

Rancang dan Bangun Prototipe Lampu Lalu Lintas Berdasarkan Kecepatan Pada Model Kendaraan Empat Roda Dengan Menggunakan Sistem Algoritma Logika Fuzzy

Regina Chelinia Erianda Putri¹, Oktavianus Ardhian Nugroho², Ignasius Tegar Adiyanto³

¹Program Studi Teknik Elektro, Fakultas Sains dan Teknologi, Universitas Sanata Dharma Yogyakarta
Jl. Paingan, Krodan, Maguwoharjo, Kec. Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta, Indonesia

²Program Studi Mesin Industri, Politeknik Industri ATMI
Jl. Kampus Hijau No. 3 Kawasan Jababeka Education Park, Jl. Raya Lemahabang, Simpangan, Kec. Cikarang Utara, Kabupaten Bekasi, Jawa Barat 17520, Indonesia

³PT. Multipanel Intermitra Mandiri
Jl. Industri Utara 4, Blok SS 6, Blk. C, Mekarmukti, Kec. Cikarang Utara, Kabupaten Bekasi, Jawa Barat 17148

Corresponding Author: ardhianatmi@gmail.com

Riwayat Artikel

Diserahkan: 17 Maret 2025
Direvisi: 5 Mei 2025
Diterima: 16 Mei 2025
Dipublikasi: 30 Mei 2025

Abstrak

Secara umum, sistem pengaturan lampu lalu-lintas di Indonesia hanya mengatur berdasarkan waktu yang tetap. Masalah akan muncul ketika terjadi akumulasi kendaraan di satu sisi. Dari hasil observasi lapangan salah satu penyebab penumpukan ini terjadi dikarenakan pengaturan nyala lampu yang tidak menyesuaikan kecepatan kendaraan yang dihubungkan dengan penetapan durasi lampu hijau. Pada penelitian ini dibuat sebuah sistem purwarupa *plant traffic light control* dengan menggunakan Sistem logika *fuzzy* yang diterapkan pada *microcontroller* ATmega32. Dengan membandingkan kecepatan mobil di setiap sisi, diperoleh data dari algoritma yang bisa digunakan untuk mengatur waktu lampu lalu lintas di setiap arah. Purwarupa yang dibuat berfungsi secara adaptif. Dari hasil penelitian ini menggunakan purwarupa didapatkan bahwa dengan mengatur durasi lampu hijau pada sistem lampu lalu-lintas didapatkan tingkat persentase keberhasilan sebesar 99.97%. Di mana jika kecepatan kendaraan tinggi nilai durasi lampu hijau menjadi cepat dan begitu pula sebaliknya.

Kata kunci: kecepatan, lampu lalu lintas, *fuzzy logic*, ATmega32.

Abstract

Generally, traffic light control systems in Indonesia only regulate based on fixed time. Problems will arise when there is an accumulation of vehicles on one side. From the results of field observations, one of the causes of this buildup is due to the setting of the lights that do not adjust the speed of the vehicle which is connected to the determination of the duration of the green light. In this study, a prototype system of *plant traffic light control* was created using a *fuzzy logic* system applied to the ATmega32 microcontroller. By comparing the speed of the car on each side, data was obtained from the algorithm that can be used to regulate the time of the traffic lights in each direction. The prototype that was created functions adaptively. From the results of this study using the prototype, it was found that by regulating the duration of the green light in the traffic light system, a success rate of 99.97% was obtained. Where if the vehicle speed is high, the green light duration value becomes fast and vice versa.

Keywords: speed, traffic light, *fuzzy logic* algorithm, ATmega32.

1. Pendahuluan

Lampu lalu lintas adalah alat yang digunakan untuk mengatur kendaraan di persimpangan jalan, zebra *cross*, dan lokasi lalu lintas lainnya. Alat ini sendiri biasa digunakan untuk mengurai arus lalu lintas supaya tidak terjadi kemacetan. Pada umumnya kemacetan sering muncul pada daerah dengan jumlah angka kendaraan yang tinggi, jumlah yang tinggi ini menyebabkan volume kendaraan menjadi tinggi dan lampu lalu lintas yang seharusnya sebagai pengurai kemacetan malah berfungsi sebaliknya sebagai penyebab kemacetan. [1]. Dari permasalahan tersebut perlu dibuat sebuah sistem yang dapat mengurai arus lalu lintas dengan baik. Pada saat ini, pengaturan lampu lalu lintas yang umum digunakan adalah sistem lampu lalu lintas yang memiliki lampu merah, kuning, dan hijau, dan waktu nyalanya telah ditentukan. Sistem konvensional ini dirasa kurang sesuai dengan perkembangan teknologi dan kondisi saat ini [2-4]. Perkembangan teknologi pada sistem lampu lalu lintas saat ini semakin maju dan berkembang di sisi lain dengan terus meningkatnya jumlah kendaraan maka kemungkinan timbulnya kemacetan akan sering terlihat terutama pada persimpangan jalan yang memiliki lampu lalu lintas sebagai sarana pengatur jalan. Pada penelitian yang pernah dilakukan oleh Putra [5] dari penelitiannya berdasarkan pengamatan yang telah dilakukan olehnya, lampu lalu lintas memiliki peranan yang sangat krusial di persimpangan jalan karena intensitas kendaraan di setiap jalan berbeda-beda. Ini membuat arus kendaraan dari satu arah bisa sangat banyak, sementara dari arah lain bisa sangat sedikit. Berdasarkan permasalahan tersebut, diperlukan sebuah perangkat yang mampu mengukur kecepatan mobil agar bisa mengatur waktu sistem lampu lalu lintas sesuai dengan keadaan lalu lintas pada persimpangan yang ada alat pengatur lalu lintas berupa lampu lalu lintas. Penelitian sebelumnya sudah dilakukan oleh Pamo G [6] misalnya

