is very useful in control systems. In fuzzy logic, the control can also be in accordance with the wishes of the control system maker. By using the FLC method for the braking control system in the form of a prototype with two inputs, namely the speed value and the distance value, the output is the braking level. This test is carried out by using the Fuzzy Logic Toolbox simulation available in MATLAB. The result is that the prototype can control braking with an average error of 3.45% when compared to the MATLAB simulation.

**Keywords:** braking control system, FLC, braking, MATLAB

## **1. PENDAHULUAN**

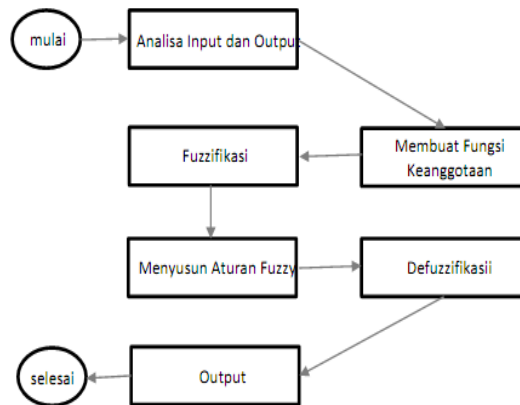
Kecelakaan lalu lintas adalah kecelakaan yang terjadi di jalan yang dapat mengakibatkan cedera atau kerusakan harta benda. Salah satu penyebab kecelakaan itu adalah pengemudi kurang hati-hati saat mengemudi, dan tiba-tiba ada benda di depannya, yang membuat pengemudi terlambat menginjak tuas rem untuk mengendalikan kecepatan kendaraan. mobil bergerak dengan kecepatan tinggi, sehingga sulit untuk mengelak. Sebelumnya sudah ada penelitian tentang sistem pengereman otomatis, namun hasil dari sistem pengereman tersebut masih belum cukup baik. Hal ini mengurangi kenyamanan pengemudi [1]. Untuk memberikan pengereman yang mulus, perlu ditambahkan rule pada logika fuzzy yang digunakan, dan juga agar mobil dapat memasuki area yang sempit. Pada pembuatan system ini akan diuraikan teori dasar dari komponen utama yang digunakannya. Kemudian akan membandingkan hasil dari simulasi dengan alat ini untuk melihat seberapa baik kinerjanya.

Pada pembuatan sistem ini akan diuraikan teori dasar dan beberapa komponen utama dari alat ini, juga penjelasan dari mulai perancangan masukan sampai dengan perbandingan hasil dengan simulasi MATLAB.

## 2. METODE PENELITIAN

### 2.1 Logika Fuzzy

Logika fuzzy adalah salah satu cara yang tepat untuk memetakan suatu ruang input ke dalam ruang output. Logika fuzzy meniru tentang cara berpikir manusia dengan menggunakan konsep ambiguitas nilai[2]. Logika fuzzy dibuat untuk mengurangi kekakuan logika kontrol biner dengan logika 1 dan logika 0[3]. Dalam logika Boolean logika berlaku antara 1 dan 0, dan logika fuzzy umumnya terdiri dari ambiguitas, fungsi keanggotaan, basis dan distorsi. Gambar 1 menunjukkan diagram Blok Diagram Fuzzy Logic Controller.



**Gambar 1.** Blok Diagram Fuzzy Logic Controller

Fungsi keanggotaan logika fuzzy digunakan untuk menghitung derajat keanggotaan suatu himpunan fuzzy. Fungsi ini dibangun berdasarkan persamaan garis yang dibentuk oleh himpunan fuzzy tersebut. Contoh fungsi dari himpunan segitiga adalah sebagai berikut :

$$f(x, a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \quad (1)$$

dimana  $f(x,a,b,c)$  adalah derajat keanggotaan, adalah nilai dari variabel, berturut-turut adalah nilai awal, tengah dan akhir dari variabel.

