

## **ENGINEERING SETTING ON TRAFFIC LIGHTS BASED ON NUMBER OF VEHICLES WITH FUZZY LOGIC ALGORITHM**

**Iswanjono<sup>1)</sup>, Regina Chelinia Erianda Putri<sup>2)</sup>, Ignasius Tegar Adiyanto<sup>3)</sup>**

<sup>1,2,3</sup>Program Studi Teknik Elektro, Fakultas Sains dan Teknologi, Universitas Sanata Dharma  
Kampus III Paingan, Maguwoharjo, Depok, Sleman, D.I Yogyakarta 55281  
e-mail: <sup>1</sup>iswan\_id@usd.ac.id, <sup>2</sup>regina.chelinia@gmail.com, <sup>3</sup>ignasiustegar96@gmail.com

### *Abstract*

*At crossroads there is often a queue of vehicles that are not evenly distributed in each branch, but the duration of the green light are not adaptive. So prototype was made to regulate the duration of green light based on the number of vehicles with help of the Fuzzy Logic Algorithm. The first stage is fuzzification, after that the rule base will direct to the defuzzification stage which produces the duration green light. The results of the study, Fuzzy Logic algorithm can set the duration of green light for each branch based on the number of vehicles. If a branch has more vehicles than other branches, the duration of time for the green light on the branch is longer than the other branches. The percentage of error testing with theoretical calculations is 0.05% and the percentage of success is 99.95%.*

*Keyword: Fuzzy Logic, Pengaturan Lampu Lalu Lintas, Mikrokontroler AVR*

### **Abstrak**

Pada persimpangan jalan sering terjadi antrian kendaraan yang tidak merata di setiap cabang, tetapi durasi lampu hijau tidak adaptif. Maka prototipe dibuat untuk mengatur durasi lampu hijau berdasarkan jumlah kendaraan dengan bantuan Algoritma Logika Fuzzy. Tahap pertama adalah fuzzifikasi, setelah itu basis aturan akan mengarahkan ke tahap defusifikasi yang menghasilkan durasi lampu hijau. Hasil dari penelitian, algoritma Fuzzy Logic dapat mengatur durasi lampu hijau untuk setiap cabang berdasarkan jumlah kendaraan. Jika cabang memiliki lebih banyak kendaraan daripada cabang lain, durasi waktu untuk lampu hijau dicabang tersebut lebih panjang dari cabang lainnya. Persentase pengujian kesalahan dengan perhitungan teoritis adalah 0,05% dan persentase keberhasilan adalah 99,95%.

**Kata Kunci :** Fuzzy Logic, Pengaturan Lampu Lalu Lintas, Mikrokontroler AVR

### **1. Pendahuluan**

Perkembangan transportasi pada zaman ini sangat pesat, hal ini menyebabkan daya tarik peminat kendaraan menjadi semakin banyak. Pertambahan jumlah kendaraan menyebabkan lalu lintas menjadi padat. Di kota-kota besar sering terjadi kemacetan pada persimpangan jalan, tingkat urbanisasi untuk bekerja maupun aktivitas masyarakat yang tidak menentu sering menimbulkan kesenjangan jumlah kendaraan pada masing-masing cabang persimpangan jalan. Kesenjangan jumlah kendaraan ini dapat terurai, namun membutuhkan waktu yang lebih lama jika hanya dengan sistem pewaktuan lampu lalu lintas biasa (tidak adaptif). Oleh karena itu dibutuhkan perangkat pengatur timer lampu lalu lintas berdasarkan jumlah kendaraan yang bertujuan untuk mengontrol lama nyala lampu lalu lintas.

Sudah ada beberapa solusi yang ditawarkan yang pertama dengan judul “Perangkat Pengatur Timer Lampu Lalu Lintas Berdasarkan Antrian Kendaraan” [1], kelebihan dari alat ini adalah menggunakan 1 mikrokontroler dan area yang diamati ada 2 cabang. Solusi kedua berjudul “Sistem Pengendali Lampu Lalulintas Berbasis Logika Fuzzy” [2] alat ini

menggunakan mikrokontroler Arduino dan MatLab serta menggunakan metode penegasan centroid. Solusi ketiga berjudul “Simulasi Pengaturan Lampu Lalu Lintas Berdasarkan Data Image Processing Kepadatan Kendaraan Berbasis Mikrokontroler Atmega16” [3] kelebihan dari alat ini adalah menggunakan digital citra dalam pemrosesannya. Solusi keempat berjudul “Pengaturan Lampu Lalu Lintas Berbasis Mikrokontroler Atmega8535” [4] kelebihan dari simulator ini adalah area yang diamati ada 4 cabang (perempatan) namun lama lampu hijau tidak adaptif. Solusi kelima berjudul “Simulasi Lampu Lalu Lintas Dengan Sensor Di Simpang Empat Menggunakan Software Automation Studio 5.0” [5] kelebihan dari simulator ini adalah area yang diamati ada 4 cabang (perempatan) menggunakan PLC Allen-Bradley.

