



Penerapan Fuzzy Logic Untuk Pendeteksian Jarak Aman Berkendara Sepeda

Laporan Tugas Akhir

**Oleh:
Al Firman Syah (4212331013)**

**Program Studi Teknik Mekatronika
Jurusan Teknik Elektro
Politeknik Negeri Batam
2025**

Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul: “Penerapan *Fuzzy Logic* Untuk Pendeteksian Jarak Aman Berkendara Sepeda” adalah hasil karya sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri. Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Batam, 06 Juni 2025



Al Firman Syah
NIM: 4212331012

Lembar Pengesahan

Laporan Tugas Akhir disusun untuk digunakan sebagai rencana kerja pada pelaksanaan Tugas Akhir

Disusun oleh:
Al Firman Syah (4212331013)

Tanggal Sidang:14, Juli, 2025

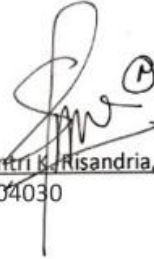
Disetujui oleh:

Penguji I



M.Naufal Aidangga, S.Pd., M.P.H
NIK:122281

Pembimbing



Sumarni K. Risandria, S.T., M.T
NIK:104030

Penguji II



Fadli Firdaus, M.Pd
NIK:122271

[Penerapan Fuzzy Logic Untuk Pendeteksian Jarak Aman Berkendara Sepeda]

Abstrak

Aktifitas bersepeda belakangan ini banyak diminati masyarakat Indonesia, selain digunakan sebagai transportasi alternatif yang ramah lingkungan, banyak orang memilih bersepeda karena nilai kesehatan yang ditawarkan. Di lingkungan perkotaan seperti Batam, bersepeda dipilih karena efisiensinya untuk mobilitas jarak dekat. Namun kepadatan lalu lintas, interaksinya dengan kendaraan lain, serta keberadaan pejalan kaki dan rambu lalu lintas, kerap membuat pengendara sepeda mengalami kecelakaan. Hal ini menjadi dorongan tersendiri bagi penulis untuk membuat sebuah alat yang akan membantu pengendara sepeda dalam menjaga jarak aman saat berkendara. Alat ini akan menggunakan Mikrokontroler ESP32 sebagai basis pemrograman yang mana didalamnya dimasukkan juga metode *fuzzy logic controller* sebagai salah satu sistem pengontrol alat. Kemudian penulis juga mengintegrasikan alat ini dengan beberapa komponen seperti *buzzer* yang akan berbunyi seperti sebuah alarm apabila jarak dan kecepatan sepeda pengendara memenuhi beberapa syarat, kemudian menyalakan lampu *LED* hijau, kuning dan merah serta pesan yang akan ditampilkan di *LCD* seperti keadaan aman, waspada, dan bahaya. Untuk mendeteksi jarak digunakan sensor *Tf-Luna LiDAR* dan kecepatan dideteksi dengan sensor *Magnetic Hall Effect*. Dari pengujian yang dilakukan didapati sensor *Tf- Luna LiDAR* memiliki rata-rata *error* 3,94 % dengan akurasi 96,06 %, sedangkan sensor *Magnetic Hall Effect* nilai rata-rata *error* adalah 6,50% % dengan akurasi 93,50 %. Dalam pengujiannya dengan beberapa orang sebagai pengguna hasil menunjukkan sistem ini efektif dalam membantu meningkatkan kewaspadaan saat berkendara sepeda terlebih dalam kondisi lalu lintas yang padat.

Kata kunci: ESP32, jarak aman, *fuzzy logic*.

[Application of Fuzzy Logic for Detecting Safe Distance for Riding a Bicycle]

Abstract

Cycling has recently become increasingly popular among people in Indonesia. Besides being an eco-friendly alternative mode of transportation, many people choose cycling for the health benefits it offers. In urban environments such as Batam, cycling is favored for its efficiency in covering short distances. However, traffic congestion, interaction with other vehicles, the presence of pedestrians, and traffic signs often lead to accidents involving cyclists. This situation has motivated the development of a device designed to help cyclists maintain a safe following distance while riding. The system utilizes an ESP32 microcontroller as its core platform, integrated with a fuzzy logic controller to manage decision-making processes. The devices include several components such as a buzzer that activated an alarm when specific speed and distance conditions are met, LED indicators (green, yellow, and red) to visually display safety levels, and an LCD screen that presents messages such as 'Safe', 'Caution', or 'Danger'. To detect a distance, the system uses a TF-Luna LiDAR sensor, while speed is measured using a magnetic Hall Effect sensor. Based on testing, the TF-Luna LiDAR sensor demonstrated an average error rate of 3.94%, corresponding to an accuracy of 96.6%. The Hall Effect sensor showed an average error of 6.50%, with an accuracy of 93.50%. Trials involving multiple users revealed that the system is effective in enhancing cyclist awareness, especially in high traffic conditions.

Keywords: ESP32, safe distance, fuzzy logic

Daftar Isi

Lembar Pengesahan	ii
Abstrak	iii
<i>Abstract</i>	iv
Daftar Isi	v
Daftar Gambar	vii
Daftar Tabel	viii
Bab 1. Pendahuluan	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	2
1.4. Manfaat	2
1.5. Batasan	2
Bab 2. Tinjauan Pustaka	3
2.1. Keamanan Berkendara	3
2.1.1. Faktor Jalan	3
2.1.2. Faktor Lingkungan	3
2.1.3. Faktor Kendaraan	3
2.1.4. Faktor Pengendara Atau Pengguna Jalan	3
2.1.5. Faktor Kecepatan	4
2.2. <i>Fuzzy Logic Controller</i>	5
2.2.1. Himpunan <i>Fuzzy</i>	5
2.2.2. Fungsi Keanggotaan	6
2.2.3. <i>Fuzzyfication</i>	11
2.2.4. <i>Rule Evaluation</i>	11
2.2.5. <i>Defuzzyfication</i>	12
2.3. Komponen Elektronika	15
2.3.1. Sensor Tf Luna Lidar	15

2.3.2. Hall Effect Sensor	16
Bab 3. Metodologi Penelitian	24
3.1. Perancangan	24
3.1.1. Perancangan Penelitian	24
3.1.2. Perancangan Sistem Kerja	25
3.1.3. Perancangan Mekanik	26
3.1.4. Perancangan Elektrikal	28
3.1.5. Perancangan <i>Fuzzy</i>	30
3.2. Alat dan Bahan	39
Bab 4. Pembahasan	40
4.1. Penentuan indikator jarak dan kecepatan sebagai <i>membership input</i> untuk <i>system fuzzy</i>	40
4.2. Analisa dan Pembahasan Sistem Fuzzy	43
4.3. Pengujian Sensor LiDAR	47
4.4. Pengujian Sensor Hall	49
4.5. Pengujian Alat Dengan Beberapa Surveyor	53
Bab 5. Kesimpulan dan Saran	56
5.1. Kesimpulan	56
5.2. Saran	57
Biodata	60
Lampiran A. Mikrokontroler program	61

Daftar Gambar

Gambar 1. Perbandingan Fungsi Keanggotaan Himpunan Fuzzy Terhadap Himpunan Crisp	5
Gambar 2. Representasi Linear Naik	6
Gambar 3. Representasi Linear Turun	7
Gambar 4. Representasi Kurva Segitiga	7
Gambar 5. Representasi Kurva Trapesium	8
Gambar 6. Representasi Kurva Bentuk Bahu	9
Gambar 7. Representasi Kurva -S pertumbuhan	10
Gambar 8. Representasi Kurva -S penyusutan	10
Gambar 9. Tf-Luna-Lidar Range Sensor	15
Gambar 10. Sensor Magnetic Effect Hall	17
Gambar 11. Diagram alir penelitian	24
Gambar 12. Blok Diagram sistem kerja alat	25
Gambar 13. Main Case Design	26
Gambar 14. Lidar Case Design	26
Gambar 15. Main Case to Bicycle handlebar Design	27
Gambar 16. Perangkat Terpasang	27
Gambar 17. Rangkaian elektrikl.....	28
Gambar 18. Kurva <i>Membership Jarak</i>	30
Gambar 19. Kurva <i>Membership Kecepatan</i>	33
Gambar 20. Grafik Singleton Output Sistem <i>Fuzzy</i>	37
Gambar 21. Pengujian Kesesuaian output terhadap sembilan rule evaluation yang ada.....	46
Gambar 22. Pengujian Sensor LIDAR	48
Gambar 23. Pengujian Jarak Pasang Sensor <i>Hall Effect</i>	50
Gambar 24. Pengujian Kecepatan	52
Gambar 25. Pengujian dengan Surveyor	55

Daftar Tabel

Tabel 1. Peraturan Menteri Perhubungan nomor PM 111 Tahun 2015 atas Penetapan Batas Kecepatan.....	4
Tabel 2. Tabel Rekomendasi Jarak Aman Berkendara Dari Panduan dan sosialisasi Kemenhub	4
Tabel 3. Spesifikasi Sensor Tf Luna LiDAR.....	16
Tabel 4. Spesifikasi Sensor <i>Hall Effect</i>	17
Tabel 5. Beberapa Literatur Penelitian Terdahulu yang dijadikan Acuan.....	19
Tabel 6. Derajat Keanggotaan <i>Membership</i> Jarak	32
Tabel 7. Derajat Keanggotaan <i>Membership</i> Kecepatan	35
Tabel 8. Rule Evaluation Beserta Hasil Hitung Kenggotaan Kecepatan 19,7 Km/h dan Jarak 400cm.....	36
Tabel 9. Level Peringatan Sistem <i>Fuzzy</i>	37
Tabel 10. Estimasi biaya	39
Tabel 11. Nilai <i>Membership input</i> untuk kecepatan dan jarak dalam sistem <i>fuzzy</i>	42
Tabel 12. <i>Rule Evaluation</i>	43
Tabel 13. Hasil pengujian kesesuaian output alarm terhadap sembilan rule evaluation yang ada	45
Tabel 14. Pengujian Sensor LiDAR	47
Tabel 15. Pengujian Sensor Hall	49
Tabel 16. Pengujian Kecepatan dengan Sensor Hall.....	52
Tabel 17. Pengujian dengan Beberapa surveyor	53

Bab 1. Pendahuluan

1.1. Latar Belakang

Aktifitas bersepeda belakangan ini banyak diminati masyarakat Indonesia, selain digunakan sebagai transportasi alternatif yang ramah lingkungan, banyak orang memilih bersepeda karena nilai kesehatan yang ditawarkan. [1] Di lingkungan perkotaan seperti Batam, bersepeda dipilih karena efisiensinya untuk mobilitas jarak dekat. Namun kepadatan lalu lintas, interaksinya dengan kendaraan lain, serta keberadaan pejalan kaki dan rambu lalu lintas, kerap membuat pengendara sepeda mengalami kecelakaan. Di Batam sendiri yang terkenal sebagai kota industri memiliki kepadatan lalu lintas yang luar biasa dan di bulan Desember tahun 2023 saja kecelakaan lalu lintas yang terjadi di kota Batam berdasarkan data dari Satlantas Polda Kepri terjadi sebanyak 75 kali, angka yang sangat besar untuk sebuah musibah.[2] Kondisi ini semakin kompleks mengingat pengguna sepeda yang berasal dari berbagai kalangan usia dan tingkat kewaspadaan yang berbeda. Hal ini membuat penulis tergerak untuk membuat sebuah alat yang mampu mendeteksi jarak aman dalam berkendara terkhusus dalam berkendara sepeda.[3]

Sebenarnya alat untuk mendeteksi jarak aman sudah pernah diciptakan melalui sebuah penelitian yang dilakukan oleh Asri Mulyani pada tahun 2018 dengan judul "Perancangan Sensor Jarak Aman Kendaraan Menggunakan Mikrokontroler NodeMCU ESP32". Namun dalam publikasi yang dilakukan penulis melihat kurang terstrukturanya perancangan perangkat lunak ataupun pemrograman yang dibuat dan menurut penulis pendekatan yang berupa pendeteksian jarak ini sangat bagus bila diterapkan dalam konsep kontrol dengan *fuzzy logic*. Untuk penelitian dengan mengimplementasikan *fuzzy logic* pada alat pendeteksi jarak aman berkendara sendiri juga sudah pernah di buat oleh orang-orang salah satunya oleh Maya Julianti pada tahun 2019 akan tetapi dalam penelitian yang telah dilakukan penulis melihat bahwa input untuk *fuzzy logic* hanya berupa pendeteksian jarak, sedangkan dalam berkendara kita mendefinisikan jarak aman secara berbeda. Jarak dua meter bisa saja aman untuk kondisi berkendara dengan kecepatan 20 Km/jam akan tetapi berbahaya bagi kecepatan 120 Km/jam. Dengan demikian penulis ingin menambahkan input untuk *fuzzy logic* yang berupa sensor kecepatan.

Dengan penambahan input berupa kecepatan pada pengkondisian *fuzzy logic* ini penulis yakin pengkondisian dalam pendeteksian jarak aman dalam berkendara menjadi lebih tepat, terstruktur dan efektif. Sebagai mikrokontroler pada alat ini penulis akan menggunakan Node MCU ESP32 yang terintegrasi dengan Sensor TF-Luna-LiDAR, sensor kecepatan dan beberapa komponen pendukung lainnya. Beberapa produk yang hampir mirip dengan alat ini adalah alarm pada mobil, akan

tetapi alat ini tidaklah bekerja dengan kondisi yang sama pada mobil. Alat ini akan bekerja saat dikendarai berbeda dengan mobil yang bekerja saat di parkir.

1.2. Rumusan Masalah

1. Bagaimana membuat pengendara sepeda mengetahui jarak berkendaranya.
2. Bagaimana meminimalisir terjadinya jarak bahaya saat bersepeda.
3. Bagaimanakah cara kerja dari sebuah alat yang mendeteksi jarak aman berkendara sepeda yang terimplementasikan *fuzzy logic* berbasis mikrokontroler.