### 2.2 Fuzzy Logic Controller

Pertama, fuzzifikasi terjadi ketika instruksi program diterjemahkan ke dalam bentuk yang dapat dimengerti oleh FLC. Ini dapat melibatkan penambahan simbol ekstra, mengubah urutan instruksi, atau mengatur ulang blok kode. Fuzzifikasi adalah sebuah proses mengubah nilai suatu masukan menjadi suatu fungsi. Nilai fuzzy digunakan sebagai input untuk mekanisme inferensi. Pada titik ini, keputusan akan dibuat dari input yang ada berdasarkan rule base yang dirancang. Terakhir, nilai keluaran (*output*) dari mekanisme inferensi fuzzy diubah kembali menjadi bentuk tegas melalui proses defuzzifikasi.

Pada proses defuzzifikasi, terdapat beberapa metode yang sering digunakan oleh para peneliti. Salah satu metode yang akan dipakai dalam sistem ini adalah metode Takagi-Sugeno-Kang (TSK). Metode ini dipakai karena output dari metode ini bukan berupa himpunan fuzzy tetapi berupa konstanta atau persamaan linier. Persamaan dari metode TSK bisa dilihat sebagai berikut:

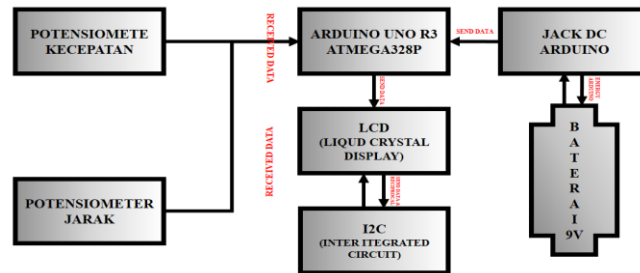
$$Z_{out} = \frac{\sum_{i=1}^n w_i z_i}{\sum_{i=1}^n w_i} \quad (2)$$

### 2.3 Perangkat Keras

Sistem kendali pengereman dapat dirancang dalam bentuk prototype. Dua unit potensiometer digunakan sebagai input kecepatan dan jarak. Mikrokontroler yang digunakan adalah board Arduino Uno ATMEGA 328p. Mikrokontroler adalah perangkat komputasi yang mencakup FLC. Pada bagian output, LCD menampilkan data input dan output.

### 2.4 Deployment Diagram

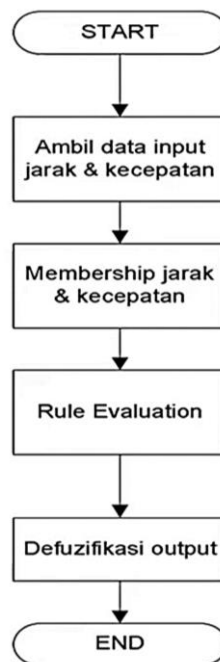
pada Gambar 2 dalah gambaran dari percobaan yang dibuat dalam bentuk deployment diagram.



Gambar 2. Deployment Diagram

### 2.5 Flowchart

Untuk memudahkan dalam membaca cara kerja dari sistem yang akan dibuat maka dibuatkan *flowchart* sebuah untuk penyelesaian masalah yang digambarkan menggunakan simbol-simbol. sebagai sarana memudahkan dalam mengetahui alur kerja dari beberapa rancangan alat yang dibuat sebelumnya. Flowchart ini terdiri dari kontroler Arduino UNO R3 dan penggunaan logika fuzzy.



Gambar 3. Flowchart

### 2.6 Algoritma

Berikut ini adalah program algoritma, dari cara kerja sistem pengereman berbasis fuzzy logic.

- a. Algoritma Kerja Alat Prototype Sistem Pengereman

```
float mf_kecepatan(float a, float b, float c) // Mencari Derajat keanggotaan
membership function segitiga kecepatan
{
    if (kecepatan>=a&&kecepatan<b) {m_kecepatan=(kecepatan-a)/(b-a); }
    if (kecepatan>=b&&kecepatan<c) {m_kecepatan=(c-kecepatan)/(c-b);}
}

float mf_jarak(float a1, float b1, float c1) // Mencari Derajat keanggotaan
membership function segitiga Jumlah jarak
{
    if (jarak>=a1&&jarak<b1) {m_jarak=(jarak-a1)/(b1-a1);}
    if (jarak>=b1&&jarak<c1) {m_jarak=(c1-jarak)/(c1-b1);}
}
```