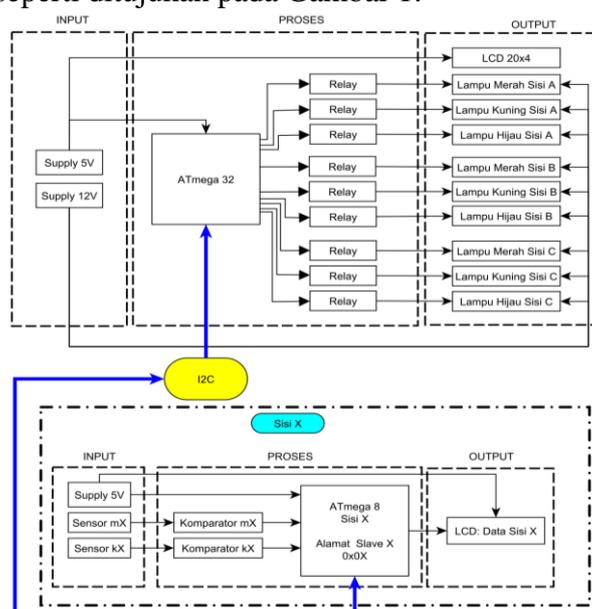
Dalam penelitian ini akan dibuat sistem pengatur lampu lalu lintas berdasarkan jumlah kendaraan dengan ATmega32 sebagai pemroses data, dan ATmega8 pada setiap cabang. Jumlah kendaraan yang akan mengantri pada tiap sisi akan dideteksi oleh sensor inframerah yang ada di ATmega8.

Prinsip kerja sistem ini adalah input berupa jumlah kendaraan tiap cabang. Tahap pertama proses pengambilan data pada mikrokontroler *slave* adalah dengan terdeteksinya *prototype* kendaraan oleh sensor inframerah. Pemasangan sensor adalah pada setiap persimpangan akan diberi 2 sensor inframerah, yang mana sensor inframerah-1 akan bekerja sebagai sensor kendaraan input dan sensor inframerah-2 akan bekerja sebagai sensor output. Kedua sensor inframerah itu akan menjadi masukan dari masing-masing mikrokontroler *slave*. mikrokontroler *master* akan bekerja sebagai prosessor untuk mengontrol lama waktu dari nyala lampu lalu lintas, sehingga dapat meningkatkan efisiensi waktu dari pengguna jalan saat melewati persimpangan.

## 2. Metodologi Penelitian

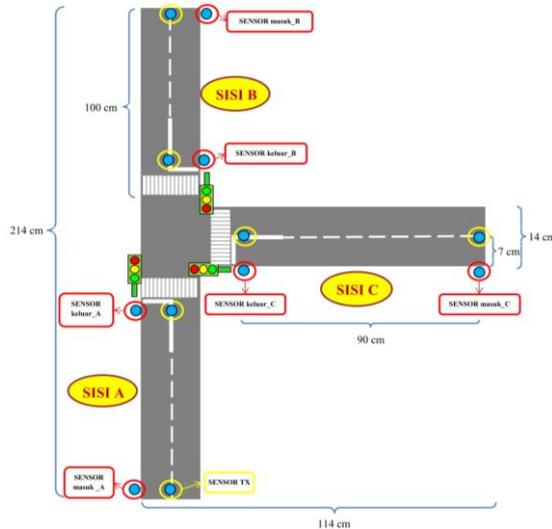
Metode yang digunakan untuk membuat pengaturan nyala lampu lalu lintas adalah sebagai berikut. Tahap pertama adalah perancangan perangkat keras dan perangkat lunak. Tahap kedua adalah mengimplementasikan rancangan perangkat lunak dengan perangkat keras menjadi sistem keseluruhan. Tahap ketiga adalah pengujian dan pengambilan data. Tahap terakhir adalah tahap analisis data dan pengambilan kesimpulan.