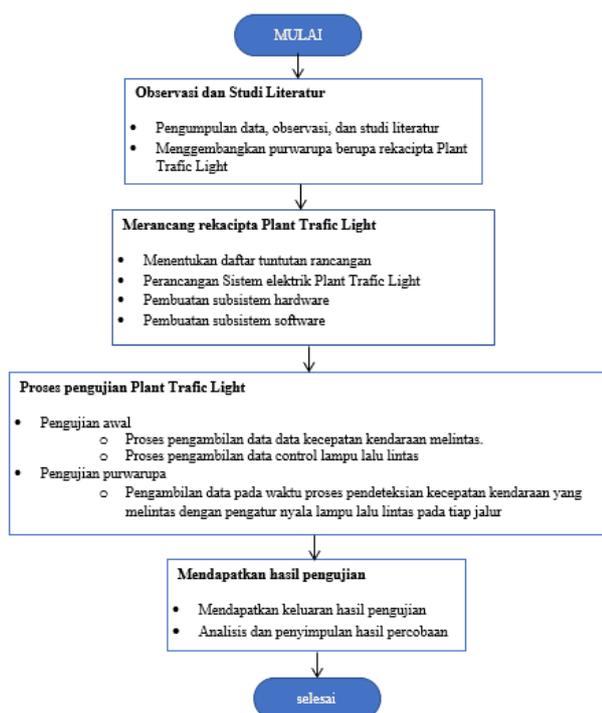
Perancangan sistem *Traffic Light* menggunakan PLC (*Programmeble Logic Controller*) sebagai pengontrol. Tetapi dalam penentuan lamanya waktu lampu hijau menyala masih berdasarkan luasan antrean kendaraan yang berhenti di tiap-tiap jalur sehingga sistem ini masih belum efektif [6]. Penelitian lain oleh Harris Cesardarmantya [7] misalnya dalam penelitiannya membuat sebuah purwarupa lampu lalu lintas dengan PLC berdasarkan panjang antrean kendaraan. Selain penggunaan sensor, deteksi pelanggaran lampu lalu lintas masih belum kurang yaitu pada metode pengenalan pola dalam pengidentifikasian kendaraan masih belum detail dilakukan. Hal itu menyebabkan pengaturan nyala lampu lalu lintas yang dirancang masih jauh dari fleksibel terutama langkah dalam menguraikan arus lalu lintas [7].

Tujuan dari penelitian ini adalah membuat sebuah purwarupa sistem pengaturan lampu lalu lintas berdasarkan kecepatan kendaraan secara *real time*. Manfaatnya adalah mengatasi permasalahan yang terjadi pada persimpangan jalan yang memiliki arus kendaraan tidak merata. Dalam studi ini, dalam mengatasi hal-hal di atas maka dibuatlah sebuah sistem pengaturan lampu lalu lintas akan dirancang sebuah purwarupanya yang didasarkan pada kecepatan kendaraan [8]. Dengan menggunakan algoritma *fuzzy logic* durasi lampu lalu lintas dapat ditentukan secara langsung. Nantinya, diharapkan alat ini bisa digunakan untuk mengatur lampu lalu lintas di persimpangan agar arus lalu lintas menjadi lebih merata dan dapat otomatis beradaptasi dengan kondisi lalu lintas yang ada pada saat yang sama. Penggunaan sistem pengaturan lalu lintas dengan metode *fuzzy logic* juga pernah dilakukan oleh Regina C.E.P [9] misalnya dalam penelitiannya menggunakan algoritma *fuzzy logic* dalam pengaturan durasi lampu hijau. Tetapi menggunakan jumlah kendaraan sebagai pengatur durasi lampu menyebabkan

pengiriman data yang dikirimkan menjadi kurang efektif sehingga perlu ditambahkan komunikasi dengan menggunakan I2C. ada beberapa batasan dalam penelitian ini yang ditetapkan. Pertama persimpangan jalan disimulasikan berupa bentuk pertigaan. Kedua pada pertigaan setiap lajunya dibuat dengan 2 arah berbeda, di mana satu lajur hanya bisa dilalui oleh satu kendaraan. Ukuran purwarupa pertigaan jalan adalah 220cm x 120cm. Dan yang ketiga *fuzzy logic* merupakan algoritma yang diterapkan dalam pembuatan sistem ini.