1.3. Tujuan

1. Merancang suatu sistem yang mampu mendeteksi jarak aman dalam berkendara sepeda yang terimplementasikan *fuzzy logic controller*.
2. Menganalisa pengaruh dari alat yang dibuat terhadap respon pengendara sepeda pada jarak berkendaranya.
3. Menganalisa cara kerja dari alat yang mendeteksi jarak aman berkendara sepeda yang terimplementasikan dengan *fuzzy logic* dan berbasis mikrokontroler.

1.4. Manfaat

1. Membantu pengendara sepeda dalam mengetahui jarak nya dengan kendaraan yang ada di depan.
2. Dengan adanya peringatan dari alat akan meminimalisir jumlah kecelakaan lalu lintas yang disebabkan oleh ketidakpekaan pengendara sepeda terhadap jarak.
3. Membuat pengendara sepeda menjadi lebih perhatian dalam menjaga jarak aman saat berkendara.

1.5. Batasan

Berikut batasan masalah yang dibuat agar permasalahan pada tugas akhir ini tidak bertambah.

1. Jarak maksimal untuk pendeteksian alat ini adalah delapan meter.
2. Setelah dilakukan perhitungan, kecepatan maksimal paling aman untuk jarak berkendara delapan meter adalah $9,6 \text{ km/h}$, maka proses pembagian indikator kecepatan dan rule evaluation akan disesuaikan dengan nilai kecepatan ini.
3. Alat ini hanya akan memberikan peringatan berupa alarm dan nyala lampu led dalam berbagai warna serta menampilkan keterangan jarak dalam LCD.
4. Pengujian alat ini hanya akan dilakukan pada sepeda.
5. Alat ini hanya akan mendeteksi kondisi jarak berkendara terhadap kendaraan lain yang ada di depan.

Bab 2. Tinjauan Pustaka

2.1. Keamanan Berkendara

Keamanan ataupun keselamatan dalam berkendara adalah upaya untuk menekan angka kecelakaan berkendara. Kecelakaan lalu lintas ini bisa terjadi kapan saja bahkan pada saat yang tidak diduga. Secara umum faktor penyebab kecelakaan lalu lintas ada empat, yakni:

2.1.1. Faktor Jalan

Faktor jalan sebagai penyebab kecelakaan lalu lintas dapat berupa *geometry* jalanan yang tidak sempurna, kerusakan atau kurangnya peralatan dan fasilitas jalan seperti *traffic light* dan rambu lalu lintas yang memberi peringatan akan kondisi jalan.[4] Akan tetapi di Indonesia faktor jalan yang paling berpotensi mengakibatkan kecelakaan adalah kerusakan jalan.

2.1.2. Faktor Lingkungan

Lingkungan yang menjadi penyebab terjadinya kecelakaan berlalu lintas adalah kondisi cuaca yang buruk seperti: hujan dan badai, dua kondisi ini sering menyebabkan kecelakaan berkendara, hujan menyebabkan jalanan licin sedangkan badai atau angin kencang seringkali mengganggu pengendara.[5] Saat hujan juga kerap menimbulkan sambaran petir yang mana bias mengagetkan pengendara dan berujung kecelakaan juga.

2.1.3. Faktor Kendaraan

Kendaraan kerap juga menjadi penyebab terjadinya kecelakaan lalu lintas biasanya faktor penyebab kecelakaan pada kendaraan berupa kondisi teknis yang tidak tepat seperti: kampas rem yang sudah habis, ini tentu saja dapat menyebabkan kecelakaan lalu lintas karena saat pengendara melakukan pengereman pada kendaraan yang kampas rem nya sudah habis tentu tidak akan terjadi perlambatan ataupun terlambatnya perlambatan tersebut terjadi sehingga bisa saja pengendara telah menabrak terlebih dahulu.[5]

2.1.4. Faktor Pengendara Atau Pengguna Jalan

Faktor pengendara yang menjadi sebab terjadinya kecelakaan lalu lintas dapat berupa kondisi fisik pengendara, keterampilan, kedisiplinan berlalu lintas dan pejalan kaki. Misalnya kedisiplinan pengendara yang mengakibatkan kecelakaan adalah pengendara yang ugal-ugalan dan tidak menjaga jarak aman dalam berkendara. Pengendara hendaknya selalu menjaga jarak aman dalam berkendara antara dia dengan kendaraan yang ada di depan dan belakang agar tidak terjadi

benturan apabila kendaraan yang didepan tiba-tiba melakukan pengereman mendadak ataupun benturan karena kendaraan di belakang yang ingin melewati kendaraan kita.[5]

2.1.5. Faktor Kecepatan

Selain empat faktor diatas faktor kecepatan juga merupakan pertimbangan penting untuk keamanan dan menurut Kementerian Perhubungan Republik Indonesia dalam Undang-Undang Lalu Lintas Angkutan Jalan, mengeluarkan Peraturan Menteri Perhubungan nomor PM 111 Tahun 2015 tentang tata cara penetapan batas kecepatan. Penetapan ini ditetapkan secara nasional dan dinyatakan dengan rambu lalu lintas.

Tabel 1. Peraturan Menteri Perhubungan nomor PM 111 Tahun 2015 atas Penetapan Batas Kecepatan

Tipe Kecepatan	Nilai	Keterangan
Rendah	60 Km/jam	Jalan Bebas hambatan
Tinggi	100 Km/jam	Jalan Bebas Hambatan
Tinggi	80 Km/jam	Jalan antar kota
Tinggi	50 Km/jam	Kawasan Perkotaan
Tinggi	30 Km/jam	Kawasan Permukiman

Batas kecepatan ini disamaratakan untuk semua jenis kendaraan sedangkan untuk sepeda di Indonesia belum ada ketetapan dan aturan khusus yang mengatur itu. Selain aturan ini ada juga sosialisasi terbaru dari Kemenhub yang dapat dilihat pada tabel dibawah.

Tabel 2. Tabel Rekomendasi Jarak Aman Berkendara Dari Panduan dan sosialisasi Kemenhub

Kecepatan	Jarak Minimal	Jarak Aman
30 Km/h	15 Meter	30 Meter
40 Km/h	20 Meter	40 Meter
50 Km/h	25 Meter	50 Meter
60 Km/h	40 Meter	60 Meter
70 Km/h	50 Meter	70 Meter
80 Km/h	60 Meter	80 Meter
90 Km/h	70 Meter	90 Meter
100 Km/h	80 Meter	100 Meter

2.2. Fuzzy Logic Controller

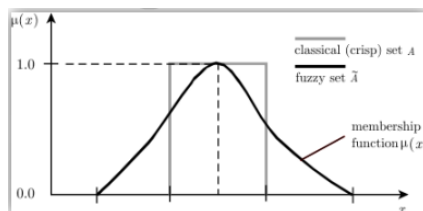
Logika *Fuzzy* adalah salah satu cabang ilmu dalam bidang *soft computing*. Logika *fuzzy* ini diperkenalkan pertama kali oleh Prof. Lotfi A. Zadeh seorang pria berkebangsaan Iran yang juga merupakan guru besar di *University of California di Berkeley* pada tahun 1965. Teori yang ia kembangkan ini mengenai logika kabur yaitu konsep nilai yang berada di antara kebenaran (*true*) dan kesalahan (*false*) yang merupakan usaha untuk menggabungkan teori matematika dengan intuisi manusia.

Logika *fuzzy* berbeda dengan logika digital biasa atau *boolean*. Logika *Boolean* hanya mengenal dua keadaan seperti ya atau tidak, 0 dan 1, ON dan OFF. Namun untuk logika *fuzzy* ia mengenal seluruh keadaan berdasarkan pengkondisian yang dilakukan pembuatnya baik itu 1 sampai 10 ataupun diantara 1 dan 10 *fuzzy* mengenal nilai benar, nilai salah juga nilai samar samar. Contoh penggunaan *fuzzy* misalnya, pada pendeteksian suhu, pendeteksi cuaca, dan masih banyak lagi.[6]

2.2.1. Himpunan Fuzzy

Himpunan *fuzzy* merupakan suatu kelompok objek yang memiliki nilai keanggotaan. Setiap anggota memiliki derajat keanggotaan tertentu yang ditentukan oleh fungsi anggota (*membership function*) tertentu pula atau biasa disebut dengan fungsi karakteristik (*characteristic function*).

Selain itu dikenal juga istilah himpunan *crisp* yang mana ia membedakan anggotanya dengan nilai nol atau satu, anggota himpunan atau bukan.[7]



Gambar 1. Perbandingan Fungsi Keanggotaan Himpunan Fuzzy Terhadap Himpunan Crisp

Nilai keanggotaan item X pada suatu himpunan A sering didefinisikan dengan $\mu_A[X]$, dimana X memiliki kemungkinan merupakan interval bilangan real dari 0 sampai dengan satu (0, 1). Dalam system fuzzy selain himpunan *fuzzy* ada

beberapa hal umum yang perlu diketahui seperti variabel *fuzzy*, semesta pembicaraan, dan domain.

2.2.2. Fungsi Keanggotaan

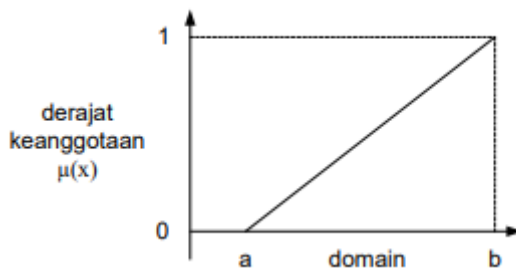
Fungsi keanggotaan adalah sebuah kurva yang memetakan titik-titik input data kedalam nilai keanggotaannya. Kurva ini memiliki interval antara 0 - 1. Untuk mendapatkan nilai keanggotaan salah satu cara yang bisa digunakan adalah dengan memulai pendekatan fungsi. Beberapa fungsi yang bisa digunakan dalam memetakan titik input data ke dalam nilai keanggotaan yang ada adalah sebagai berikut:

1. Representasi Linear

Berdasarkan namanya "linear" pemetaan input kederajatan keanggotaan representasi linear ini digambarkan sebuah garis lurus. Ada dua himpunan *fuzzy* dalam representasi linear, yakni:

a. Kenaikan himpunan

Kenaikan himpunan, dimulai dari nilai yang derajat keanggotaannya nol (0) bergerak ke kanan menuju nilai yang derajat keanggotaannya lebih tinggi.



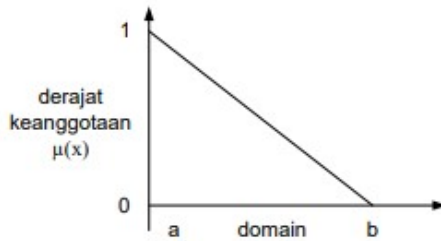
Gambar 2. Representasi Linear Naik

Fungsi Keanggotaan:

$$\mu [x, a, b] = \begin{cases} 0; & x \leq a \\ (x - a)/(b - a); & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (1)$$

b. Penurunan Himpunan

Dimulai dari nilai yang derajat keanggotaannya paling tinggi yang ada disisi kiri bergerak ke kanan menuju nilai yang derajat keanggotaannya lebih rendah.[7]



Gambar 3. Representasi Linear Turun

Fungsi Keanggotaan:

$$\mu [x, a, b] = \begin{cases} (b - a); & a \leq x \leq b \\ 0 & x \geq b \end{cases} \quad (2)$$

Dimana,

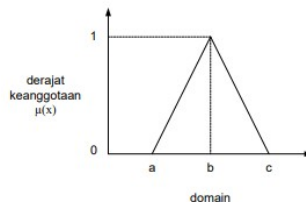
a = Nilai dominan dengan derajat keanggotaan satu.

b = Nilai dominan dengan derajat keanggotaan nol.

x = Nilai input yang diubah kedalam bilangan fuzzy.

2. Representasi Kurva Segitiga

Kurva segitiga merupakan gabungan dua representasi linear yaitu linear naik dan turun. Sebagaimana ditunjukkan dibawah ini:



Gambar 4. Representasi Kurva Segitiga

Fungsi Keanggotaan:

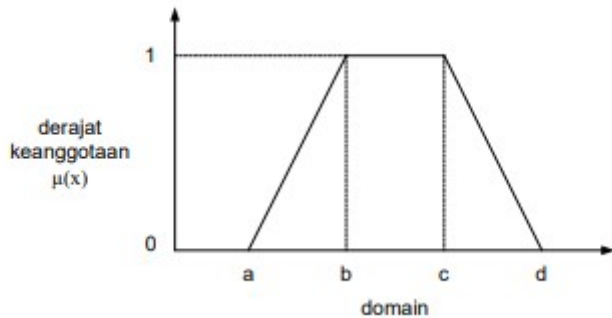
$$\mu [x, a, b, c] = \begin{cases} 0 & x \leq a \text{ atau } x \geq d \\ (x - a)/(b - a) & a \leq x \leq b \\ (c - x)/(c - b) & b \leq x \leq c \\ 0 & x \geq d \end{cases} \quad (3)$$

Dimana,

- a =Nilai dominan terkecil dengan derajat keanggotaan nol.
- b =Nilai dominan terkecil dengan derajat keanggotaan satu.
- c =Nilai dominan terbesar dengan derajat keanggotaan satu.
- d =Nilai dominan terbesar dengan derajat keanggotaan nol.