**Gambar 4.** Program Algoritma dan Cara Kerja Sistem

- b. Algoritma Program Untuk Menghitung Derajat Keanggotaan dari Membership Function Segitiga Pada Kecepatan

```
if(kecepatan>=15&&kecepatan<=20)
    {mf_kecepatan(0,15,20); sangat_lambat =m_kecepatan;
    c_sedang=0; cepat=0; sangat_cepat=0; } // sangat_lambat

if(kecepatan>=15&&kecepatan<=25)
    {mf_kecepatan(15,20,25); lambat=m_kecepatan;
    cepat=0; sangat_cepat=0; } // lambat

if(kecepatan>=20&&kecepatan<=30)
    {mf_kecepatan(20,25,30); sangat_lambat =0;
    c_sedang=m_kecepatan; sangat_cepat=0; } // Sedang

if(kecepatan>=25&&kecepatan<=35)
    {mf_kecepatan(25,20,35); sangat_lambat =0;
    lambat=0; cepat=m_kecepatan; } // cepat

if(kecepatan>=30&&kecepatan<=35)
    {mf_kecepatan(30,35,0); sangat_lambat =0;
    lambat=0; c_sedang=0; sangat_cepat=m_kecepatan; } // sangat cepat
```

**Gambar 5.** Program Algoritma dari Membership Function Kecepatan

- c. Algoritma Program Untuk Menghitung Derajat Keanggotaan dari Membership Function Segitiga Pada Jarak

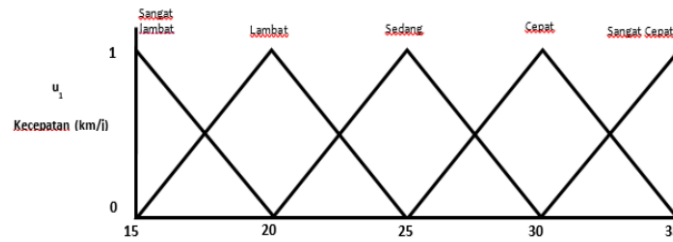
```
if(jarak>=0&&jarak<=15)
    {mf_jarak(0,0,15); sangat_dekat=m_jarak;
    j_sedang=0; jauh=0; sangat_jauh=0; } // Sangat dekat

if(jarak>=0&&jarak<=25)
    {mf_jarak(0,15,25); dekat=m_jarak;
    jauh=0; sangat_jauh=0; } // dekat
if(jarak>=15&&jarak<=35)
    {mf_jarak(15,25,35); sangat_dekat=0;
    j_sedang=m_jarak; sangat_jauh=0; } // sedang
if(jarak>=25&&jarak<=40)
    {mf_jarak(15,35,40); sangat_dekat=0;
    dekat=0; jauh=m_jarak; } // jauh
if(jarak>=35&&jarak<=40)
    {mf_jarak(35,40,0); sangat_dekat=0;
    dekat=0; j_sedang=0; sangat_jauh=m_jarak; } // sangat jauh
```

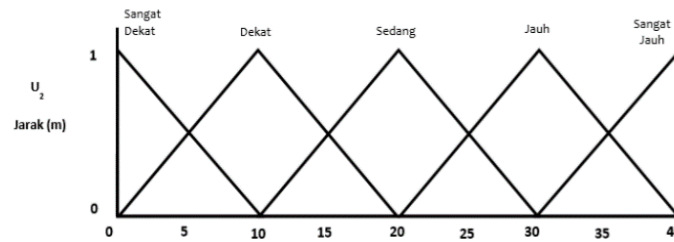
**Gambar 6.** Program Algoritma dari Membership Function Jarak

## 2.7 Desain FLC (*Fuzzy Logic Controller*)

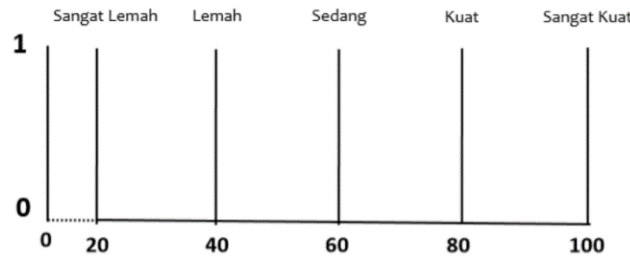
Fungsi keanggotaan input dan output dapat dilihat pada Gambar 7-9 di bawah ini.