2. Metode

Cara yang diterapkan untuk mengembangkan sistem rekayasa pengaturan nyala lampu lalu lintas yang didasarkan pada kecepatan kendaraan menggunakan algoritma *fuzzy logic*. Algoritma ini adalah peningkatan dari logika *boolean* yang mengenalkan konsep kebenaran sebagian. Sementara itu *fuzzy logic* menggantikan kebenaran *boolean* dengan tingkat kebenaran [10, 11]. Gambar 1 menunjukkan alur *flowchart* penelitian yang dilakukan dari perancangan sampai dengan uji coba purwarupa yang dibuat.



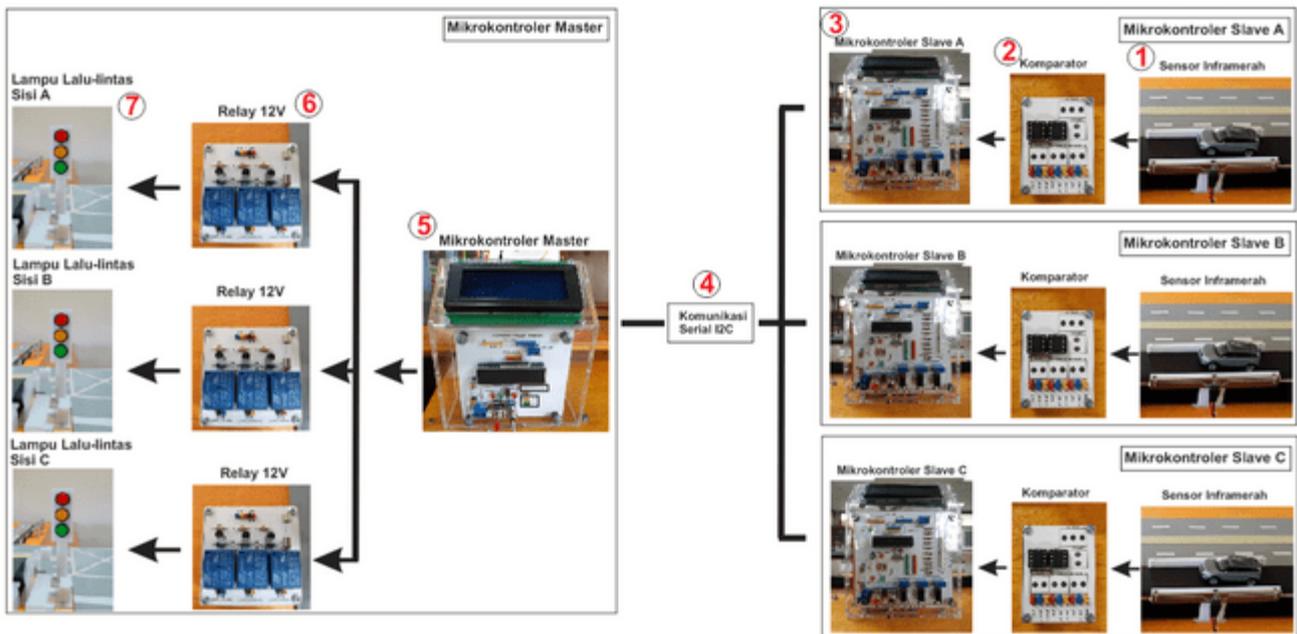
Gambar 1. *Flowchart* Alur pengerjaan

Setelah tahap observasi, langkah selanjutnya adalah membuat sub sistem *hardware* [12], yang termasuk merakit rangkaian komparator untuk sensor inframerah dan menggunakan transistor sebagai sakelar untuk menyalakan lampu lalu lintas. Kemudian, pembuatan sub sistem *software* dilakukan, yang mencakup pemrograman. Pada fase ini, peneliti memanfaatkan *microcontroller* ATmega8 [13] sebagai *slave* dan *microcontroller* ATmega32 [14] sebagai master dan pengambil keputusan. Bahasa pemrograman C [15] digunakan untuk memasukkan keputusan ke dalam *microcontroller* dengan bantuan *software code vision AVR* [16]. Tahap ketiga melibatkan pengambilan data, di mana data kecepatan kendaraan yang lewat diobservasi bersamaan dengan pengendalian lampu lalu lintas, pada tahap ini sebuah purwarupa berupa *plant traffic light* dibuat untuk menyimulasikan percobaan. Tahap terakhir adalah analisis dan penyimpulan hasil percobaan, yang meliputi analisis data dengan mendeteksi kecepatan kendaraan yang lewat dan pengaturan waktu nyala lampu lalu lintas untuk setiap jalur. Penyimpulan data dilakukan dengan membandingkan hasil yang tercatat dengan durasi nyala lampu lalu lintas.