3. Representasi Kurva Trapesium

Kurva trapesium umumnya memiliki bentuk seperti trapesium dengan beberapa titik bernilai satu, sebagaimana ditunjukkan gambar berikut:



Gambar 5. Representasi Kurva Trapesium

Fungsi Keanggotaan:

$$\mu [x, a, b, c, d] = \begin{cases} 0 & x \leq a \text{ atau } x \geq d \\ (x - a)/(b - a) & a \leq x \leq b \\ 1 & b \leq x \leq c \\ (d - x)/(d - c) & \end{cases} \quad (4)$$

Dimana,

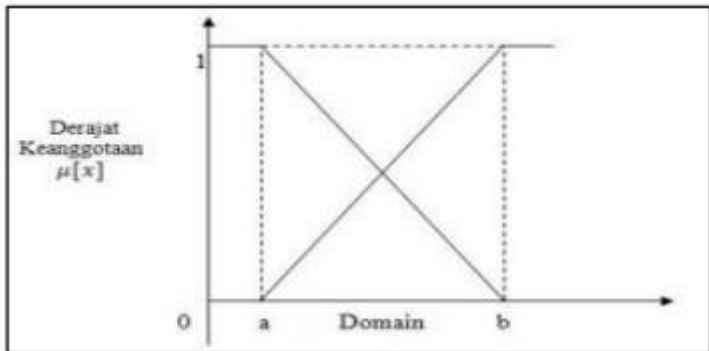
a = Nilai dominan terkecil dengan derajat keanggotaan nol.

b = Nilai dominan dengan derajat keanggotaan satu.

c = Nilai dominan terbesar dengan derajat keanggotaan nol.

4. Representasi Kurva Bentuk Bahu

Titik yang berada di tengah-tengah variabel direpresentasikan dalam bentuk segitiga. Sisi kanan dan kiri akan naik dan turun namun terkadang salah satu sisi variabel tersebut tidak mengalami perubahan. Himpunan *fuzzy* bahu bukan segitiga dalam hal ini digunakan untuk mengakhiri variabel pada daerah *fuzzy*. Bahu kiri akan bergerak dari benar ke salah, sedangkan bahu kanan bergerak sebaliknya.[6]



Gambar 6. Representasi Kurva Bentuk Bahu

Fungsi Keanggotaan:

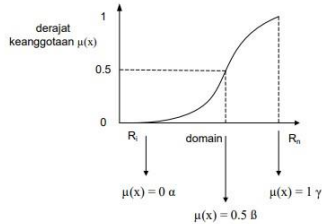
$$\mu [x, a, b.] = \begin{cases} 0 & x \leq b \\ (b - x)/(b - a) & a \leq x \leq b \\ 1 & x \geq a \\ 0 & x \leq a \\ (x - a)/(b - a) & a \leq x \leq b \\ 1 & x \geq b \end{cases} \quad (5)$$

5. Representasi Kurva - S

Representasi kurva-s atau sigmoid merupakan kurva pertumbuhan dan penyusutan dengan bentuk kenaikan dan penurunan tak linear.[6]

a. Kurva S Pertumbuhan

Kurva ini bergerak dari sisi kiri (0) ke kanan (1). Fungsi keanggotaannya akan bertumpu pada 50% nilai keanggotaannya yang biasa disebut titik infleksi [7].



Gambar 7. Representasi Kurva -S pertumbuhan

Fungsi Keanggotaan:

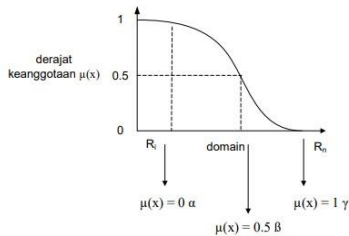
$S [x, a. \beta. \gamma]$

$$= \begin{cases} 0 & x \leq a \\ 2 \left(\frac{x-a}{\gamma-a} \right)^2 & a \leq x \leq \beta \\ 1 - 2 \left(\frac{\gamma-x}{\gamma-a} \right)^2 & \beta \leq x \leq \gamma \\ 1 & x \geq \gamma \end{cases}$$

$$\begin{aligned} x &\leq a \\ a &\leq x \leq \beta \\ \beta &\leq x \leq \gamma \\ x &\geq \gamma \end{aligned}$$

(6)

b. Kurva S Penyusutan



Gambar 8. Representasi Kurva -S penyusutan

Fungsi Keanggotaan:

$$S [x, a. \beta. \gamma] = \left\{ \begin{array}{l} 1 \\ 1 - 2 \left(\frac{x-a}{\gamma-a} \right)^2 \\ 2 \left(\frac{\gamma-x}{\gamma-a} \right)^2 \\ 0 \end{array} \right\} \quad \begin{array}{l} x \leq a \\ a \leq x \leq \beta \\ \beta \leq x \leq \gamma \\ x \geq \gamma \end{array} \quad (7)$$

2.2.3. Fuzzyfication

Dalam *fuzzy logic* proses yang dilakukan pertama kali adalah *fuzzyfication* (fuzzifikasi). Fuzzifikasi adalah sebuah proses yang digunakan untuk mengubah variabel non *fuzzy* (numerik) menjadi variabel *fuzzy* (linguistik). Perubahan ini dilakukan melalui fungsi keanggotaan yang telah disusun maka, nilai masukan tersebut menjadi informasi berguna untuk proses pengolahan secara *fuzzy*.

2.2.4. Rule Evaluation

Rule evaluation dalam *fuzzy* merujuk pada proses penilaian atau evaluasi aturan *fuzzy* dalam sebuah sistem control *fuzzy logic*. Setiap aturannya memiliki bagian evaluasi yang menghubungkan kondisi input dengan aksi atau output yang harus diambil dalam evaluasinya. Umum nya *rule fuzzy* atau *rule evaluation* dinyatakan dalam bentuk “If-Then” yang merupakan inti dari relasi *fuzzy*.

Dalam penerapan *rule* ataupun kaidah - kaidah *fuzzy* dikenal juga suatu istilah yang disebut sistem inferensi, dimana inferensi singkatnya merupakan sebuah penarikan kesimpulan dari sekumpulan kaidah atau rule fuzzy.[6] Metode Inferensi yang umumnya dikenal ada dua yakni, metode inferensi “Sugeno” dan “mamdani.” Pada proyek ini akan diterapkan metode inferensi mamdani.

1. Inferensi fuzzy metode Mamdani

Metode *fuzzy* Mamdani merupakan salah satu bagian dari *fuzzy inference system* yang berguna untuk penarikan kesimpulan atau suatu pengambilan keputusan terbaik dalam permasalahan yang tidak pasti. Metode *fuzzy* mamdani diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Dalam prosesnya metode

ini menggunakan kaedah-kaedah linguistik dan memiliki algoritma *fuzzy* yang dapat dianalisis secara matematika, sehingga lebih mudah dipahami.[8] Proses pengambilan keputusan dengan menggunakan metode Mamdani untuk memperoleh keputusan yang terbaik, dilakukan dengan melalui beberapa tahapan, yaitu:

1. Pembentukan himpunan *fuzzy*
2. aplikasi fungsi implikasi
3. komposisi aturan
4. defuzzifikasi

Alasan dipilihnya inferensi Mamdani sebagai sistem inferensi dalam proyek ini adalah:

1. Kemudahan Interpretasi
2. Toleransi Terhadap Ketidakpastian
3. Penggunaan Variabel Linguistik
4. Aplikasi yang Luas
5. Kompatibilitas Dengan Sistem Manusia

Metode mamdani dapat menggambarkan penalaran yang mirip dengan manusia, sehingga lebih mudah diadopsi dalam aplikasi yang melibatkan interaksi ataupun pengambilan keputusan oleh manusia. Misalnya dalam proyek ini yang mana output *fuzzy* akan membantu keputusan harus diambil oleh manusia, apakah harus menambah kecepatan atau tidak saat alarm berbunyi.

Pada umumnya baik Sugeno atau Mamdani merupakan metode fuzzy yang bagus dalam berbagai pengaplikasian. Akan tetapi yang menjadi alasan utama penulis lebih memilih Inferensi Mamdani adalah bentuk ekspresi dari rule *evaluation* nya yang mudah di mengerti.

2.2.5. Defuzzification

Defuzifikasi merupakan sebuah proses mendapatkan hasil yang bisa diukur dengan logika yang tajam. Defuzifikasi mengartikan derajat keanggotaan himpunan *fuzzy* menjadi suatu keputusan tertentu atau nilai riil. Input dari *defuzzification* merupakan suatu himpunan yang didapat melalui komposisi aturan-aturan, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut. *Defuzzification* merupakan lanjutan dari proses *rule base*. [6] Adapun beberapa cara dalam proses *defuzzification* adalah:

1. Center of Gravity / centroid

Dengan cara ini, solusi *crisp* di dapatkan dengan mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan setengah dari jumlah total nilai keanggotaan pada daerah *fuzzy*. [9] Secara umum dirumuskan pada persamaan 8 untuk variabel kontinyu dan persamaan 9 untuk variabel diskrit.

$$z = \frac{\int_b^a z\mu(z) dz}{\int_a^b \mu(z) dz} \quad (8)$$

$$z = \frac{\sum_{j=1}^n z\mu(z_j)}{\sum_{j=1}^n \mu(z_j)} \quad (9)$$

2. Bisektor

Pada metode *defuzzifikasi* ini, solusi *crisp* diperoleh dengan cara mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan setengah dari jumlah total nilai keanggotaan pada daerah *fuzzy*. [9]

$$\int_a^{z^{BOA}} \mu(z) dz = \int_{z^{BOA}}^{\beta} \mu(z) dz \quad (10)$$

Dimana:

$$\alpha = \min \{z/z \in Z\} \quad \beta = \max \{z/z \in Z\} \quad (11)$$

3. Mean of maximum

Inferensi jenis ini bekerja dengan cara menghitung nilai rata-rata dari titik domain yang memiliki derajat keanggotaan maksimum pada output *membership function*. Nilai rata-rata ini akan menjadi hasil akhir dari sistem *fuzzy*. Metode Mean of maximum ini termasuk metode yang sederhana dan efisien karena tidak

memerlukan perhitungan integral seperti yang terjadi pada *centroid*, meskipun akurasi cenderung lebih rendah jika bentuk distribusi output tidak merata atau simetris. Rumus yang digunakan dalam metode ini adalah.

$$z = \frac{X_1 + X_2 + X_3 + X_n}{n} \quad (12)$$

4. Largest of maximum

Largest of maksimum menentukan nilai output dengan cara mengambil nilai terbesar dari domain output yang memiliki derajat keanggotaan maksimum. Setelah proses *inferensi fuzzy* menghasilkan kurva atau bentuk *fuzzy* pada variabel output, akan muncul beberapa nilai output yang nilainya maksimum, metode ini akan mencari nilai yang paling besar kemudian menjadikannya hasil defuzifikasi.

5. Smallest of maximum

Pada cara ini, solusi *crisp* diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum. Teknik *fuzzy logic* menggunakan suatu bentuk teori matematika, yang disebut himpunan *fuzzy*, untuk mensimulasikan proses pertimbangan normal manusia dengan memungkinkan komputer untuk berperilaku sedikit lebih seksama dan logis daripada yang dibutuhkan sedikit lebih seksama dan logis daripada yang dibutuhkan metode komputer konvensional. Jadi, tujuan utama *fuzzy logic* adalah bagaimana membuat komputer bisa beroperasi seperti layaknya logika manusia dan menghilangkan batas antara manusia dan kemampuan komputer.[9]

2.3. Komponen Elektronika

2.3.1. Sensor Tf Luna Lidar

Merupakan sensor lidar cahaya yang bekerja menggunakan prinsip (*Time of Flight ToF*) yang mana prinsip ini menghitung lamanya waktu dari cahaya yang dipancarkan untuk dipantulkan kembali ke sensor setelah cahaya menabrak objek, ia mengadopsi sinar infrared dengan panjang gelombangnya 850 nm sebagai sumber cahaya yang dipancarkan.[10] Dengan prinsip kerja yang demikian tentunya kecepatan berkendara mempengaruhi nilai pembacaan.



Gambar 9. Tf-Luna-Lidar Range Sensor

Spesifikasi dari Tf Luna Lidar ini adalah:

Tabel 3. Spesifikasi Sensor Tf Luna LiDAR

Rentang pengukuran	0.2m – 8m
Resolusi jarak	1cm
Frekuensi	100Hz
Resistensi cahaya sekitar	70 Klux
Sumber Cahaya	VCSEL
Panjang Gelombang pusat	850nm
Keamanan mata	Class1(IEC60825
Tegangan sumber	3.7V-5.2V
Arus rata rata	≤70mA
Daya	≤0.35W
Arus Puncak	150mA
Komunikasi	UART/I2C
Dimensi	35mm*21.25mm*13.5mm

Dengan prinsip kerja yang telah disebutkan sebelumnya maka perhitungan jarak deteksi sensor ini dirumuskan sebagai berikut:

$$d = \frac{c}{2} \cdot \frac{1}{2\pi f} \cdot \Delta\varphi \quad (13)$$

Dimana:

d =Jarak ukur

c =Kecepatan Cahaya

$\Delta\varphi$ =perubahan sudut

2.3.2. Hall Effect Sensor

Merupakan sebuah sensor yang biasa di gunakan untuk mendeteksi kekuatan dan arah medan magnet dari sebuah magnet permanen, keluaran dari sensor ini bervariasi tergantung kekuatan medan magnet yang dideteksi. Sensor ini di aktifkan oleh medan magnet eksternal dan medan magnet sendiri memiliki dua karakteristik penting yakni kerapatan fluks (B) dan polaritas kutub (utara dan selatan). Sinyal output dari sensor ini yang berupa fungsi kerapatan magnet di sekitar perangkat, saat keluaran tersebut melebihi ambang batas yg ditetapkan, sensor akan menghasilkan tegangan keluaran yang disebut tegangan hall V_H .