**Gambar 7.** Fungsi Keanggotaan Kecepatan



**Gambar 8.** Fungsi Keanggotaan Jarak

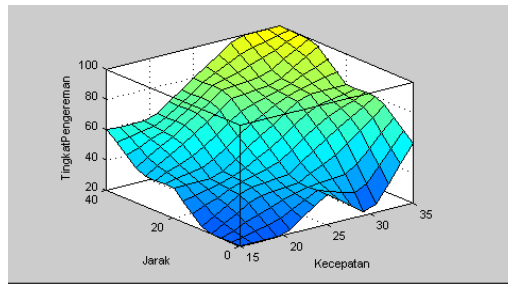


**Gambar 9.** Fungsi Keanggotaan Tingkat Pengereman

Fungsi keanggotaan kecepatan dan jarak memiliki lima fungsi, masing-masing dengan kategori kekuatan yang berbeda. Kecepatan yang terukur adalah 15-35 km/jam, sedangkan jarak pengereman 0-40 cm. Fungsi pengereman dibagi menjadi lima bagian, tetapi direpresentasikan sebagai nilai tunggal menggunakan metode TSK. Basis aturan dirancang berdasarkan data input dan output yang ditentukan, seperti yang terlihat pada Tabel 1. Fungsi basis aturan ditemukan menggunakan Fuzzy Logic Toolbox di MATLAB seperti pada Gambar 10.

**Tabel 1.** Tabel Basis Aturan (*Rule Base*)

Jarak Kecepatan	Sangat Lambat	Lambat	Sedang	Cepat	Sangat Cepat
Sangat Dekat	SL	SL	L	K	K
Dekat	SL	SL	S	K	SK
Sedang	SL	L	S	K	SK
Jauh	SL	L	S	SK	SK
Sangat Jauh	L	L	K	SK	SK



**Gambar 10.** Output Model Sugeno Pada Tingkat Pengereman

Jika disajikan dalam bentuk bahasa pemrograman, maka rule base pada tabel 1 akan terlihat seperti ini:

1. If (Kecepatan = sangat lambat && jarak = sangat\_dekat) {tingkat pengereman = SL}
2. If (Kecepatan = sangat lambat && jarak = dekat) {tingkat pengereman = SL}
3. If (Kecepatan = sangat lambat && jarak = sedang) {tingkat pengereman = SL}
4. If (Kecepatan = sangat lambat && jarak = jauh) {tingkat pengereman = SL}
5. If (Kecepatan = sangat lambat && jarak = sangat\_jauh) {tingkat pengereman = L}
6. If (Kecepatan = lambat && jarak = sangat\_dekat) {tingkat pengereman = SL}
7. If (Kecepatan = lambat && jarak = dekat) {tingkat pengereman = SL}
8. If (Kecepatan = lambat && jarak = sedang) {tingkat pengereman = L}
9. If (Kecepatan = lambat && jarak = jauh) {tingkat pengereman = L}
10. If (Kecepatan = lambat && jarak = sangat\_jauh) {tingkat pengereman = L}
11. If (Kecepatan = sedang && jarak = sangat\_dekat) {tingkat pengereman = L}
12. If (Kecepatan = sedang && jarak = dekat) {tingkat pengereman = S}
13. If (Kecepatan = sedang && jarak = sedang) {tingkat pengereman = S}
14. If (Kecepatan = sedang && jarak = Jauh) {tingkat pengereman = S}
15. If (Kecepatan = sedang && jarak = sangat jauh) {tingkat pengereman = K}
16. If (Kecepatan = cepat && jarak = sangat dekat) {tingkat pengereman = K}
17. If (Kecepatan = cepat && jarak = dekat) {tingkat pengereman = K}
18. If (Kecepatan = cepat && jarak = sedang) {tingkat pengereman = K}
19. If (Kecepatan = cepat && jarak = jauh) {tingkat pengereman = SK}
20. If (Kecepatan = cepat && jarak = sangatjauh) {tingkat pengereman = SK}
21. If (Kecepatan = sangatcepat && jarak = sangat dekat) {tingkat pengereman = K}
22. If (Kecepatan = sangatcepat && jarak = dekat) {tingkat pengereman = SK}
23. If (Kecepatan = sangatcepat && jarak = sedang) {tingkat pengereman = SK}
24. If (Kecepatan = sangatcepat && jarak = jauh) {tingkat pengereman = SK}
25. If (Kecepatan = sangatcepat && jarak = sangat jauh) {tingkat pengereman = SK}