Secara umum, sistem desain *plant traffic light* ditampilkan dalam Gambar 2. Pada gambar 2 terdapat bagian seperti 1. Sensor inframerah, 2. Komparator, 3. *Microcontroller* slave, 4. Komunikasi *serial* I2C, 5. *Microcontroller* master, 6. *Relay* 12 Volt, dan 7. Tiang *traffic light*. Penjelasan *microcontroller* slave A, B, dan C, pada *plant traffic light* berisi sensor inframerah, sensor ini terpasang di jalur konveyor dengan tujuan untuk mendeteksi kecepatan dan jumlah kendaraan. Selanjutnya, sinyal dari sensor inframerah akan dianalisis untuk menghasilkan nilai *output* yang akan digunakan untuk interupsi dalam *microcontroller*. Ketika sensor inframerah ini menangkap keberadaan kendaraan, *microcontroller* akan menghitung kecepatan

kendaraan tersebut dan menyimpan data itu ke dalam memori *microcontroller*. *Microcontroller* juga dilengkapi dengan layar LCD yang berfungsi untuk menunjukkan kecepatan

sebuah tiang lampu lalu-lintas yang akan menyalakan lampu merah, kuning dan hijau. Sistem akan menyala sesuai instruksi yang dijalankan oleh algoritma fuzzy logic yang



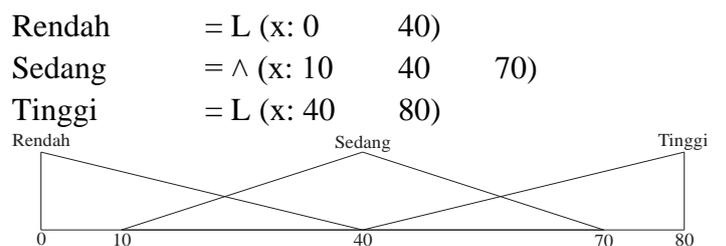
Gambar 2. *Plant traffic light sistem*

kendaraan saat melewati sensor inframerah. Lalu selanjutnya, kecepatan yang didapat akan dikirim ke *microcontroller* utama menggunakan komunikasi I2C [17]. Di dalam *microcontroller* utama, informasi tentang kecepatan kendaraan dari sisi A, B, dan C akan menjadi *input* untuk algoritma logika *fuzzy*. Algoritma ini berfungsi untuk menentukan lamanya lampu hijau menyala di setiap sisi jalan. Jika kendaraan bergerak cepat, lamanya lampu hijau akan diperpendek, dan sebaliknya. Lampu hijau akan menyala secara adaptif setelah periode *all red* yang terjadi sesaat setelah lampu hijau berakhir. Pada setiap situasi *all red*, sistem di *microcontroller* utama akan menjalankan kembali algoritma logika *fuzzy* berdasarkan kecepatan kendaraan yang diterima dari *microcontroller slave* [18], sehingga sistem adaptif ini dapat menyesuaikan durasi lampu hijau dengan kecepatan kendaraan pada setiap sisi jalan. Setelah data kecepatan diolah dalam *microcontroller master*, selanjutnya durasi lampu hijau akan diimplementasikan dalam

diimplementasikan pada *microcontroller master*.

2.1. Proses *Fuzzifikasi*

Proses *Fuzzifikasi* adalah merupakan suatu proses perubahan variabel non-*Fuzzy* (*crisp*) ke dalam variabel *Fuzzy* [19, 20]. Gambar 3 menunjukkan bentuk *framework input X*, yang berasal dari salah satu *microcontroller slave*. Proses *framework input* yang berupa kecepatan kendaraan rendah, sedang dan tinggi. Pada proses ini bertujuan untuk mendapatkan nilai derajat keanggotaan tertinggi untuk diolah pada proses selanjutnya. Untuk nilai keanggotaan *Fuzzifikasi* adalah:



Gambar 3. *Framework Input X*

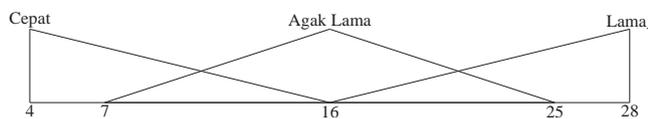
2.2. Basis Aturan dan Pengambilan Keputusan

Setelah melalui tahap *Fuzzifikasi*, selanjutnya akan dilakukan tahap pemilihan data *input* derajat keanggotaan yang memiliki nilai terkecil menggunakan fungsi minimum, yaitu: $\min(\text{Derajat_Keanggotaan A, Derajat_Keanggotaan B, Derajat_Keanggotaan C})$. Fungsi ini akan menghasilkan nilai *input* untuk derajat keanggotaan yang paling kecil dari derajat keanggotaan A, B, dan C. Nilai derajat keanggotaan yang minimum ini kemudian akan diproses dalam tahap *deFuzzifikasi*.