Gambar 10. Sensor Magnetic Effect Hall

Tabel 4. Spesifikasi Sensor Hall Effect

Tegangan Sumber	4.5-25V
Arus Sumber	4.4 mA
Tegangan saturasi keluaran	165 mV
Waktu kenaikan output	0.04 μ s
Waktu turun output	0.18 μ s
Kerapatan Flux Magnetic	Tak terbatas
Dimensi	4.6mm x 4.65mm x 4.25mm

Sensor ini memiliki output linear maupun digital. Output linear(analog) diambil dari tegangan output *operational amplifier* dengan tegangan keluaran yang berbanding lurus terhadap medan magnet yang melewati sensor Hall, Di rumuskan sebagai:

$$V_H = R_H \left(\frac{l}{t} \times B \right) \quad (14)$$

Dimana:

V_H =Tegangan Hall dalam volt

R_H =Koefisien Efek Hall

I =Arus

t =Ketebalan sensor dalam mm

B =Flux Magnetic

Sensor hall effect keluaran analog memberikan tegangan keluaran kontinu yang meningkat dengan medan magnet yang kuat serta menurun dengan medan magnet yang lemah. Sinyal akan meningkat saat kekuatan medan magnet itu meningkat hingga akhirnya akan jenuh oleh batasan yang diberikan oleh catu daya. Sedangkan sensor hall effect dengan keluaran digital mempunyai pemicu (*trigger*) Schmitt dengan hysteresis bawaan yang terhubung pada OP-AMP, yang mana Ketika fluks manetik yang melalui sensor hall melebihi nilai yang ditetapkan, output sensor akan beralih dengan cepat dari kondisi "OFF" ke kondisi "ON" tanpa pantulan kontak apapun.[11]

Tabel 5. Beberapa Literatur Penelitian Terdahulu yang dijadikan Acuan

No	Tahun	Penulis	Jurnal	Metode	Kelemahan
1	2020	Hammada Abbas, Sarida Sarida, Suradi Suradi, Ismail Ismail	Perancangan dan simulasi system pengontrolan jarak aman pengendara mobil menggunakan sensor ultrasonic berbasis Arduino uno	Menggunakan Arduino Uno dan dua buah sensorsonik untuk mendeteksi jarak	Dalam penelitian ini, alat yang dibuat belum terpasang pada mobil. Hanya dilakukan simulasi dengan mobil mremote control
2	2020	Immanuel Yosua Lonteng, gunawan Gunawan, Isa Rosita	Rancang Bangun Simulasi Alat Pendeteksi Jarak Aman Antar kendaraan Menggunakan Sensor Ultrasonik Berbasis Arduino	Menggunakan servo motor/ dc motor untuk simulasi perubahan kecepatan berdasarkan jarak objek terhadap sensor ultrasonik	Masih berupa simulasi pada motor servo

3	2023	Yonas Juniantiko Putro, Theophilus Wellem	Implementasi system untuk mendeteksi jarak aman kendaraan bermotor menggunakan arduino dan sensor ultrasonik	Mendeteksi jarak dengan sensor ultrasonik dan memberi notifikasi via buzzer dan tampilan LCD	Jangkauan Sensor terlalu kecil
4	2018	Asri Mulyani	Perancangan Sensor Jarak Aman Kendaraan Bermotor Menggunakan Mikrokontroler Arduino Uno R3	Merancang sistem pendeteksian jarak aman berkendara menggunakan ultrasonik	Hanya menjadikan jarak sebagai parameter kondisi
5	2020	Samsudin Samsudin, Maya Juliana Ritonga, Muhammad Ikhsan	Penerapan logika <i>Fuzzy</i> Pada sistem peringatan jarak aman sepeda motor berbasis mikrokontroler	Menggunkan <i>fuzzy</i> dalam sistem peringatan	Pemasangan pada sepeda motor cukup membahayakn alat dan pengendara

6	2019	Ratna Aisuwarya	<i>Design of Bicycle's Speed Measurement System Using Hall Effect Sensor</i>	Menggunakan Sensor Hall Effect untuk mengukur kecepatan sepeda	Sejauh penelitian yang dilakukan dengan sensor ini cukup baik. Tetapi akan jauh lebih baik jika disediakan fitur pelacakan lokasi (GPS) .
7	2023	Agustina Dwi Cahyani	Implementasi Fuzzy Logic Control Pada Alat Pengering Cengkeh Otomatis	Menggunakan Fuzzy untuk mengontrol pengeringan cengkeh otomatis dengan sensor suhu DS18B20, sensor <i>loadcell</i> , sensor kadar air sebagai input.	Masih menggunakan telegram sebagai perantara, baiknya di buatkan web tersendiri untuk mengontrol dan monitoring

8	2022	Muhammad Khoirul Imam, Endi Permata, Desmira Desmira	Sistem Kontrol Penyiram otomatis Tnaman Tomat Menggunakan Wemos D1 R1	Menggunakan <i>Fuzzy Logic</i> untuk melakukan penmyiraman otomatis	<i>Rule Evaluation fuzzy</i> yang dideskripsikan dan diuji hanya pada kondisi tidak di siram sedangkan untuuk keadaan disiram tidak ditsebutkan
9	2023	Junaidi	Implementasi <i>Fuzzy Logic</i> Dengan Metode Mamdani Untuk Sistem Pendukung Keputusan Kinerja Dosen	Menggunakan <i>Fuzzy Logic</i> Mamdani untuk menentukan kinerja dosen	Prose penegasan atau <i>Defuzzyfikasi</i> masih menggunakan <i>MATLAB online</i> , baiknya menggunakan fasilitas kampus atau dibuat sendiri jika ingin diterapkan di kampus

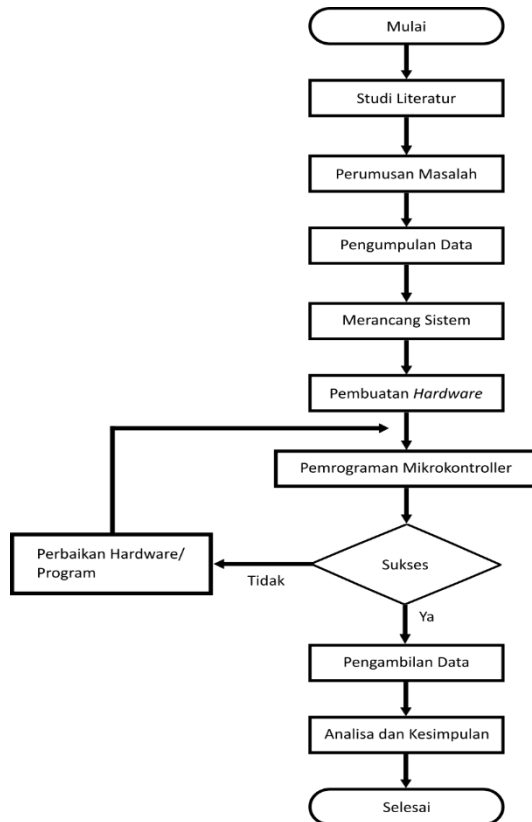
10	2022	Yesyurun Masiakh Agape, Deddy Susilo, Andreas Febrianto	Prancangan Alat Deteksi Menggunakan Fuzzy Mamadani Dan Sensor MG-811 Guna Menghindari Keracunan Akibat Gas.	Menggunakan metode FIS Mamdani untuk mendeteksi keracunan akibat gas pada suatu ruangan	Pada penelitian ini dikatakan akan melibatkan alat yang akan mengurangi CO2 atau exhaust namun pada pengujiannya tidak dicantumkan integrasi dari alat tersebut dengan sistem yang ada
----	------	---	---	---	--

Bab 3. Metodologi Penelitian

3.1. Perancangan

3.1.1. Perancangan Penelitian

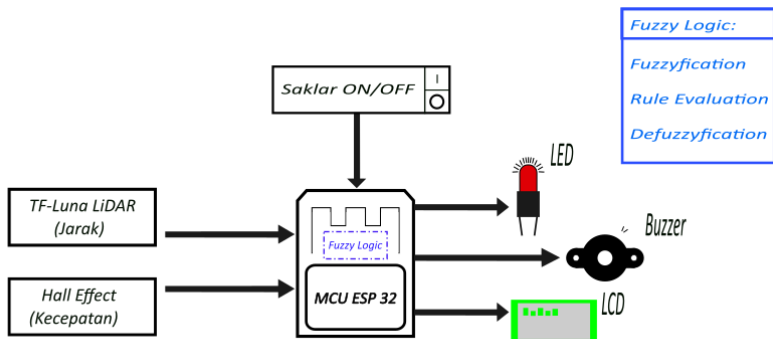
Dalam mengerjakan penelitian dengan metode deskriptif kuantitatif ini diperlukan sebuah rancangan penelitian, berikut merupakan tahapan rancangan penelitian.



Gambar 11. Diagram alir penelitian

3.1.2. Perancangan Sistem Kerja

Selain diagram alir dari proses penelitian dirancang juga blok diagram yang akan menggambarkan sistem kerja dari alat ini, blok diagram ini dengan singkat dapat menggambarkan cara dari alat ini bekerja:

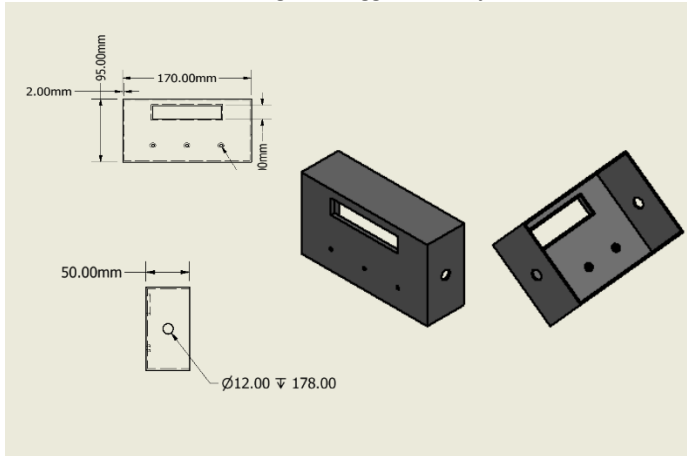


Gambar 12. Blok Diagram sistem kerja alat

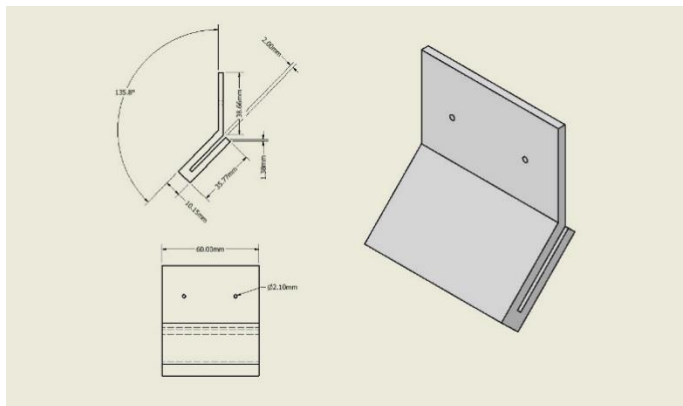
Sistem kerja dari gambar diatas mulai dari *trigger* oleh saklar *on/off*, selanjutnya sensor jarak dan kecepatan akan mulai mendeteksi dan ketika sensor mendapati nilai yang menjadi *membership function* dari *fuzzy*, proses *fuzzy logic* akan dimulai. Diawali dengan *Fuzzyfication* atau perubahan variabel nyata menjadi variabel *fuzzy* yang kemudian variabel itu akan melewati proses penyesuaian dengan aturan yang di buat atau disebut juga proses *Rule Evaluation* tahapan akhir dalam proses ini adalah *Defuzzyfication*, di sini hasil dari inferensi akan di konversi ke bilangan real. Nah setelah proses ini yang akan dilakukan adalah mempertimbangkan apakah akan menyalakan *LED* atau tidak, membunyikan *buzzer* atau tidak, yang pertimbangan itu tadi mengikuti metode defuzzifikasi yang dipakai. Untuk metode tersebut penulis memakai metode defuzzifikasi inferensi mamdani jenis *centroid*.

3.1.3. Perancangan Mekanik

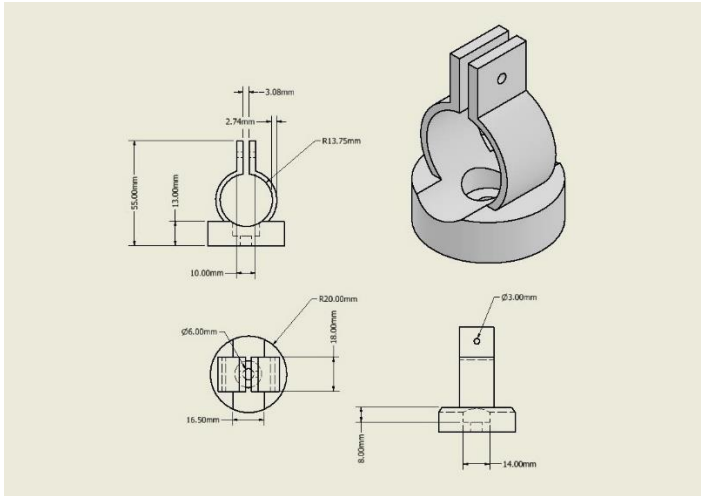
Desain mekanik dibuat dengan menggunakan *software Autodesk Inventor*.



Gambar 13. Main Case Design



Gambar 14. Lidar Case Design



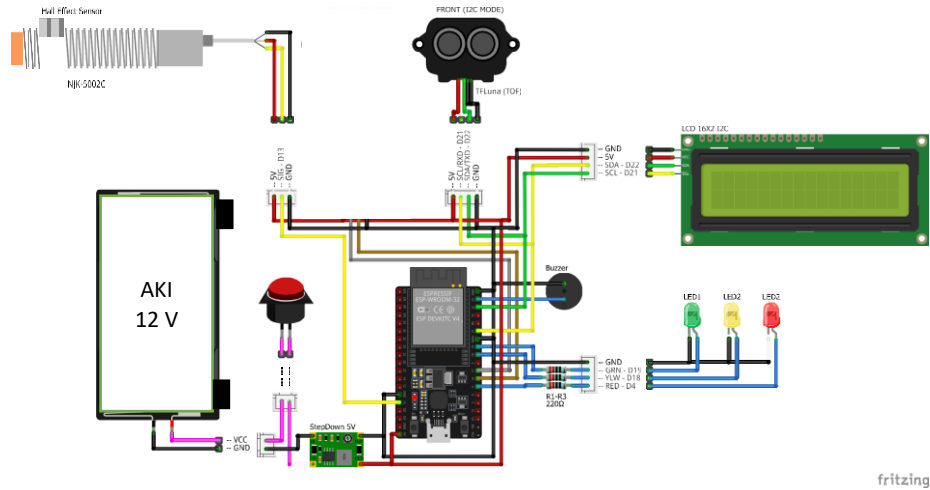
Gambar 15. Main Case to Bicycle handlebar Design



Gambar 16. Perangkat Terpasang

3.1.4. Perancangan Elektrikal

Selain desain mekanik dirancang juga desain elektrikal yang nantinya akan menjadi suatu sistem yang akan menjalankan alat ini.