### 3. HASIL DAN PEMBAHASAN

Pada bagian ini akan membahas mengenai implementasi serta evaluasi dari sebuah alat prototype Sistem Kendali Pengereman untuk menghindari tabrakan dari kecepatan dan menghentikan suatu kendaraan , adapun pembahasannya sebagai berikut.

#### 3.1 Implementasi Metode

Pada pembuatan prototype sistem kendali pengereman ini, penulis menggunakan metode prototyping. Dimana metode prototyping ini memiliki beberapa tahapan untuk menggunakan metode ini supaya dapat berjalan dengan sesuai yang diinginkan seperti pada tahap perancangan sebagai tolak ukurnya.

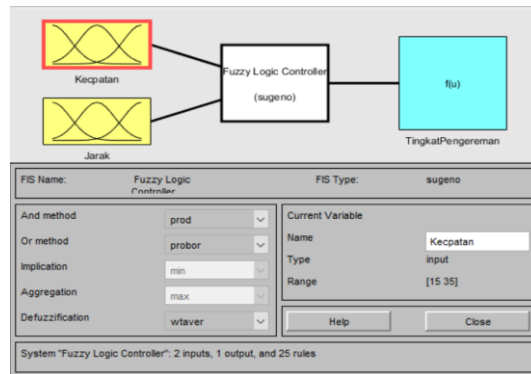
Pada tahap ini dilakukan perancangan simulasi sistem kendali pengereman yang akan diterapkan pada prototype dalam pengujian mode pengereman.

- a. Sistem akan berfungsi untuk menentukan mode pengereman berdasarkan logika fuzzy yang telah ditanamkan pada mikrokontroler
- b. Potensiometer Jarak diposisikan pada bagian samping kendaraan
- c. Potensiometer Kecepatan diposisikan pada bagian samping kendaraan

- d. Sistem memberikan informasi terkait tingkat pengereman, jarak dalam satuan cm, dan kecepatan dalam satuan km/h

### 3.2 Simulasi

Simulasi digunakan untuk memprediksi kinerja sistem yang dirancang sebelum benar-benar diuji pada perangkat keras. Hasil simulasi dapat digunakan sebagai acuan dan pembandingan terhadap hasil pengujian perangkat keras. Fuzzy Logic Toolbox di MATLAB dapat digunakan untuk simulasi. Tampilan dari simulasi yang dilihat pada Gambar 11, pada perangkat tersebut dapat digunakan untuk merancang fungsi keanggotaan input dan output yang sesuai dengan metode defuzzifikasi.



Gambar 11. Fuzzy Logic Toolbox

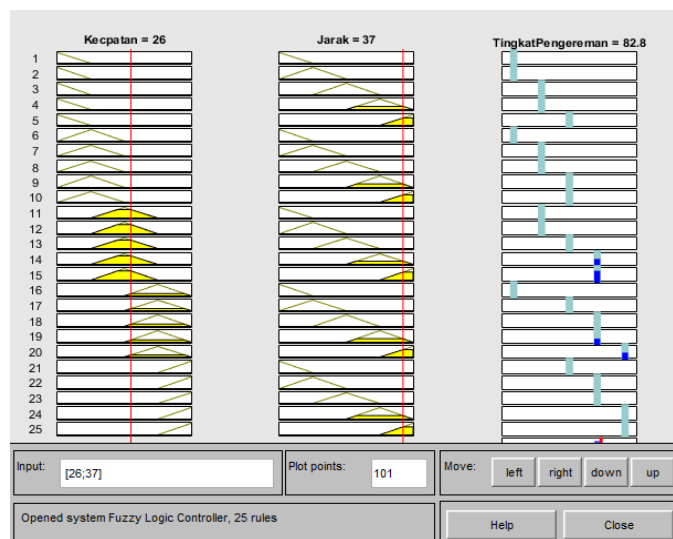
### 3.3 Realisasi Rancangan Prototype

Realisasi dari prototype yang dirancang dapat dilihat pada Gambar 15. Pada Gambar 15 masukan kecepatan dan masukan jarak menggunakan dua unit potensiometer. kecepatan dan jarak diasumsikan dengan putaran dari potensiometer.