2.3. Proses DeFuzzifikasi

Pada Gambar 4 bentuk *framework output X* ditampilkan dengan menggunakan proses *deFuzzifikasi*. Pada proses *deFuzzifikasi* bagian tersebut mempunyai *framework output* berupa durasi nyala lampu hijau, yaitu cepat, agak lama dan lama. Untuk nilai keanggotaannya adalah:

$$\begin{aligned} \text{Cepat} &= L(x: 4 \quad 16) \\ \text{Agak Lama} &= \wedge(x: 7 \quad 16 \quad 25) \\ \text{Lama} &= L(x: 16 \quad 28) \end{aligned}$$



Gambar 4. *Framework Input X*

3. Hasil dan Pembahasan

Selanjutnya adalah proses pembuatan *plant traffic light*. Gambar 5 menunjukkan seluruh

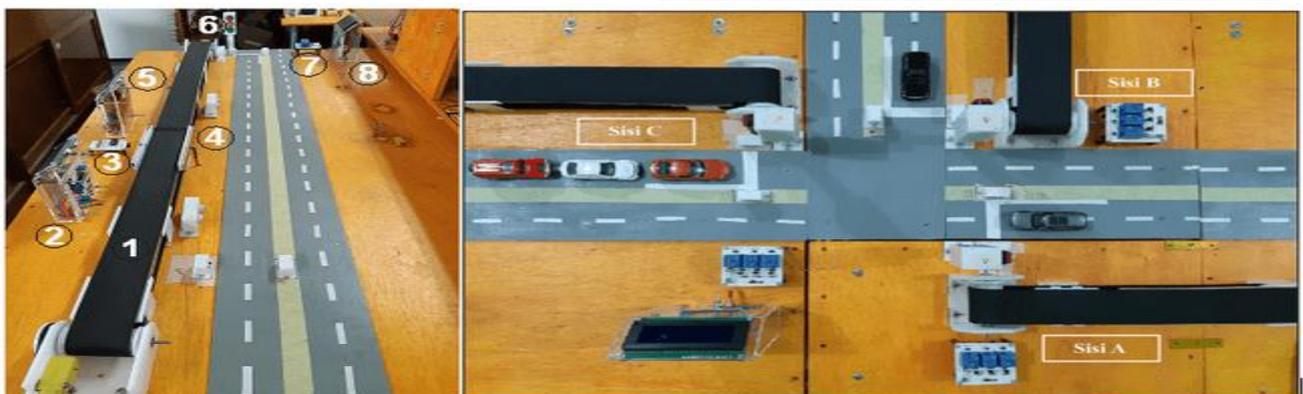
sistem *plant traffic light*. Purwarupa mobil pertama-tama diletakkan di atas jalur *konveyor*, lalu *konveyor* akan dioperasikan pada kecepatan yang diinginkan. Setelah mobil mulai bergerak, sensor inframerah yang berada di samping *konveyor* akan mendeteksi mobil tersebut, dan tegangan dari sensor inframerah tersebut akan dibandingkan dengan komparator. Jika *output* dari komparator menunjukkan angka 1, maka *microcontroller* yang berfungsi sebagai *slave* akan menghitung kecepatan mobil yang lewat dan menampilkan angka kecepatan tersebut di layar LCD. Setelah mendapatkan kecepatan kendaraan, data tersebut akan dikirim ke *microcontroller* master melalui komunikasi serial I2C. Informasi tentang kecepatan mobil di sisi A, sisi B, dan sisi C digunakan sebagai masukan untuk algoritma logika *fuzzy*. Ini bertujuan untuk menentukan seberapa lama lampu hijau akan menyala pada lampu lalu lintas di setiap sisi.

3.1. Pengujian Kecepatan Kendaraan oleh Alat Bantu Konveyor

Setelah dilakukan pengujian dan percobaan untuk alat bantu konveyor di uji dan di catat pada Tabel 1.