Gambar 17. Rangkaian elektrikal

Seperti yang terlihat di gambar, alat ini akan di *supply* oleh aki yang kapasitasnya 12v, baterai ini terhubung ke *switch* yang akan terhubung pula ke regulator 5V atau 3.3V, kemudian koneksi dilanjutkan ke pin input. Untuk rangkaian *main device* mikrokontroler ESP32 akan terhubung dengan sensor Tf-Luna-LiDAR, *supply* untuk Tf Luna LiDAR adalah pin 3.3V pada ESP yang terhubung langsung ke VCC sensor Tf-Luna-LiDAR. Komunikasi antara Tf Luna LiDAR dan ESP32 ialah komunikasi lewat I2C Serial Clock Line dan Serial Data Line (SCL/SDA). Untuk Display nya akan disajikan dalam sebuah LCD I2C 16x2 lewat pin SCL SDA yang juga akan terhubung ke pin SCL dan SDA dari sensor hall effect. Sebagai output yang berupa LED dan buzzer akan terhubung secara berurutan dari LED merah, kuning, hijau, dan buzzer ke pin GPIO 26, 27, 14, 12, yang sebelum terhubung ke LED akan melewati resistor terlebih dahulu.

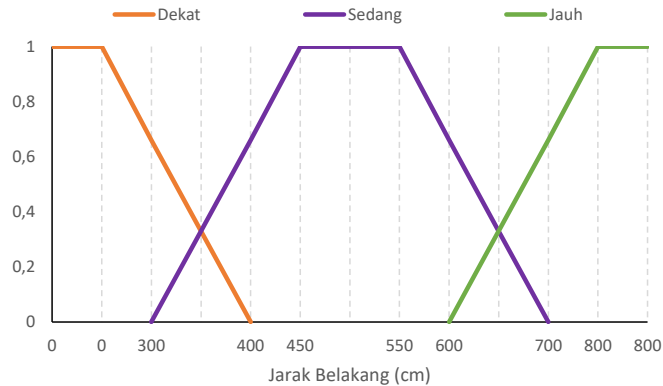
3.1.5. Perancangan Fuzzy

1. Membership Function dan Derajat Keanggotaan

Dalam sistem *fuzzy* yang diterapkan pada alat ini kami membagi jarak 800 cm menjadi tiga bagian dan menjadikannya sebagai *membership input* untuk sistem *fuzzy*.

Dekat :0 cm – 400 cm
Sedang :300 cm – 700 cm
Jauh :600 cm – 800 cm

Dengan nilai yang ada ini bila kita gambarkan dalam sebuah grafik akan membentuk kurva singleton sebagai berikut.



Gambar 18. Kurva Membership Jarak

Dari kurva di atas bila dilakukan perhitungan kita akan mendapatkan derajat keanggotaan yang beragam, sebagai contoh kasus untuk melakukan perhitungan kita akan mencoba mencari nilai derajat keanggotaan dari jarak 400 cm.

$$\mu \text{ dekat } [400] = \frac{d-x}{d-c} \quad \text{maka,} \quad (15)$$

$$\mu \text{ dekat } [400] = \frac{400 - 400}{400 - 0} = \frac{0}{400} = 0$$

Dengan:

$$x = 400$$

$$d = 400$$

$$c = 0$$

$$\mu \text{ sedang } [400] = \frac{x-a}{b-a} \quad \text{maka,} \quad \mu \text{ sedang } [400] = \frac{400 - 300}{450 - 300} = \frac{100}{150}$$

Dengan:

$$x = 400$$

$$a = 300$$

$$b = 450$$

$$= \frac{100}{150} = \frac{2}{3} = 0,66$$

$$\mu \text{ jauh } [400] = 0$$

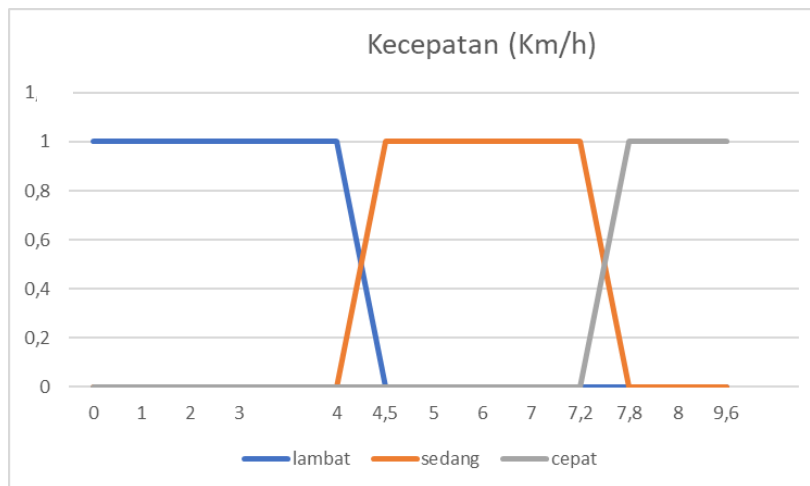
Dalam input jarak yang lebih banyak, derajat keanggotaan *membership function* yang ada ialah sebagai berikut.

Tabel 6. Derajat Keanggotaan *Membership* Jarak

Jarak (cm)	Crisp Value		
	Dekat	Sedang	Jauh
0	1	0	0
300	0,66	0,44	0
350	0,33	0,33	0
400	0	0,66	0
450	0	1	0
500	0	1	0
550	0	1	0
600	0	0,66	0
650	0	0,33	0,33
700	0	0	0,66
800	0	0	1
800	0	0	1

Membership input kecepatan akan dideteksi lewat sensor *magnetic hall effect* dengan nilai.

Lambat :0 km/h – 4,5 km/h
Sedang :4 km/h – 7,8 km/h
Cepat :7,2 km/h – 9,6 km/h



Gambar 19. Kurva *Membership* Kecepatan

Dari kurva di atas bila dilakukan perhitungan kita akan mendapatkan derajat keanggotaan yang beragam, sebagai contoh kasus untuk melakukan perhitungan kita akan mencoba mencari nilai derajat keanggotaan dari 7,4 Km/h.

$$\mu \text{ lambat } [7,4] = 0$$

$$\mu \text{ sedang } [7,4] = \frac{d-x}{d-c} \quad \text{maka,} \quad \mu \text{ sedang } [7,4] = \frac{7,8 - 7,4}{7,8 - 7,2} = \frac{0,4}{0,6} = 0,66$$

Dengan:

$$x = 7,4$$

$$c = 7,2$$

$$d = 7,8$$

$$\mu \text{ cepat } [7,4] = \frac{x-a}{b-a} \quad (16)$$

Dengan:

$$x = 7,4$$

$$a = 7,2$$

$$b = 7,8$$

$$\mu \text{ cepat } [7,4] = \frac{7,4 - 7,2}{7,8 - 7,2} = \frac{0,2}{0,6} = 0,33$$

Kemudian dalam input yang lebih banyak, derajat keanggotaan nya disajikan dalam tabel berikut.

Tabel 7. Derajat Keanggotaan *Membership* Kecepatan

Kecepatan (Km/h)	Derajat Kenggotaan		
	lambat	sedang	cepat
0	1	0	0
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
4,2	0,4	0,6	0
4,5	0	1	0
5	0	1	0
6	0	1	0
7	0	1	0
7,2	0	1	0
7,4	0	0,66	0,33
7,8	0	0	1
8	0	0	1
9	0	0	1
9,6	0	0	1

2. Rule Evaluation

Rule evaluation ini di buat dengan nilai berbeda untuk tiap orang karena *rule* di buat berdasarkan intuisi ataupun nalar manusia sehingga tiap orang memiliki intuisi berbeda. Bila kita mengacu pada contoh kasus pada perhitungan *membership function* sebelumnya, dimana nilai jarak terdeteksi 400 cm dan kecepatan terdeteksi 7,4 km/h dan kita menggunakan metode *fuzzy mamdani* dengan aturan **"IF A AND B THEN"** maka *Rule Evaluation* dihitung dengan rumus.

$$\mu \text{ Rule} = \min (\mu A, \mu B)$$

Tabel 8. Rule Evaluation Beserta Hasil Hitung Kenggotaan Kecepatan 7,4 Km/h dan Jarak 400cm

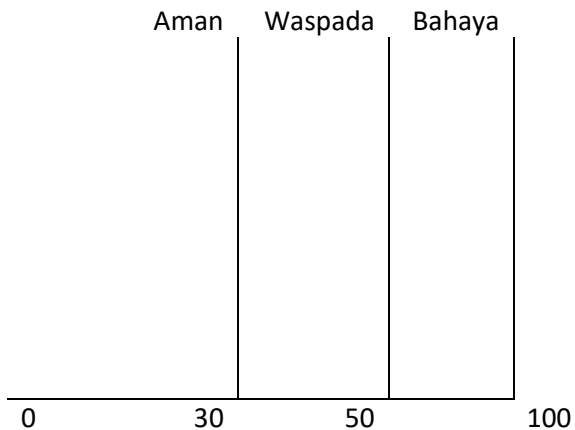
		x (Distance)		
		0	0,66	0
		Dekat	Sedang	Jauh
y (Speed)	Lambat 0	Waspada 0	Aman 0	Aman 0
	Sedang 0,66	Bahaya 0	Waspada 0,66	Aman 0
	Cepat 0,33	Bahaya 0	Bahaya 0,33	Waspada 0

Rule set ini bila mencapai tiap kondisi yang ada akan menghasilkan output pada perangkat berupa peringatan seperti suara buzzer dan nyala lampu LED yang mana level peringatan tersebut berupa.

Tabel 9. Level Peringatan Sistem Fuzzy

Level Peringatan		LED	BUZZER
1	Aman	HIJAU	MATI
2	Waspada	KUNING	BEEP 0.7s
3	Bahaya	MERAH	BEEP 0.2s

Bila kita buat persentase keadaan aman lewat grafik output membership singleton maka gambarannya adalah sebagai berikut.



Gambar 20. Grafik Singleton Output Sistem Fuzzy

3. Defuzzyfication

Dalam proses defuzzifikasi ini karena kita menggunakan metode inferensi Mamdani yang mana itu adalah metode *Centroid/Center of Gravity* dan kita kembali lagi ke contoh kasus perhitungan sebelumnya dengan nilai kecepatan 7,4 km/h dan jarak 400 cm, maka rumus yang paling sesuai untuk digunakan adalah.

$$z = \frac{\sum x_j \mu(x_j)}{\sum \mu(x_j)} \quad (17)$$

$$\text{Sehingga, } z = \frac{(0.66 \times 50) + (0.33 \times 100)}{(0.66 + 0.33)} = \frac{33 + 33}{0.99} = 66,67$$

Dari proses defuzzifikasi ini kita mendapatkan nilai 66,67 sebagai output atau nilai tegas, yang mana dalam alat yang kita gunakan ini kita menghitung persentase bahaya saat berkendara. Maka 66,67 % ini termasuk kedalam keadaan bahaya dimana jika keadaan bahaya terdeteksi alat akan memberi output berupa pesan notifikasi bahaya, bunyi beep *buzzer* selama 0,2 detik dan menyalanya lampu merah.

3.2. Alat dan Bahan

Berikut merupakan estimasi biaya dari belanja alat dan bahan yang akan dikeluarkan dalam pembuatan alat pendeteksi jarak aman berkendara sepeda berbasis sensor TF- Luna LiDAR dan sensor *Hall Effect*.

Tabel 10. Estimasi biaya

No.	Alat/bahan	Harga Satuan (Rp.)	Jumlah	Total (Rp.)	Keterangan
1	Mikrokontroller ESP32	80.000	1	80.000	Dana Pribadi
2	Sensor Tf-Luna-LiDAR	370.000	1	370.000	Dana Pribadi
3	Sensor Hall Effect	50.000	1	50.000	Dana Pribadi
4	Baterai 9V (<i>Non Rechargeable</i>)	15.000	1	15.000	Dana Pribadi
5	LED	1000	3	3.000	Dana Pribadi
6	Buzzer	20.000	1	20.000	Dana Pribadi
7	Modul regulator step down	8.000	1	8.000	Dana Pribadi
8	LCD I2C 16 x 2	26.000	1	26.000	Dana Pribadi
9	<i>Main Case</i>	12.000	1	12.000	Dana Pribadi
10	<i>Case</i> 3D Printing	600.000	Secukupnya	600.000	Dana Pribadi
11	Kabel Jumper	200	Banyak	50.000	Dana Pribadi
12	PCB	20.000	1	20.000	
	Total			1.254.000	

Bab 4. Pembahasan

Pada bab ini kita akan membahas dan menguji beberapa poin penting pada penelitian yang dilakukan ini

5.1. Penentuan indikator jarak dan kecepatan sebagai *membership input* untuk *system fuzzy*

Untuk menentukan nilai jarak dekat, sedang dan jauh, baiknya mengikuti suatu landasan atau aturan yang ada akan tetapi di Indonesia, standar jarak dekat, sedang dan jauh untuk sepeda belum ada tetapi untuk mobil dia memiliki standar. Standar ini juga diatur dalam PP No 43 tahun 1993 Pasal 63 yang mana disitu Kemenhub menjelaskan bahwa jarak aman berkendara adalah 3 detik dari kendaraan yang ada di depan maju sampai kita memajukan kendaraan. Jarak waktu tersebut memberi ruang gerak bagi pengendara untuk berhenti, berpindah jalur, atau berhenti secara tiba-tiba. Sehingga penulis mengikuti aturan ini dalam penentuan indikator fuzzy dalam penelitian ini.