### 3.4 Pengujian

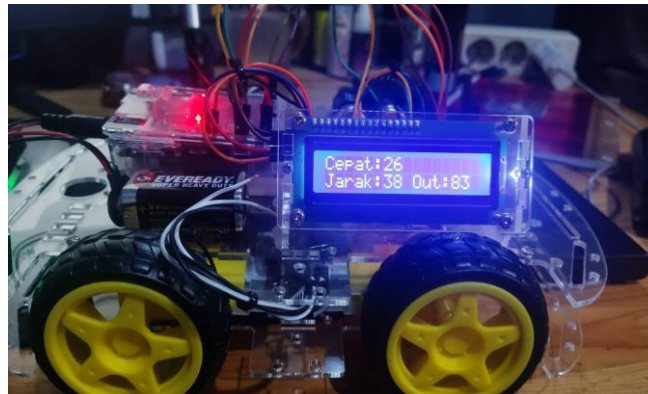
Salah satu pengujian dilakukan setelah proses dari perancangan sistem telah berhasil. Pengujian ini berguna untuk mengetahui akurasi sebuah instrumen yang digunakan pada penelitian ini, Prototipe yang digunakan akan diuji dan di bandingkan hasilnya dengan simulasi pada MATLAB.

- a. Pengujian Logika Fuzzy Pengereman Pada Saat Kecepatan dan Jarak menggunakan MATLAB



Gambar 12. Pengujian dengan simulasi MATLAB

b. Pengujian Logika Fuzzy Pengereman Pada Saat Kecepatan dan Jarak menggunakan Prototype



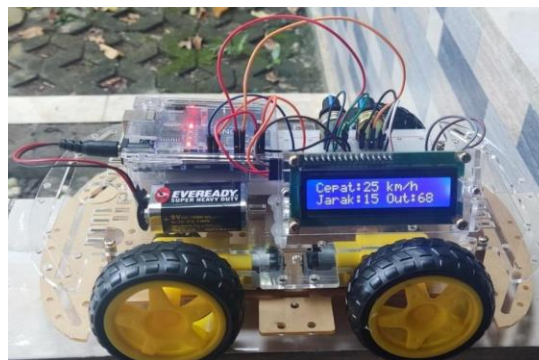
**Gambar 13.** Pengujian dengan Simulasi MATLAB

### 3.5 Data Hasil Pengujian

Prototype tersebut kemudian diuji dan di bandingkan hasilnya dengan simulasi pada software MATLAB. Hasil pengujian dapat dilihat pada Tabel 2 di bawah ini.

**Tabel 2.** Hasil Pengujian Menggunakan Simulasi Matlab dan Prototype

Pengujian Ke	Masukan		Keluaran (Tingkat Pengereman)		Error (%)
	Kecepatan	Jarak	Simulasi	Prototype	
1	19	5	28	20	0.8
2	21	18	43.2	34	1.4
3	26	37	82.8	81	0.3
4	20	30	60	43	1.7
5	16	26	42	27	2.1
6	17	34	48	28	0.6
7	34	12	76.8	96	1.3
8	19	40	60	38	2.8
9	34	40	100	100	0
10	22	8	37.6	38	0.2
11	20	15	40	26	5
12	30	10	60	81	2.2
13	18	26	47.2	34	1.9
14	25	5	40	60	6.7
15	23	37	72	64	0.8
Rata - rata					2.5



**Gambar 14.** Realisasi Rancangan Dalam Bentuk Prototype

Dari lima belas data pengujian pada Tabel 2, terlihat bahwa prototipe mampu mengendalikan pengereman dengan nilai yang tidak jauh berbeda dengan nilai simulasi MATLAB. Hal ini dikarenakan sensor yang digunakan



pada prototype diasumsikan menggunakan potensiometer sehingga menghasilkan data yang lebih akurat. Meskipun sebenarnya akan lebih bagus jika sensor nyata digunakan dan dengan meningkatkan karakteristik sensor, serta pemrosesan sinyal yang baik, besarnya nilai kesalahan akan berkurang diminimalkan untuk tujuan implementasi dunia nyata.