Perancangan perangkat keras sistem akan menggunakan 4 mikrokontroler, 1 ATmega32 yang bertindak sebagai *Master* dan 3 ATmega8 yang bertindak sebagai *Slave*. Komunikasi yang digunakan untuk komunikasi antar mikrokontrolernya adalah dengan komunikasi serial I2C, seperti ditunjukkan pada Gambar 1.



Gambar 1. Plant keseluruhan sistem

Sedangkan Gambar 2 merupakan penempatan sensor pada *prototype* yang dibuat. Secara umum, *proses kerja* dari sistem dimulai dari sensor masing-masing *slave* yang memiliki 2 sensor inframerah. Jumlah dari kendaraan dapat diketahui dari selisih antara sensor inframerah-1 (sensor-*in*) sebagai sensor yang mendeteksi kendaraan masuk dan sensor inframerah-2 (sensor-*out*) sebagai sensor yang mendeteksi kendaraan keluar dari area pengamatan.



Gambar 2. Penempatan sensor pada *prototype*

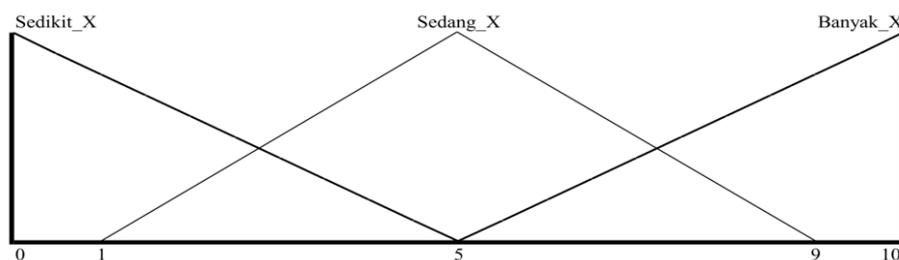
Jumlah kendaraan menjadi masukan untuk diproses oleh mikrokontroler *master*. Pada mikrokontroler *master* akan membandingkan jumlah kendaraan dari ketiga cabang dipersimpangan, cabang yang memiliki jumlah kendaraan yang lebih banyak maka akan mendapatkan waktu penyalan lampu hijau yang lebih lama. Pada setiap mikrokontroler *master* maupun *slave* memiliki LCD sebagai penampil data data pada mikrokontroler *slave* LCD yang digunakan berukuran 16x2, sedangkan penampil pada mikrokontroler *master* memiliki LCD berukuran 20x4.

**2.1. Algoritma Fuzzy Logic**

2.1.1. Framework Input dan Output

Pada perancangan perangkat lunak adalah pemrograman *master* dan *slave*. Pada programming *master* terdapat algoritma *fuzzy logic* yang digunakan sebagai penentu dari lama lampu hijau memiliki framework seperti ditunjukkan pada Gambar 3 sebagai framework input dengan spesifikasi sebagai berikut,

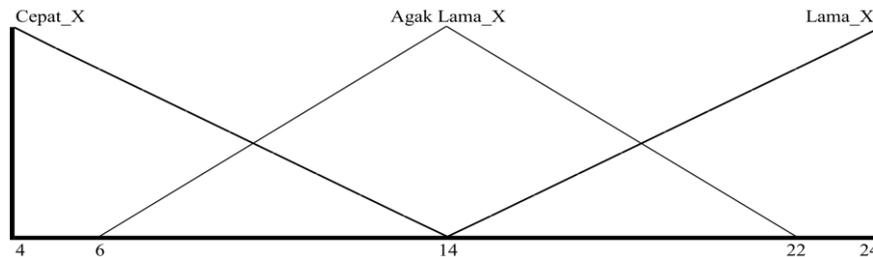
- x : Cabang X
- T : {Sedikit\_X, Sedang\_X, Tinggi\_X}
- X : [0 10]
- M : Sedikit\_X = L (x: 0 5)
- Sedang\_X =  $\wedge$  (x: 1 5 9)
- Banyak\_X = L (x: 5 10)



Gambar 3. Framework input tiap cabang

Sedangkan Gambar 4 sebagai framework output untuk masing-masing cabang dengan spesifikasi sebagai berikut,

- x : Lampu Hijau X
- T : {Cepat\_X, Agak Lama\_X, Lama\_X}
- X : [4 24]
- M : Cepat\_X = L (x: 4 14)
- Agak Lama\_X = ^ (x: 6 14 22)
- Lama\_X = L (x: 14 24)