Tabel 1. Pengujian Kecepatan Kendaraan pada Alat Bantu Konveyor

<i>Konveyor</i>	Waktu	<i>Speed</i>	
	Antar Sensor (s)	cm/dtk	Km/jam
<i>Max Speed</i>	0,448	15,62	0,56
<i>Min Speed</i>	2,443	2,86	0,10



Gambar 5. *Plant traffic light* sistem

Tabel 2 berfungsi untuk mengetahui kecepatan maksimum dan minimum. Pada tabel kecepatan maksimum didapat sebesar 0,56 km/jam dengan waktu tempuh 0,448 detik dan kecepatan minimum yang diperoleh adalah 0,10 km/jam dengan waktu tempuh 2,443 detik. Namun kecepatan tersebut belum memenuhi kondisi *input* yang digunakan dalam proses algoritma *fuzzy logic* untuk mencapai kecepatan 80 km/jam. Sehingga untuk mencapai kondisi *input fuzzy logic* dilakukan *scalling* yang dilakukan oleh persamaan 1 sebagai berikut:

$$skala = \frac{\text{Kecepatan Akhir}}{\text{kecepatan Diperoleh}} \quad (1)$$

Maka didapat:

$$skala = \frac{80 \text{ km/jam}}{0,56 \text{ km/jam}}$$

$$skala = 143$$

Oleh karena itu, ukuran yang diperlukan untuk mendapatkan *input fuzzy logic* berkaitan dengan kecepatan pada model menjadi 1 : 143. Selanjutnya, kecepatan kendaraan setelah melakukan *scalling* pada kecepatan terendah adalah 14,3 Km/jam, sedangkan pada kecepatan tertinggi mencapai 80,08 Km/jam, sebagaimana diperlihatkan dalam Tabel 2.

Tabel 2. Kecepatan Kendaraan Setelah Scalling

Kecepatan Kendaraan	Skala 1:143
15,62 cm/detik = 0,56 Km/jam	80,08 Km/jam
2,86 cm/detik = 0,10 Km/jam	14,3 Km/jam

3.2. Pengujian Algoritma Fuzzy logic

Hasil Uji coba pada algoritma logika *fuzzy* dengan menggunakan *microcontroller* ATmega32 dilakukan untuk menerapkan desain algoritma logika *fuzzy* sesuai dengan metode dan rumus yang telah dibuat. Selanjutnya, informasi tersebut diproses di *microcontroller* utama ATmega32 untuk menghasilkan data keluaran yang menunjukkan durasi lampu hijau di setiap sisi. Data keluaran ini dapat dipantau melalui tampilan LCD yang terdapat pada

microcontroller utama, seperti yang terlihat pada Gambar 6.



Gambar 6. Penampil LCD Microcontroller Master

Selanjutnya, informasi dari sistem itu akan dibandingkan dengan hasil perhitungan yang didapat dari teori. Kemudian, hasil dari data tersebut akan dihitung persentasenya, untuk mendapatkan nilai keberhasilan yang berdasarkan dua sumber data percobaan itu. Tabel 3 dan 4 menunjukkan hasil pengujian dari 64 percobaan yang telah dilakukan, tetapi hanya data dari percobaan nomor 49 hingga 64 yang ditampilkan karena memiliki kesalahan bila dibandingkan dengan perhitungan teoritis.

Tabel 3. Pengujian Algoritma Fuzzy logic dengan Sistem Data Sistem Microcontroller

Perc	Data Sistem Microcontroller					
	Input			Output		
	A	B	C	A	B	C
49	78	19	16	6,9	25,1	25,1
50	78	19	33	6,9	25,1	16
51	78	19	53	6,9	25,1	16
52	78	19	74	6,9	25,1	6,9
53	78	36	19	6,9	16	25,1
54	78	53	19	6,9	16	25,1
55	78	36	35	5	16	16
56	78	36	44	4,9	16	16
57	78	50	35	6	16	16
58	78	50	52	6,4	16	16
59	78	36	74	4,9	16	4,9
60	78	50	79	6	16	6
61	78	79	19	6,9	6,9	25,1
62	78	79	35	5	5	16
63	78	79	53	6,6	6,6	16
64	78	79	79	4,3	4,3	4,3

Tabel 4. Pengujian Algoritma *Fuzzy logic* dengan Sistem Data Perhitungan Teoritis

Perc	Data Sistem <i>Microcontroller</i>					
	Input			Output		
	A	B	C	A	B	C
49	78	19	16	6,9	25,2	25,2
50	78	19	33	6,9	25,2	16
51	78	19	53	6,9	25,2	16
52	78	19	74	6,9	25,2	6,9
53	78	36	19	6,9	16	25,2
54	78	53	19	6,9	16	25,2
55	78	36	35	5,0	16	16
56	78	36	44	4,8	16	16
57	78	50	35	6	16	16
58	78	50	52	6,4	16	16
59	78	36	74	4,9	16	4,9
60	78	50	79	6,0	16	6
61	78	79	19	6,9	6,9	25,2
62	78	79	35	5,0	5,0	16
63	78	79	53	6,6	6,6	16
64	78	79	79	4,3	4,3	4,3