Dalam tabel rekomendasi jarak aman oleh Kemenhub kita dapati bahwa tidak disajikan kecepatan aman untuk jarak delapan meter, akan tetapi kita mengenal ada aturan 3 detik dari kendaraan yang ada di depan maju sampai kita memajukan kendaraan. Kenapa 3 detik, waktu 3 detik adalah kombinasi dari 1- 1,5 detik untuk waktu reaksi manusia sebagai pengemudi ditambahkan dengan 1,5-2 detik waktu reaksi mekanis dari kendaraan. Sebagai penjelasan lebih lanjut terkait aturan 3 detik ini, kita misalkan bahwa kita berkendara dengan sebuah kendaraan lain yang ada didepan, kendaraan yang ada di depan ini baru saja melewati sebuah patokan, sebagai pengendara yang ada dibelakangnya kita harus mencapai patokan yang baru dilewati tersebut dalam waktu 3 detik. Kira-kira berapa kecepatan konstan yang kita perlukan untuk mencapai patokan tersebut selama 3 detik, maka kecepatan tersebutlah yang menjadi kecepatan maksimal paling aman untuk jarak berkendara saat itu. Dengan keterbatasan jangkauan sensor yang delapan meter, penulis mencari nilai kecepatan aman terhadap jarak ini dan waktu reaksi 3 detik tadi. Sehingga digunakan rumus sebagai berikut.

$$s = v \times t \iff v = \frac{s}{t} \quad (18)$$

Dimana,

s =Jarak(m)

v =Kecepatan (Km/h)

t = Waktu(s)

Nilai jarak kita ambil dari nilai sensor yang 8 meter, sedangkan waktu di ambil dari aturan yaitu 3 detik. Maka dengan rumus yang ada perhitungannya menjadi.

$$v = \frac{8 \text{ meter}}{3 \text{ detik}}$$
$$= 2,67 \text{ m/s}$$

Hasil perhitungannya masih dalam meter/second untuk itu kita perlu rubah ke Km/h.

$$v = 2,67 \text{ m/s} \times \frac{3600}{1000}$$
$$= 2,67 \text{ m/s} \times 3,6$$
$$= 9.6 \text{ Km/h}$$

Selanjutnya perlu kita lakukan perhitungan juga untuk menentukan batas awal kecepatan berkendara yang cepat dan menyesuaikannya dengan batas awal jarak jauh pada sensor, dimana nilai awal jarak adalah enam meter.

$$v = \frac{6 \text{ meter}}{3 \text{ detik}}$$
$$= 2, \text{ m/s}$$

Kemudian diubah juga ke km/h sehingga mendapatkan hasil.

$$v = 2 \text{ m/s} \times \frac{3600}{1000}$$
$$= 2 \text{ m/s} \times 3,6$$
$$= 7,2 \text{ Km/h}$$

Sehingga dengan ini kita dapatkan bahwa nilai kecepatan maksimal yang memenuhi kondisi aman terhadap jarak delapan meter adalah 9.6 Km/h dan kecepatan yang aman untuk jarak enam meter sebagai batas awal jarak jauh adalah 7,2 km/h. Jadi dengan kecepatan ini dan jarak berkendara enam sampai delapan meter waktu reaksi 3 detik tadi dapat kita penuhi tanpa terjadinya kecelakaan. Rentang kecepatan 7.2 km/h sampai dengan 9,6 km/h ini akan menjadi acuan untuk menentukan kondisi aman untuk kombinasi rule cepat dan jauh dalam sistem fuzzy nantinya. Walaupun orang orang biasanya berkendara dengan kecepatan cepat yang melebihi nilai ini, akan tetapi kita akan menjadikan

nilai ini menjadi angka maksimal yang akan memberi output alarm aman. Hal ini dilakukan agar penentuan rule fuzzy sesuai standard.

Tabel 11. Nilai *Membership input* untuk kecepatan dan jarak dalam sistem *fuzzy*

No	Indikator Kecepatan sepeda	Nilai (kph)	Indikator jarak	Jarak (0m-8m)
1	Lambat	0 – 4,5 Kph	Dekat	0 – 4
2	Sedang	4 – 7,8 Kph	Sedang	3 – 7
3	Cepat	7,2 – 9,6 Kph	Jauh	6 - 8

5.2. Analisa dan Pembahasan Sistem Fuzzy

Alat ini menjalankan sebuah program *fuzzy* dalam pengambilan keputusan untuk membunyikan alarm atau tidak serta jenis alarm seperti apa yang harus di bunyikan. Dari sini akan dimunculkan sebuah pengujian dimana yang akan kita uji adalah kesesuaian output dari alat yang berupa peringatan kondisi aman, waspada dan bahaya. Untuk lebih jelasnya kita akan mengacu pada rule evaluation yang sudah dirancang sebagai acuan output.

Tabel 12. Rule Evaluation

		x (Distance)		
		Dekat	Sedang	Jauh
y (Speed)	Lambat	Waspada	Aman	Aman
	Sedang	Bahaya	Waspada	Aman
	Cepat	Bahaya	Bahaya	Waspada

Dari tabel *rule evaluation* diatas kita dapatkan sembilan aturan keadaan yang berupa.

1. **IF** jarak dekat **AND** kecepatan lambat **THEN**, kondisi adalah waspada (pesan notifikasi "waspada", *buzzer* beep selama 0.7 detik dan menyalanya lampu LED kuning).
2. **IF** jarak sedang **AND** kecepatan lambat **THEN**, kondisi adalah aman (pesan notifikasi "aman", dan menyalanya lampu LED hijau).
3. **IF** jarak jauh **AND** kecepatan lambat **THEN**, kondisi adalah Aman (pesan notifikasi "Aman", dan menyalanya lampu LED hijau).
4. **IF** jarak dekat **AND** kecepatan sedang **THEN**, kondisi adalah bahaya (pesan notifikasi "Bahaya", *buzzer* beep selama 0.2 detik dan menyalanya lampu LED merah).
5. **IF** jarak sedang **AND** kecepatan sedang **THEN**, kondisi adalah waspada (pesan notifikasi "Waspada", *buzzer* beep selama 0.7 detik dan menyalanya lampu LED kuning).
6. **IF** jarak jauh **AND** kecepatan sedang **THEN**, kondisi adalah aman (pesan notifikasi "Aman", dan menyalanya lampu LED hijau).

7. **IF** jarak dekat **AND** kecepatan cepat **THEN**, kondisi adalah bahaya (pesan notifikasi “Bahaya”, *buzzer* beep selama 0.2 detik dan menyalanya lampu *LED* merah).
8. **IF** jarak sedang **AND** kecepatan cepat **THEN**, kondisi adalah bahaya (pesan notifikasi “Bahaya”, *buzzer* beep selama 0.7 detik dan menyalanya lampu *LED* kuning).
9. **IF** jarak jauh **AND** kecepatan cepat **THEN**, kondisi adalah aman (pesan notifikasi “Aman”, dan menyalanya lampu *LED* hijau).

Dalam pengujiannya sembilan aturan yang ada ini berhasil berjalan pada alat dengan kondisi jarak dan kecepatan tertentu akan menghasilkan output tertentu pula sesuai dengan aturan yang sudah dirancang. Berikut data pengujiannya.

Tabel 13. Hasil pengujian kesesuaian output alarm terhadap sembilan rule evaluation yang ada

NO	Jarak Terdeteksi	Kecepatan Terdeteksi	Output seharusnya	Output Sebenarnya	Keterangan
1	164 cm	3 Km/h	Waspada	Waspada	Sesuai dengan aturan satu
2	502 cm	2 Km/h	Aman	Aman	Sesuai dengan aturan dua
3	723 cm	4 Km/h	Aman	Aman	Sesuai dengan aturan tiga
4	233 cm	7 Km/h	Bahaya	Bahaya	Sesuai dengan aturan empat
5	409 cm	5 Km/h	Waspada	Waspada	Sesuai dengan aturan lima
6	800 cm	7 Km/h	Aman	Aman	Sesuai dengan aturan enam
7	294 cm	8 Km/h	Bahaya	Bahaya	Sesuai dengan aturan tujuh
8	513 cm	9 Km/h	Bahaya	Bahaya	Sesuai dengan aturan delapan
9	763 cm	8 Km/h	Aman	Aman	Sesuai dengan aturan sembilan



Gambar 21. Pengujian Kesesuaian output terhadap sembilan rule evaluation yang ada

5.3. Pengujian Sensor LiDAR

Untuk memastikan keakuratan deteksi dari sensor LiDAR ini perlu dilakukannya pengujian. Pengujian dilakukan dengan membandingkan nilai terdeteksi dengan nilai yang diukur secara manual dengan meteran.

Tabel 14. Pengujian Sensor LiDAR

Pegujian	Jarak (system) (cm)	Jarak (manual) (cm) Menggunakan meteran	Perbedaan Jarak	Error
1	380 cm	350cm	30 cm	8.57 %
2	450 cm	458 cm	8 cm	1.75 %
3	580 cm	587 cm	7 cm	1.19 %
4	700 cm	711 cm	10.1 cm	2.98 %
5	200 cm	195 cm	5 cm	2.56 %
6	750 cm	788 cm	38 cm	4.82 %
7	530 cm	569 cm	39 cm	6.85 %
8	600 cm	632 cm	32 cm	5.06 %
9	650 cm	675 cm	25 cm	3.70 %
10	680 cm	693 cm	13 cm	1.88 %
Jumlah Error				39.36 %
Rata-rata Error				3.94 %

Selain melakukan pengukuran kami juga mengukur persentase error dengan rumus.

$$Error = \left(\frac{Sistem - Manual}{Manual} \right) \times 100 = \frac{380 - 350}{350} \times 100 = 8.57 \% \quad (19)$$

$$Rata-rata Error = \frac{jumlah\ kesalahan}{Banyaknya\ pengujian} \times 100 = \frac{39.36}{10} \times 100 = 3.94 \% \quad (20)$$

$$Ketepatan\ Akurasi = 100 \% - Rata-rata\ error = 100 - 3.94 = 96.06 \% \quad (21)$$



Gambar 22. Pengujian Sensor LiDAR

5.4. Pengujian Sensor Hall

Proses pengujian sensor hall dilakukan dengan mengukur tingkat ketelitian jarak deteksi antara sensor dan magnet yang ada. Untuk magnet yang digunakan adalah piringan magnet kecil yang nantinya akan dipasang di sekitar piringan cakram sepeda. Pengujian dilakukan dengan mencari jarak terbaik untuk pemasangan magnet terhadap sensor.

Tabel 15. Pengujian Sensor Hall

No	Jarak magnet ke sensor (mm)	Tegangan (VDC)	Kondisi deteksi
1	0 mm	4,900 V	Terdeteksi
2	2 mm	4,900 V	Terdeteksi
3	5 mm	4,899 V	Terdeteksi
4	8 mm	4,899 V	Terdeteksi
5	10 mm	4,899 V	Terdeteksi
6	12 mm	1,129 V	Tidak Terdeteksi
7	15 mm	1,128 V	Tidak Terdeteksi
8	18 mm	1,129 V	Tidak Terdeteksi
9	20 mm	1,129 V	Tidak Terdeteksi
10	30 mm	0 V	Tidak Terdeteksi



Gambar 23. Pengujian Jarak Pasang Sensor *Hall Effect*

Setelah menguji jarak deteksi magnet terbaik dengan sensor hall ditemukan bahwa dari jarak 0 mm - 10 mm tegangan kerja sensor berada di kisaran 4,80 VDC – 5,00 VDC dan magnet terdeteksi dengan baik, sedangkan pada jarak 11 mm - 30 mm tegangan kerja sensor ada di kisaran 0 VDC – 1,12 VDC, magnet tidak terdeteksi. Dengan demikian penulis menetapkan bahwa jarak pemasangan magnet terhadap sensor adalah 3 mm.

Selain itu sensor hall efek ini digunakan sebagai sensor yang akan mendeteksi kecepatan sepeda, yang mana dalam pendeteksian kecepatan kita akan mencari nilai RPM terlebih dahulu.

$$RPM = \frac{60}{t} \quad \text{menjadi, } RPM = \frac{60 \times 1.000.000}{1000.000} = \frac{60.000.000}{1000.000} \quad (22)$$

Dimana,

RPM = *Revolution per meter*

60 = Waktu dalam satuan menit (s)

t = Waktu per putaran (μ s)

Karena waktu perputaran dalam bentuk mikrodetik maka, dilakukan penyesuaian dengan 60 detik dikalikan 1.000.000 agar setara dengan mikrodetik.