Gambar 4. Framework output tiap cabang

### 2.1.2. Kondisi Rules

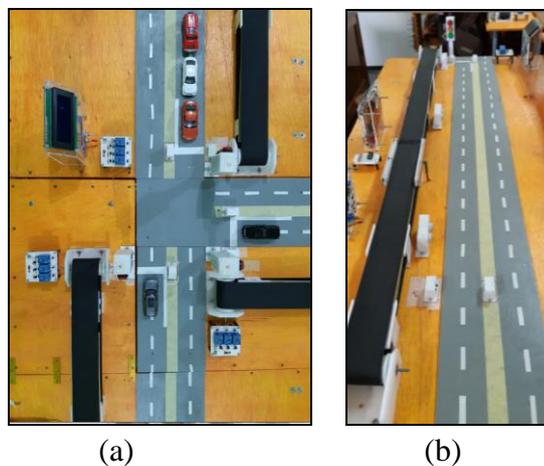
Pada pemilihan *rule*, sebelumnya ditentukan kondisi-kondisi seperti ditunjukkan pada Tabel 1. Saat cabang A, cabang B dan cabang C memiliki nilai maka output akan menyesuaikan lampu mana yang mendapat giliran untuk menyala.

Tabel 1 Kondisi Rules

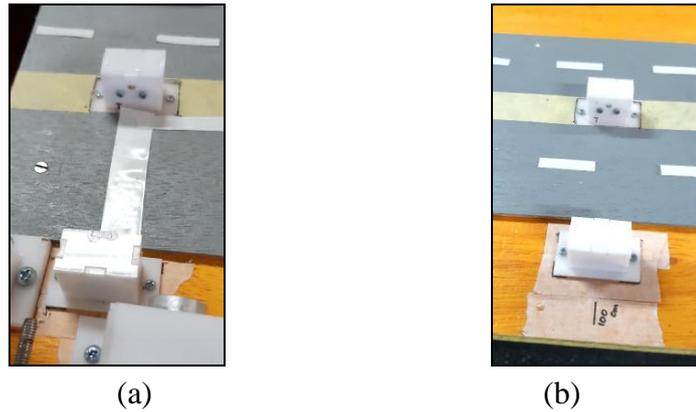
No	Input A	Input B	Input C	Output A	Output B	Output C
1.	Sedikit	Sedikit	Sedikit	Cepat	Cepat	Cepat
2	Sedikit	Sedikit	Sedang	Cepat	Cepat	Agak Lama
3	Sedikit	Sedikit	Banyak	Cepat	Cepat	Lama
4	Sedikit	Sedang	Sedikit	Cepat	Agak Lama	Cepat
5	Sedikit	Banyak	Sedikit	Cepat	Lama	Cepat
6	Sedikit	Sedang	Sedang	Cepat	Agak Lama	Agak Lama
7	Sedikit	Sedang	Banyak	Cepat	Agak Lama	Lama
8	Sedikit	Banyak	Sedang	Cepat	Lama	Agak Lama
9	Sedikit	Banyak	Banyak	Cepat	Lama	Lama
10	Sedang	Sedikit	Sedikit	Agak Lama	Cepat	Cepat

### 3. Hasil dan Pembahasan

Perangkat keras dari *prototype* rekayasa pengaturan lampu lalu lintas terdiri atas minimum sistem ATmega8, ATmega32. Bentuk dari fisik *prototype* rekayasa pengaturan lampu lalu lintas seperti ditunjukkan pada Gambar 5.(a) dan (b). Sensor kendaraan ditunjukkan pada Gambar 6. (a) dan (b)



Gambar 5 (a) *Prototype* Tampak Atas. (b) *Prototype* keseluruhan salah satu sisi



Gambar 6 (a) Sensor Kendaraan Keluar. (b) Sensor Kendaraan Masuk.

### 3.1. Perhitungan Teoritis

Lama lampu secara teoritis dengan Algoritma Fuzzy metode Mamdani (*Max-Min*) dengan metode defuzzifikasi MOM (*Mean of Maximum*).