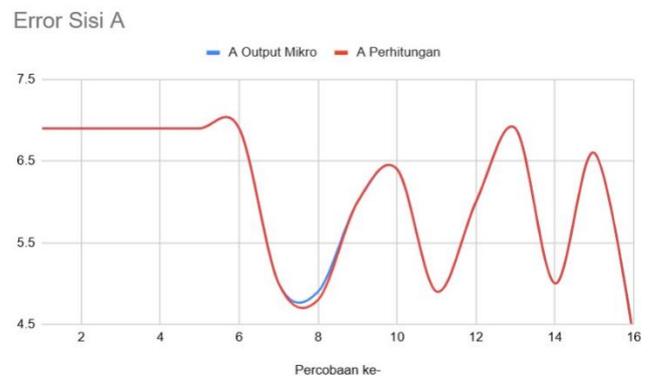
Berdasarkan data output perhitungan algoritma *fuzzy logic* yang diimplementasikan dalam *microcontroller* master ATmega32 dengan perhitungan secara teoritis. Tabel 5 menunjukkan bahwa data *output* hasil akhir keduanya memiliki selisih *error* sebesar 0,0274%.

Tabel 5. Nilai *error output*, jumlah *error* antar sisi, dan rata-rata *error* dalam %

<i>Error Output</i> (%)			
A	B	C	
0	0,40	0,40	
0	0,40	0	
0	0,40	0	
0	0,40	0	
0	0	0,40	
0	0	0,40	
0	0	0	
2,08	0	0	
0	0	0	
0	0	0	
0	0	0	
0	0	0	
0	0	0,40	
0	0	0	
0	0	0	
0	0	0	
0,03	0,02	0,02	Jumlah Error antar sisi (%)
	0,03		Rata-rata Error (%)

Selanjutnya persamaan 2 digunakan untuk mengetahui persentase keberhasilan dari hasil

kedua data pengujian di atas. persamaan (2) persentase keberhasilan = $100\% - \text{Rata-rata error} (\%)$ maka, persentase keberhasilan = $100\% - 0,03\% = 99,97\%$. Gambar 7 menunjukkan persenan *error output* pada sisi A.

**Gambar 7. Error Output**

Gambar 7 menunjukkan data hasil kedua pengujian dari data tabel 5 selisih *error* sebesar 0,0274% yang didapat dari hasil output mikro dan hasil penghitungan. Selanjutnya nilai algoritma *fuzzy logic* diimplementasikan ke *microcontroller* ATmega32. Lalu untuk membuat pengaturan nyala pada purwarupa *plant traffic light* nilai persentase keberhasilan yang didapat sebesar 99,97%. Pada Tabel 4 terdapat data pada percobaan ke-50 yang menunjukkan, *input* kecepatan kendaraan dapat mempengaruhi durasi lampu hijau menyala. Apabila kecepatan kendaraan rendah maka durasi lampu hijau akan semakin lama, sebaliknya apabila kecepatan kendaraan tinggi maka durasi lampu hijau akan semakin cepat.

4. Kesimpulan

Dari hasil tes dan pengumpulan informasi pada Sistem Pengaturan Nyala Lampu pada purwarupa *plant traffic light* berdasarkan kecepatan mobil dengan Metode *Fuzzy logic*, dapat disimpulkan beberapa hal sebagai berikut.

1. Sistem sinyal lalu lintas dapat mengubah lama waktu lampu hijau menyala sesuai dengan kecepatan mobil menggunakan metode logika *fuzzy*. Ini diterapkan dalam *microcontroller* ATmega32 dengan tingkat keberhasilan

mencapai 99,97%. Jika laju kendaraan cepat, maka waktu lampu hijau akan lebih singkat, tetapi jika laju kendaraan lambat, maka waktu lampu hijau akan lebih lama. Lama nyala lampu hijau di setiap sisi dapat beradaptasi dengan menyesuaikan keadaan kecepatan mobil setelah lampu merah total dioperasikan. Kecepatan mobil pada alat bantu konveyor memiliki batas terendah dan tertinggi, yaitu 14,3 km/jam dan 80,08 km/jam dengan rasio 1:143. Komunikasi serial I2C dapat mengirim informasi dari *microcontroller* yang berfungsi sebagai slave ke *microcontroller* yang berfungsi sebagai master sesuai dengan data yang dikirimkan, dengan tingkat keberhasilan mencapai 100%.

5. Ucapan Terima Kasih

Ucapan terima kasih kepada Ignasius Tegar Adiyanto dari pihak PT. Multipanel Intermitra Mandiri yang telah memberi dukungan dalam penelitian, baik berupa sarana maupun dana terhadap penelitian yang telah dilakukan.