Namun Kecepatan yang diinginkan disini adalah kecepatan dalam kilometer per jam, yang mana kita ketahui untuk mencari kecepatan dalam kilo meter perjam tersebut digunakan rumus.

$$KPH = \frac{\text{Jarak tempuh}}{\text{waktu}} \quad (23)$$

Akan tetapi rumus di atas perlu kita sesuaikan lagi karena kecepatannya kita dapat dari putaran roda, yang membuat nilai jarak tempuhnya dihitung sebagai berikut.

$$\text{Jarak tempuh(keliling roda dalam menit)} = RPM \times 2\pi r$$

$$\text{Jarak tempuh dalam jam} = RPM \times 2\pi r \times 60$$

Karena yang kita butuhkan adalah kilometer per jam maka rumus yang ada tadi harus kita bagi dengan 1 kilometer, sehingga rumus akhirnya adalah.

$$\text{Kecepatan(Km/h)} = \frac{RPM \times 2\pi r \times 60}{1000} \text{ atau Kecepatan(Km/h)} = \frac{2\pi r \times RPM \times 60}{1000}$$

Untuk pengujian dibawah ini penulis memutar pedal sepeda agar rodanya berputar kemudian mengamati dan menghitung *RPM* dengan menggunakan *timer* hp kemudian melakukan perhitungan secara manual dan membandingkan nya dengan nilai kecepatan yang ditampilkan oleh sistem pada LCD. Sebagai contoh pada pengujian satu nilai *RPM* dari pengamatan dengan timer hp adalah 10 kali putaran dalam 14,53 detik, sehingga.

$$\text{Jika di rubah ke menit maka } 14,53 \text{ detik menjadi } \frac{14,53}{60} = 0,2421 \text{ menit}$$

$$\text{Maka, } RPM = \frac{10 \text{ putaran}}{0,2421 \text{ menit}} = 41.30 \text{ rpm}$$

Dari *RPM* ini kita ubah ke kecepatan dalam bentuk *km/h*,

$$\begin{aligned} \text{Kecepatan (km/h)} &= \frac{2\pi r \times RPM \times 60}{1000} \text{ maka, } \text{km/h} = \frac{\pi \cdot d \times RPM \times 60}{1000} \\ &= \frac{3.14 \times 0.6604 \times 41.30 \times 60}{1000} \\ &= 5.13 \text{ km/h} \end{aligned}$$

Dimana,
d = 26 inci atau 0.6604 meter.

Selanjutnya penulis sajikan lebih banyak data pengujian yang dilakukan dalam tabel dibawah ini.

Tabel 16. Pengujian Kecepatan dengan Sensor Hall

Pegujian	Kecepatan terbaca (sistem) (Km/h)	Kecepatan (hitung manual) (Km/h)	Perbedaan Kecepatan	Error
1	6 Km/h	5.13 Km/h	0.87	16.95 %
2	8 Km/h	8.70 Km/h	0.70	8.04 %
3	16 Km/h	16.75 Km/h	0.75	4.47 %
4	12 Km/h	11.12 Km/h	0.88	7.91 %
5	21 Km/h	20.11 Km/h	0.89	4.42 %
6	14 Km/h	13.15 Km/h	0.85	6.46 %
7	10 Km/h	9.35 Km/h	0.65	6.95 %
8	18 Km/h	17.71 Km/h	0.29	1.63 %
9	21 Km/h	20.10 Km/h	0.90	4.47 %
10	24 Km/h	24.93 Km/h	0.93	3.73 %
Jumlah Error				65.03%
Rata -rata Error				6.50 %

$$\text{Rata-rata Error} = \frac{\text{jumlah kesalahan}}{\text{Banyaknya pengujian}} \times 100 = \frac{65.03}{10} \times 100 = 6.50\% \quad (24)$$

$$\text{Error} = \left(\frac{\text{Sistem} - \text{Manual}}{\text{Manual}} \right) \times 100 = \frac{6 - 5.14}{5.14} \times 100 = 16.95\% \quad (25)$$

$$\text{Ketepatan Akurasi} = 100\% - \text{Rata-rata error} = 100 - 6.50\% = 93.50\% \quad (26)$$



Gambar 24. Pengujian Kecepatan

5.5. Pengujian Alat Dengan Beberapa Surveyor

Untuk memenuhi tujuan dua dari penelitian ini yaitu menganalisa pengaruh dari alat yang dibuat terhadap respon pengendara sepeda pada jarak berkendaranya. Kami memberikan kesempatan pada lima orang untuk mencoba mengendarai sepeda yang sudah dipasang alat ini dan kemudian menanyakan pertanyaan tentang bagaimana respon yang mereka berikan terhadap jarak berkendaranya setelah diberi peringatan oleh alat ini.

Tabel 17. Pengujian dengan Beberapa surveyor

No	Nama	Jumlah kondisi aman waspada dan bahaya dalam jarak tempuh 1 Km						Respon pengendara terhadap jarak dan komentar tentang kinerja alat
		Aman		Waspada		Bahaya		
		Terbaca	Tidak	Terbaca	Tidak	Terbaca	Tidak	
1	Robi	6	0	3	2	4	2	Saat alat ini berbunyi dan menunjukkan kondisi bahaya dan waspada, saya langsung mengurangi kecepatan agar jarak dan kecepatan tersebut tidak bahaya atau waspada lagi. Alat ini cukup membantu dalam memberi peringatan saat bersepeda, meskipun untuk sepeda dan sepeda pendeteksiannya cukup sulit.
2	Aldi	3	0	5	1	3	0	Saya beberapa kali terkejut oleh alat ini karena gaya bersepeda saya yang santai dan mengamati kendaraan sekitar, Saya

								terkadang langsung melakukan rem saat alatnya berbunyi. Untuk kondisi bersepeda yang santai dan tidak terlalu fokus di jalan, alat ini membantu sekali untuk memperingatkan kondisi bahaya yang ada
3	Oggi	4	0	4	2	5	1	Saya bersepeda dengan fokus sehingga perkiraan kondisi bahaya dan waspada saya dengan alat ini cukup sesuai, beberapa kali saya memprediksi kondisi akan bahaya dengan kecepatan dan jarak bersepeda saya saat itu. Saya tentu saja langsung melakukan pengereman saat kondisi bahaya terdeteksi.
4	Andre	4	0	2	1	3	2	Saat alat ini berbunyi saya tentunya langsung mengurangi kecepatan sehingga kondisi bahaya tadi langsung hilang. Untuk memperingatkan kita tentu alat ini cukup berguna tapi bagi yang sudah handal bersepeda dan memiliki intuisi terhadap keadaan-keadaan yang membahayakan alat ini tidak begitu berpengaruh jika pengendara sepeda itu fokus.
5	Soni	3	0	3	2	4	2	Alat ini membantu saya melakukan pengereman dengan tepat kondisi bahaya terjadi, meskipun sempat eberpa kali alat inni tidak mendeteksi kondisi yang seharusnya berbahaya.



Gambar 25. Pengujian dengan Surveyor

Bab 5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan pembahasan dari penelitian yang dilakukan, penulis bisa menarik kesimpulan bahwa “Penerapan *fuzzy logic* pada pendeteksian jarak aman berkendara sepeda” berhasil dilakukan. Dengan beberapa poin kesimpulan.

1. Sensor LiDAR sebagai pendeteksi jarak bekerja dengan baik saat mendapati sasaran ukur berupa material ataupun benda dengan permukaan luas, dalam kasus ini adalah mobil. Tingkat pendeteksian pada mobil lebih mudah dibandingkan kendaraan yang ukurannya kecil. Adanya debu pada sensor kerap mempengaruhi pendeteksian. Dari data yang diambil nilai akurasi yang didapat adalah 96.06%.
2. Sensor *magnetic hall effect* berfungsi sangat baik sebagai pendeteksi kecepatan dengan pendekatan RPM (Rotasi Per Menit). Namun harus dipastikan bahwa jarak antara sensor dan magnet di pasang pada jarak 3 mm, sehingga sensor mampu bekerja dengan baik. Akurasi sensor *magnetic hall effect* ini jika kita lihat pada data pengujian adalah sebesar 93.50 %.
3. Sistem fuzzy pada alat bekerja dengan baik, hal ini bisa kita lihat dari kesesuaian output yang berupa alarm aman, waspada, dan bahaya terhadap jarak dan kecepatan berkendara yang benar benar bekerja sesuai dengan sembilan *rule evaluation* yang dibuat.
4. Pengujian dengan para surveyor untuk melihat respon pengendara terhadap jarak berkendaranya menghasilkan kesimpulan yang cukup baik, dimana para pengendara bereaksi dengan mengerem dan menjaga jarak berkendaranya sesaat setelah alarm berbunyi, beberapa dari surveyor menilai alat ini sangat membantu bagi orang-orang yang baru belajar membawa sepeda. Meskipun beberapa kali alat ini tidak mendeteksi keadaan-keadaan tersebut.
5. Dari Analisa yang dilakukan terhadap cara kerja dari alat pendeteksi jarak aman yang terimplementasi *fuzzy logic* berbasis mikrokontroler ini, bisa disimpulkan bahwa sistem yang menggabungkan sensor *Tf Luna LiDAR* sebagai jarak, Sensor *Hall effect* sebagai kecepatan dan *microcontroller* yang ditanamkan sistem *fuzzy* ini mampu membuat keputusan untuk menentukan tingkat kondisi berkendara yaitu aman, waspada dan bahaya berbasis dengan tingkat kedekatan objek yang cukup efektif.
6. Penelitian dengan judul “Penerapan Fuzzy Logic Untuk Mendeteksi Jarak Aman Berkendara Sepeda” berdasarkan analisa yang sudah dilakukan sebelumnya menghasilkan kesimpulan bahwa penerapan *fuzzy logic* pada alat ini berhasil dilakukan.

5.2. Saran

Beberapa saran yang bisa penulis berikan jika suatu saat ada yang berniat melakukan penelitian serupa ataupun mengembangkan ide ini.

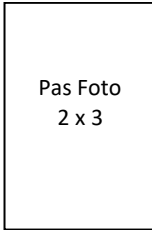
1. Agar sensor LiDAR lebih akurat maka pastikan pemasangan sensor pada sepeda berada dalam posisi tegak lurus 90 derajat terhadap tanah.
2. Pastikan penginstallan sensor *Hall effect* benar-benar kuat untuk mencegah kerusakan.
3. Untuk peringatan secara visual yang lebih baik, gunakan *LCD* yang lebih besar dan lampu yang lebih terang.
4. Gunakan sensor jarak dengan jangkauan yang lebih jauh untuk perancangan membership function yang lebih baik.

Daftar Pustaka

- [1] R. Rahadiansyah, "Jumlah Sepeda Naik 10 kali lipat, Kecelakaan masih tinggi", *detikOto*. Accessed: Jul. 19, 2025. [Online]. Available: <https://oto.detik.com/berita/d-5258089/jumlah-pesepeda-naik-hampir-10-kali-lipat-kecelakaan-masih-tinggi>
- [2] R. Wiranata, "Lonjakan Kecelakaan Lalu Lintas di Batam, Polresta Barelang Catat 776 Kasus di 2023," *Batamnews*.
- [3] M. Juliana Ritonga, "SISTEM PERINGATAN JARAK AMAN SEPEDA MOTOR," *Skripsi*, p. 17, Nov. 2019, Accessed: Dec. 08, 2024. [Online]. Available: <http://repository.uinsu.ac.id/9924/2/%28Skripsi%29%20Maya%20Juliana%20Ritonga-71153013.pdf>
- [4] N. K. Sa'diyah, "Kajian Terhadap Faktor-Faktor Penyebab Kecelakaan Lalu Lintas Dalam Upaya Perbaikan Pencegahan Kecelakaan Lalu Lintas," *Perspektif*, vol. 22, pp. 241–244, Sep. 2017, Accessed: Dec. 02, 2024. [Online]. Available: <https://erepository.uwks.ac.id/9672/>
- [5] Herfando, "ANALISIS FAKTOR PENYEBAB KECELAKAAN LALU LINTAS DAN AUDIT KESELAMATAN JALAN RAYA BUKIT TINGGI-MEDAN KM 65 KUMPULAN KABUPATEN PASAMAN," *Skripsi Universitas Muhammadiyah Sumbar*, p. 4, Sep. 2022, Accessed: Dec. 08, 2024. [Online]. Available: <http://eprints.umsb.ac.id/881/1/HERFANDO%20%28%20181000222201165%29.pdf>
- [6] A. J Rindengan and Y. A R Langi, *Sistem-Fuzzy*. CV. Patra Media Grafindo Bandung, 2019.
- [7] N. Khairina, S. Kom, and M. Kom, "LOGIKA FUZZY," 2019. Accessed: Dec. 08, 2024. [Online]. Available: <https://khairina.blog.uma.ac.id/wp-content/uploads/sites/394/2019/09/BAB-1-LOGIKA-FUZZY.pdf>
- [8] A. D. Cahyani, D. Dewatama, and L. Kamajaya, "IMPLEMENTASI FUZZY LOGIC CONTROL PADA ALAT PENERING CENGKEH OTOMATIS," 2023, [Online]. Available: <https://jmi.rivierapublishing.id/index.php/rp>
- [9] Sutikno and I. Waspada, "PERBANDINGAN METODE DEFUZZIFIKASI SISTEM KENDALI LOGIKA FUZZY MODEL MAMDANI PADA MOTOR DC," 2011. Accessed: Dec. 08, 2024. [Online]. Available: <https://media.neliti.com/media/publications/112322-ID-perbandingan-metode-defuzzifikasi-sistem.pdf>
- [10] N. I. Assidqi, D. Astharini, and S. Hamid, "Light-Based Positioning System Using Arduino," *EXSACT-A*, vol. 1, p. 28, Dec. 2023, doi: 10.36722/exc.v1i1.2286.
- [11] R. Aisuwarya, M. Azmi Riyan, and R. Eka Putri, "Design of Bicycle's Speed Measurement System Using Hall Effect Sensor," in *Journal of Physics: Conference*

Series, Institute of Physics Publishing, Dec. 2019. doi: 10.1088/1742-6596/1339/1/012018.