#### 3.1.1. Perhitungan Fuzzifikasi

Setiap data yang masuk ke *master* adalah jumlah kendaraan dari tiap *slave*, masing-masing data dari tiap *slave* akan memiliki kategori terpilih yang didapat dari perbandingan semua nilai derajat keanggotaan di tiap kategori. Secara teoritis suatu data dapat termasuk dalam 2 kategori sekaligus, sehingga untuk menentukan hanya ada 1 kategori terpilih adalah dengan cara membandingkan kedua kategori tersebut. Kategori yang memiliki nilai derajat keanggotaan tertinggi akan dijadikan sebagai kategori terpilih dari data itu. Pemilihan kategori pada tahap ini merupakan tahap *MAX* dari metode Mamdani. Tahap fuzzifikasi seperti ditunjukkan Tabel 2

Tabel 2 Data dengan kategori terpilih

Data	Perhitungan Fuzzifikasi				Kategori terpilih
	Sedikit	Sedang 1	Sedang 2	Banyak	
0	$\frac{(5-0)}{5} = 1$	-	-	-	Sedikit (SK)
1	$\frac{(5-1)}{5} = 0.8$	$\frac{(1-1)}{4} = 0$	-	-	Sedikit (SK)
2	$\frac{(5-2)}{5} = 0.6$	$\frac{(2-1)}{4} = 0.25$	-	-	Sedikit (SK)
3	$\frac{(5-3)}{5} = 0.4$	$\frac{(3-1)}{4} = 0.5$	-	-	Sedang 1 (SD1)
4	$\frac{(5-4)}{5} = 0.2$	$\frac{(4-1)}{4} = 0.75$	-	-	Sedang 1 (SD1)
5	$\frac{(5-5)}{5} = 0$	$\frac{(5-1)}{4} = 1$	$\frac{(9-5)}{4} = 1$	$\frac{(5-5)}{5} = 0$	Sedang 2 (SD2)
6	-	-	$\frac{(9-6)}{4} = 0.75$	$\frac{(6-5)}{5} = 0.2$	Sedang 2 (SD2)
7	-	-	$\frac{(9-7)}{4} = 0.5$	$\frac{(7-5)}{5} = 0.4$	Sedang 2 (SD2)
8	-	-	$\frac{(9-8)}{4} = 0.25$	$\frac{(8-5)}{5} = 0.6$	Banyak (BY)
9	-	-	$\frac{(9-9)}{4} = 0$	$\frac{(9-5)}{5} = 0.8$	Banyak (BY)
10	-	-	-	$\frac{(10-5)}{5} = 1$	Banyak (BY)

Sehingga untuk kategori terpilih untuk kasus 1 kendaraan untuk sisi A, 4 kendaraan untuk sisi B, dan 9 kendaraan pada sisi C adalah Sedikit pada cabang A dengan nilai derajat keanggotaan 0,8 (secara teoritis) dan 80 (pada program), Sedang 1 pada cabang B dengan nilai derajat keanggotaan 0,75 (secara teoritis) dan 75 (pada program) dan Banyak untuk cabang C dengan nilai derajat keanggotaan 0,8 (secara teoritis) dan 80 (pada program). Pada tahap fuzzifikasi input datanya adalah jumlah kendaraan setiap cabang dan keluarannya adalah nilai derajat keanggotaan.

### 3.1.2. Pemilihan Basis Aturan

Setelah didapatkan kategori terpilih dari setiap cabang maka selanjutnya akan diarahkan pada perhitungan defuzzifikasi dengan bantuan dari Basis Aturan. Setiap kategori terpilih akan mengarahkan ke perhitungan defuzzifikasi. Pada kasus tersebut memiliki kategori yang Sedikit, Sedang dan Banyak akan mengarah pada Tabel 1 nomor 7. Keluaran lama lampu hijau cabang A berada di kategori Cepat, cabang B berada di kategori Agak Lama sedangkan cabang C berada di kategori Lama. Sehingga masukan dari proses basis aturan adalah kategori yang dimiliki setiap cabang dan keluarannya adalah kategori output terpilih untuk proses defuzzifikasi.

### 3.1.3. Perhitungan Defuzzifikasi

Pada proses akhir algoritma *Fuzzy Logic* adalah tahap defuzzifikasi, tahap ini akan mengembalikan data menjadi lama lampu hijau. Tahap defuzzifikasi adalah tahap dimana input nilai derajat keanggotaan dari jumlah kendaraan setiap cabang dan keluarannya adalah lama lampu hijau.

Pada tahap defuzzifikasi ini nilai derajat keanggotaan yang digunakan adalah derajat keanggotaan yang paling minimum. Tahap ini merupakan tahapan *MIN* dari metode Mamdani. Kemudian untuk proses penegasan pada proses defuzzifikasi adalah dengan metode MOM (*Mean of Maximum*). Proses defuzzifikasi dengan penegasan MOM seperti ditunjukkan pada Tabel 3.