Daftar Pustaka

- [1] N. V. Susilo, "Implementasi Metode Logika *Fuzzy* Dalam Pembangunan Sistem Optimalisasi Lampu Lalu Lintas," UAJY, 2016.
- [2] S. Furqon, I. Santoso, and Y.A.A. Soetrisno, "Perancangan Sistem Pengontrolan Lampu Lalu Lintas Berdasarkan Tingkat Kepadatan Kendaraan Menggunakan Metode *Fuzzy*," vol. 9, no. 1, pp. 88-96, 2020.
- [3] J. P. Manalu, "Optimalisasi Waktu Nyala Lampu Hijau Menggunakan *Fuzzy Logic* Pada Persimpangan Jalan Sisingamangaraja-Jalan Turi Kota Medan," 2023.
- [4] S. Amini, T. Fatimah, and R. Pradana, "Rancang Bangun Prototipe Sistem Monitoring Lampu Lalu Lintas Cerdas Berbasis Internet of Things," in *Prosiding Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)*, 2023, vol. 2, no. 1, pp. 407-415.
- [5] B. C. Putra, "Klasifikasi Kendaraan Bergerak Dengan Logika *Fuzzy* Berbasis Pengolahan Citra," Tesis Pascasarjana pada Jurusan Matematika ITS, 2016.
- [6] P. Gunoto, M. Irsyam, and T. K. Wijaya, "Pengembangan Sistem Traffic Lights Berdasarkan Kepadatan Kendaraan Menggunakan PLC," vol. 4, no. 3, 2015.
- [7] H. Cesardarmantya, D. Triyanto, and Y. Brianorman, "Prototype Lampu Lalu Lintas Berbasis PLC Berdasarkan Panjang Antrian Kendaraan Pada Perempatan Jalan," vol. 1, no. 2, 2013.
- [8] I. T. Adiyanto. (2018, 2018). Rekayasa Pengaturan Nyala Lampu Lalu Lintas Berdasarkan Kecepatan Kendaraan Dengan Algoritma *Fuzzy Logic*. Available: http://library.usd.ac.id/Data%20PDF/F.%20Sains%20dan%20Teknologi/Teknik%20Elektro/145114014_full.pdf
- [9] I. Iswanjono, R. C. E. Putri, and I. T. Adiyanto, "Engineering Setting on Traffic Lights Based on Number of Vehicles with *Fuzzy logic* Algorithm," vol. 4, pp. 313-321, 2018.
- [10] N. A. Nafi and E. Wahjono, "Desain Inverter Satu Fasa 400Watt dengan Kontrol *Fuzzy* Sugeno Berbasis *Microcontroller* Untuk Sistem Uninterruptible Power Supply," vol. 9, no. 2, pp. 224-234, 2023.
- [11] A. Azhar and H. Subagiyo, "Perancangan *Fuzzy logic* Model Sugeno untuk Wall Tracking pada Robot Pemadam Api," vol. 1, no. 1, pp. 1-11, 2015.
- [12] R. Hidayat, "Bak Sampah Otomatis Berbasis Robot Line Follower Sebagai Sarana Kemudahan dalam Membuang Sampah di Rumah Sakit," vol. 2, no. 2, pp. 70-77, 2017.
- [13] T. I. Desse, R. Suppa, and M. Mukramin, "Alat pengering ikan asin berbasis *microcontroller* arduino uno," vol. 13, no. 1, 2025.
- [14] D. F. Feriyanto and E. Engineering, "Identifikasi Dan Pemilihan *Microcontroller* untuk *Project Internet Of Things* (IOT)," vol. 7, no. 1, pp. 22-29, 2025.
- [15] K. Rosida, S. H. Wijoyo, and W. S. Wardhono, "Pengembangan Media

- Pembelajaran Booklet pada Mata Pelajaran Informatika Materi Algoritma Bahasa Pemrograman Prosedural," vol. 9, no. 4, 2025.
- [16] M. Malahayati, A. R. Rais, M. R. Fachri, and A. Akhyar, "The Application of AVR Codevision Software in Microprocessor and Microcontroller Programming Subjects at Muhammadiyah Vocational School Banda Aceh," vol. 9, no. 1, pp. 1-8, 2025.
- [17] C. Pradhana and M. Sulaiman, "Simulasi komunikasi serial dengan protokol I2C menggunakan Arduino IDE dan Proteus 8," vol. 3, no. 1, 2020.
- [18] I. Sanyoto, A. Wibowo, and L. Agustine, "Reminder kit secara wireless berbasis *microcontroller* AVR," vol. 13, no. 2, pp. 33-44, 2014.
- [19] R. A. Purnomo, D. Syauqy, and M. H. Hanafi, "Implementasi Metode *Fuzzy* Sugeno Pada Embedded System Untuk Mendeteksi Kondisi Kebakaran Dalam Ruangan," vol. 2, no. 4, pp. 1428-1435, 2018.
- [20] H. Setiawan, F. Saintek, and KomputerVol, "Implementasi Algoritma *Fuzzy* Topsis Pada Sistem Rekomendasi Beasiswa," vol. 1, 2021.