#### 4. KESIMPULAN

Sistem FLC (Fuzzy Inference Control) prototype ini berhasil mengimplementasikan hasil yang diinginkan. Yaitu menggunakan sensor yang diasumsikan dengan potensiometer, Hasil pengujian menunjukkan bahwa Simulasi MATLAB dan prototype yang telah dirancang dapat mengendalikan pengereman dengan rata - rata kesalahan 2.5%, dan Sistem kendali pengereman ini dapat bekerja ketika prototype dikendalikan dengan kecepatan dan jarak, dengan rata-rata tingkat keberhasilan melakukan pengereman mencapai 95%.

#### DAFTAR PUSTAKA

- [1] E. W. Saragih, M. R. Lubis, A. Wanto, Solikhun and Jalaluddin, "Rancang Bangun Sistem Rem Otomatis pada Kendaraan Menggunakan Sensor Ultrasonik," *Jurnal Penelitian Inovatif*, vol. 1, no. 2, pp. 85-94, 2021.
- [2] B. W. Nugraha, A. Wibowo and A. Surya, "Sistem Pengereman Berbasis Fuzzy Logic Pada Sepeda Motor Listrik (Braking System Using Fuzzy Logic on Electric Motorcycle)," *e-Proceeding of Engineering*, vol. 6, no. 3, pp. 10146-10154, 2019.
- [3] F. Jannah, Sudiyono and T. A. Setiawan, "Perancangan Dan Analisis Sistem Pengereman Hydraulic Pada Mobil Minimalis Roda Tiga," *Proceedings Conference on Design and Manufacture and Its Application*, Politeknik Perkapalan Negeri Surabaya, pp. 94-104, Januari 2018.
- [4] M. Sugianto and H. Amri, "Sistem Keselamatan Sepeda Motor Untuk Menghindari Kecelakaan Berbasis Mikrokontroler," *Jurnal INOVTEK - Seri Elektro*, vol. 1, no. 28, pp. 28-35, 2019.
- [5] S. Alam and G. A. Maulana, "Rancang Bangun Sistem Pengereman Otomatis Menggunakan Arduino Uno Dan Sensor Ultrasonik," *JTT (Jurnal Teknologi Terapan)*, vol. 1, no. 1, pp. 69-75, 2020.
- [6] I. G. I. Perdana, H. Wibowo and A. Ridwan, "Pengukuran kecepatan kendaraan berbasis mikrokontroler guna menunjang keselamatan dalam berkendara," *Jurnal Penelitian Politeknik Penerbangan Surabaya*, vol. 4, no. 4, pp. 237-246, 2021.
- [7] A. Tarwanto and V. Arinal, "Implementasi dan Monitoring Sistem Keamanan Kendaraan Berbasis IoT pada Bengkel Cahaya," *Jurnal Sosial dan Teknologi (SOSTECH)*, vol. 1, no. 8, pp. 887-892, 2021.
- [8] A. Putra and D. Romahadi, "Sistem Keamanan Sepeda Motor Berbasis Iot (Internet of Things) Dengan Smartphone Menggunakan Nodemcu," *JTT: Jurnal Teknologi Terpadu*, vol. 9, no. 1, pp. 77-87, 2021.
- [9] D. Prameswari and Yohanes, "Analisa Sistem Pengereman Pada Mobil Multiguna Pedesaan," *Jurnal Teknik ITS*, vol. 8, no. 1, pp. 67-73, 2019.
- [10] A. I. Agung, W. Aribowo and A. C. Hermawan, "Analisis Penerapan Mikrokontroler Pada Pengereman Dinamik Motor Induksi Tiga Fasa," *Jurnal Teknik Elektro*, vol. 10, no. 1, pp. 99-108, 2021.