Biodata



Nama : Al Firman Syah
TTL : Kerinci/20-Januari-1998
Agama : Islam
Alamat : Bida Asri 1 Blok D2 no1, Baloi Permai,
Batam Kota, Kota Batam
E-mail : alfirmansyah101@gmail.com
Pendidikan :
SD : SDN 63/III Belui
SMP : SMPN 15 Kerinci
SMA : SMKN 2 Sungai Penuh

Lampiran A. Mikrokontroler program

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <HardwareSerial.h>

// Please adjust these values for the speedometer
#define METRIC // Tire unit METRIC or IMPERIAL
#define TIRE_WITDH 80 // unit based on tire type (Metric or Imperial)
#define TIRE_HEIGHT 90 // unit based on tire type (Metric or Imperial)
#define VELG_DIAMETER 17 // unit INCH

//convert all value to meters (calculate radius)
#ifndef METRIC
#define WHEEL_RADIUS_M ((VELG_DIAMETER + (TIRE_HEIGHT / 25.4)) / 2) /
39.37
#endif

#ifndef IMPERIAL
#define HEIGHT_BASED_RATIO (TIRE_HEIGHT * TIRE_WITDH) / 100
#define WHEEL_RADIUS_M ((VELG_DIAMETER + HEIGHT_BASED_RATIO) / 2) /
39.37
#endif

// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 16, 2); // 0x27,0x3f

//Front sensor
#define SLAVE_ADDRESS 0x10 // I2C Indicates the address of the secondary
device
#define DATA_LENGTH 9 // data length
#define SCL 21
#define SDA 22

//LED & BUZZER
#define GRN_LED_PIN 19
#define YEL_LED_PIN 18
#define RED_LED_PIN 4
#define BUZZER_PIN 23
#define BAHAYA_BEEP 0.2 // unit second
```

```

#define WASPADA_BEEP 0.7 // unit second

//Hall Effect Sensor
#define HALL_PIN 13

bool booting = false;
unsigned long duration;
unsigned long rpm = 0;
static long kph = 0;
static long zeroMillis;
bool oneCycle;

#define SIMULATION_VALUE 0 //1:simulation value sensor 0:read value from
sensor
#if SIMULATION_VALUE
int speed_kmh = 50; // km/h
int distance_front = 800; // cm
#else
int distance_front, speed_kmh;
#endif

// -----
// FUZZY PARAMETERS
float uDekat_F, uSedang_F, uJauh_F, // Jarak Depan
      uLambat, uSedang, uCepat, // Kecepatan
      miuJarak_F[3], miuKecepatan[3], min_val[9], crisp_rule_peringatan,
      out_fuzzyPeringatan;

volatile float warning = 0.00;

#define STOP 0
#define AMAN 1
#define WASPADA 2
#define BAHAYA 3

// Use only core 1
#if CONFIG_FREERTOS_UNICORE
static const BaseType_t app_cpu = 0;
#else
static const BaseType_t app_cpu = 1;
#endif

```

```

void i2cTask(void *parameter);
void fuzzyTask(void *parameter);
void buzzerLedTask(void *parameter);

void setup() {
  Serial.begin(115200); // Initializing serial port
  pinMode(HALL_PIN, INPUT);
  booting = false;

  xTaskCreatePinnedToCore(
    i2cTask,
    "i2c Task",
    4096,
    NULL,
    1,
    NULL,
    app_cpu);

  xTaskCreatePinnedToCore(
    fuzzyTask,
    "fuzzyTask",
    4096,
    NULL,
    1,
    NULL,
    app_cpu);

  xTaskCreatePinnedToCore(
    buzzerLedTask,
    "buzzerLedTask",
    4096,
    NULL,
    1,
    NULL,
    app_cpu);

  while (!booting) { delay(5); }
}

void loop() {

```

```

if (!SIMULATION_VALUE) kmhSpeed();
serial_debug();
}

void kmhSpeed() {
duration = pulseIn(HALL_PIN, RISING, 500000);

if (duration > 0) {
oneCycle = false;
rpm = 60000.0 / duration * 1000;
kph = (2 * PI * WHEEL_RADIUS_M * (double)rpm * 60) / 1000;
} else {
if (millis() - zeroMillis >= 10000 && !oneCycle) {
zeroMillis = millis();
oneCycle = true;
rpm = 0;
kph = 0;
}
}

speed_kmh = kph;
}

void i2cTask(void *parameter) {
Wire.begin(SCL, SDA);
lcd.init();
lcd.backlight();
for (byte i = 0; i <= 16; i++) {
lcd.setCursor(i, 0);
lcd.write(0xff);
lcd.setCursor(16 - i, 1);
lcd.write(0xff);
delay(20);
}
buzz(150, 0);
buzz(0, 150);
buzz(150, 0);
buzz(0, 150);

lcd.clear();
lcd.setCursor(0, 0);

```

```

lcd.print("SAFE DISTANCE ");
lcd.setCursor(0, 1);
lcd.print(" WARNING SYS.");
delay(3000);
lcd.clear();
booting = true;

while (1) {
  unsigned char buf[] = {0x5A, 0x05, 0x00, 0x01, 0x60};
  Wire.beginTransmission(SLAVE_ADDRESS);
  Wire.write(buf, 5);
  Wire.endTransmission();
  Wire.requestFrom(SLAVE_ADDRESS, DATA_LENGTH);
  uint8_t data[DATA_LENGTH] = {0};
  int index = 0;
  while (Wire.available() > 0 && index < DATA_LENGTH) {
    data[index++] = Wire.read();
    if (index == DATA_LENGTH) {
      if (!SIMULATION_VALUE) {
        distance_front = data[2] + (data[3] * 256);
        if (distance_front >= 800 || distance_front < 1) distance_front = 800;
      }
    }
  }

  char SPEED[20], DISTANCE[50];
  sprintf(SPEED, "%03d km/h", speed_kmh);
  sprintf(DISTANCE, "%03dCM", distance_front);

  lcd.setCursor(0, 0);
  lcd.print(SPEED);
  lcd.setCursor(9, 0);
  if (almostEqual(warning, STOP)) lcd.print("-----");
  else if (almostEqual(warning, AMAN)) lcd.print(" AMAN");
  else if (almostEqual(warning, WASPADA)) lcd.print("WASPADA");
  else if (almostEqual(warning, BAHAYA)) lcd.print(" BAHAYA");
  printCenter(1, DISTANCE);
  delay(100);
}
}

```

```

void fuzzyTask(void *parameter) {
    while (!booting) { delay(5); }
    while (1) {
        fuzzifikasi();
        defuzzifikasi();
        warning = out_fuzzyPeringatan;
    }
}

void buzzerLedTask(void *parameters) {
    pinMode(GRN_LED_PIN, OUTPUT);
    pinMode(YEL_LED_PIN, OUTPUT);
    pinMode(RED_LED_PIN, OUTPUT);
    pinMode(BUZZER_PIN, OUTPUT);
    while (!booting) { delay(5); }
    while (true) {
        if (almostEqual(warning, STOP) || almostEqual(warning, AMAN)) {
            digitalWrite(GRN_LED_PIN, HIGH);
            digitalWrite(YEL_LED_PIN, LOW);
            digitalWrite(RED_LED_PIN, LOW);
            digitalWrite(BUZZER_PIN, LOW);
        } else if (almostEqual(warning, WASPADA)) {
            digitalWrite(GRN_LED_PIN, LOW);
            digitalWrite(RED_LED_PIN, LOW);
            digitalWrite(YEL_LED_PIN, HIGH);
            digitalWrite(BUZZER_PIN, HIGH);
            vTaskDelay((WASPADA_BEEP * 100) / portTICK_PERIOD_MS);
            digitalWrite(YEL_LED_PIN, LOW);
            digitalWrite(BUZZER_PIN, LOW);
            vTaskDelay((WASPADA_BEEP * 900) / portTICK_PERIOD_MS);
        } else if (almostEqual(warning, BAHAYA)) {
            digitalWrite(GRN_LED_PIN, LOW);
            digitalWrite(YEL_LED_PIN, LOW);
            digitalWrite(RED_LED_PIN, HIGH);
            digitalWrite(BUZZER_PIN, HIGH);
            vTaskDelay((BAHAYA_BEEP * 100) / portTICK_PERIOD_MS);
            digitalWrite(RED_LED_PIN, LOW);
            digitalWrite(BUZZER_PIN, LOW);
            vTaskDelay((BAHAYA_BEEP * 900) / portTICK_PERIOD_MS);
        }
    }
}

```

```

}

// Fuzzification curves
#define Kurva_Turun 1
#define Kurva_Segitiga 2
#define Kurva_Trapesium 3
#define Kurva_Naik 4

float hitung_member(int pilih, float input, float A, float B, float C, float D) {
    switch (pilih) {
        case Kurva_Turun:
            if (input >= A && input <= B) return 1;
            else if (input > B && input < C) return (C - input) / (C - B);
            else if (input >= C) return 0;
            break;
        case Kurva_Segitiga:
            if (input <= A || input >= C) return 0;
            else if (input > A && input < B) return (input - A) / (B - A);
            else if (input > B && input < C) return (C - input) / (C - B);
            else if (input == B) return 1;
            break;
        case Kurva_Trapesium:
            if (input <= A || input >= D) return 0;
            else if (input > A && input <= B) return (input - A) / (B - A);
            else if (input > C && input <= D) return (D - input) / (D - C);
            else if (input > B && input <= C) return 1;
            break;
        case Kurva_Naik:
            if (input <= A) return 0;
            else if (input > A && input < B) return (input - A) / (B - A);
            else if (input >= B) return 1;
            break;
    }
    return 0;
}

void fuzzifikasi() {
    //Kecepatan
    uLambat = hitung_member(Kurva_Turun, speed_kmh, 0, 10, 60, 0);
    uSedang = hitung_member(Kurva_Trapesium, speed_kmh, 40, 65, 85, 110);
    uCepat = hitung_member(Kurva_Naik, speed_kmh, 90, 150, 150, 0);
}

```

```

//Jarak Depan
uDekat_F = hitung_member(Kurva_Turun, distance_front, 0, 0, 400, 0);
uSedang_F = hitung_member(Kurva_Trapesium, distance_front, 300, 450,
500, 700);
uJauh_F = hitung_member(Kurva_Naik, distance_front, 600, 800, 800, 0);
}

// [kecepatan][jarak depan] fuzzy rules 3x3=9
int rule_set[3][3] = {
    {BAHAYA, WASPADA, AMAN}, // Lambat
    {BAHAYA, WASPADA, AMAN}, // Sedang
    {BAHAYA, WASPADA, AMAN} // Cepat
};

int max_jarak_f_idx = 0, max_kecepatan_idx = 0;

void defuzzifikasi() {
    float numerator = 0, denominator = 0, centre_of_area = 0;

    int indx = 0;

    // arrays for membership values
    float Data_arrKec[3] = { uLambat, uSedang, uCepat };
    float Data_arrJar_F[3] = { uDekat_F, uSedang_F, uJauh_F };

    for (int i = 0; i < 3; i++) { // kecepatan
        for (int j = 0; j < 3; j++) { // jarak depan

            miuKecepatan[i] = Data_arrKec[i];
            miuJarak_F[j] = Data_arrJar_F[j];

            // Cari max value dan indeksnya
            if (miuKecepatan[i] > out_fuzzyPeringatan) {
                out_fuzzyPeringatan = miuKecepatan[i];
                max_kecepatan_idx = i;
            }
            if (miuJarak_F[j] > out_fuzzyPeringatan) {
                out_fuzzyPeringatan = miuJarak_F[j];
                max_jarak_f_idx = j;
            }
        }
    }
}

```

```

int ruleVal = rule_set[i][j];

min_val[indx] = min(miuKecepatan[i], miuJarak_F[j]);
numerator += min_val[indx] * ruleVal;
denominator += min_val[indx];
indx++;
}
}
if (speed_kmh > 0 && denominator != 0) {
    centre_of_area = numerator / denominator;
    out_fuzzyPeringatan = centre_of_area;
} else {
    out_fuzzyPeringatan = STOP;
}
crisp_rule_peringatan = rule_set[max_kecepatan_idx][max_jarak_f_idx];
}

void printCenter(int locationY, String text) {
    int centerPoint = (16 - text.length()) / 2;
    lcd.setCursor(centerPoint, locationY);
    lcd.print(text);
}

void buzzz(int tim_on, int tim_off) {
    digitalWrite(BUZZER_PIN, HIGH);
    delay(tim_on);
    digitalWrite(BUZZER_PIN, LOW);
    delay(tim_off);
}

boolean almostEqual(float a, float b) {
    const float DELTA = .00001;
    if (a == 0) return fabs(b) <= DELTA;
    if (b == 0) return fabs(a) <= DELTA;
    return fabs((a - b) / max(fabs(a), fabs(b))) <= DELTA;
}

// Serial debug info
void serial_debug() {
    Serial.print("Speed: ");

```

```

Serial.print(speed_kmh);
Serial.print("km/h | Lambat:");
Serial.print(uLambat);
Serial.print(" Sedang:");
Serial.print(uSedang);
Serial.print(" Cepat:");
Serial.print(uCepat);
String member_kecepatan[] = {"Lambat", "Sedang", "Cepat"};
for (byte i = 0; i < 3; i++) {
  if (out_fuzzyPeringatan == miuKecepatan[i]) {
    Serial.print(" [" + member_kecepatan[i] + "]");
  }
}

Serial.print(" | Jarak Depan: ");
Serial.print(distance_front);
Serial.print("cm | Dekat:");
Serial.print(uDekat_F);
Serial.print(" Sedang:");
Serial.print(uSedang_F);
Serial.print(" Jauh:");
Serial.print(uJauh_F);
String member_jarak_depan[] = {"Dekat", "Sedang", "Jauh"};
for (byte i = 0; i < 3; i++) {
  if (out_fuzzyPeringatan == miuJarak_F[i]) {
    Serial.print(" [" + member_jarak_depan[i] + "]");
  }
}

Serial.print(" | Lvl Peringatan: ");
Serial.print(out_fuzzyPeringatan);
if (out_fuzzyPeringatan != STOP) {
  int rule_set_warning[] = {AMAN, WASPADA, BAHAYA};
  String member_output_warn[] = {"Aman", "Waspada", "Bahaya"};
  for (byte i = 0; i < 3; i++) {
    if (crisp_rule_peringatan == rule_set_warning[i]) {
      Serial.print(" [" + member_output_warn[i] + "]");
    }
  }
} else {
  Serial.print(" [STOP]");
}

```

```
}  
Serial.println();  
}
```