Tabel 3 Defuzzifikasi Penegasan MOM

No	Kategori	Rumus
1.	Cepat	$x_{Cepat,X} = \frac{(14 - 10\mu(x)) + 4}{2}$
2.	Agak Lama 1	$x_{AgakLama,X} = \frac{((22 - 8\mu(x)) + (8\mu(x) + 6))}{2}$
3.	Agak Lama 2	$x_{AgakLama,X} = \frac{((22 - 8\mu(x)) + (8\mu(x) + 6))}{2}$
4.	Lama	$x_{Lama,X} = \frac{(10\mu(x) + 14) + 24}{2}$

Pada kasus tersebut derajat terendah dari ketiga *slave* adalah 0,75 dan hasil dari basis aturan untuk kasus tersebut pada output cabang A adalah cepat. Sehingga menggunakan rumus penegasan MOM pada Tabel 3.

$$x_{Cepat,X} = \frac{(14 - (10 * 0,75)) + 4}{2} = 5,25$$

Sehingga lama lampu hijau untuk cabang A yang berisi 1 kendaraan adalah 5,25, namun pada pemrograman dibulatkan 1 angka dibelakang koma menjadi 5,2 detik. Pada output cabang B adalah Agak Lama. Sehingga menggunakan rumus penegasan MOM menjadi,

$$x_{AgakLama,X} = \frac{((22 - (8 * 0,75)) + ((8 * 0,75) + 6))}{2} = 14$$

Sehingga lama lampu hijau untuk cabang B yang berisi 4 kendaraan adalah 14 detik. Pada output cabang C adalah Lama. Sehingga menggunakan rumus penegasan MOM menjadi,

$$x_{Lama,X} = \frac{((10 * 0,75) + 14) + 24}{2} = 22,75$$

Sehingga lama lampu hijau untuk cabang C yang berisi 9 kendaraan adalah 22,75 detik. Namun pada pemrograman dibulatkan 1 angka dibelakang koma menjadi 22,7 detik

### 3.2. Hasil Data Percobaan

Pengujian software dan hardware pada *prototype* dilakukan untuk menguji penggabungan sistem menjadi satu kesatuan.

Dari pengujian data yang telah dilakukan dapat dikatakan berhasil. Komunikasi I2C dapat mengirimkan data dari *slave* ke *master*, hal ini dapat dilihat di Tabel 4, Tabel 5 dan Tabel 6, pada kolom “*Data Master*” memiliki nilai yang sama dengan “*Jumlah*” pada tiap cabang. Terdapat perbedaan dari lama lampu hijau antara perhitungan teoritis dengan pengujian alat (Tabel 4, Tabel 5 dan Tabel 6) karena pada programing yang ditampilkan hanya 1 angka dibelakang koma.

Saat lampu hijau pada cabang A menyala, kondisi lampu pada cabang B dan C adalah lampu merah menyala. Setelah nyala lampu hijau dari cabang A selesai maka akan dilanjutkan dengan menyalakan semua lampu merah atau kondisi (*allred*) selama 3 detik. Kemudian dilanjutkan dengan perhitungan kembali Algoritma *Fuzzy Logic* oleh mikrokontroler untuk menentukan nyala lampu hijau pada cabang B. Data pengujian nyala lampu hijau untuk cabang B, begitu seterusnya.

Keterangan Tabel 4, Tabel 5 dan Tabel 6:

- Perc. = Percobaan  
 Perhit. = Perhitungan teoritis  
 err (%) = Error

Tabel 4 Hasil Percobaan Cabang A

Perc.	SLAVE			MASTER				Perhit.	err (%)
	A	B	C	Data Master			Cabang A		
	Jumlah	Jumlah	Jumlah	A	B	C	Waktu		
1	0	0	0	0	0	0	4	4	0.00
2	2	0	3	2	0	3	6.5	6.5	0.00
3	1	2	6	1	2	6	6	6	0.00
4	1	0	8	1	0	8	6	6	0.00
5	2	4	0	2	4	0	6	6	0.00
6	2	5	1	2	5	1	6	6	0.00
7	2	4	4	2	4	4	6	6	0.00
8	0	3	7	0	3	7	6.5	6.5	0.00
9	0	6	3	0	6	3	6.5	6.5	0.00
10	1	5	7	1	5	7	6.5	6.5	0.00
11	1	4	9	1	4	9	5.2	5.25	0.95

Tabel 5 Hasil Percobaan Cabang B

Perc.	SLAVE			MASTER				Perhit.	err (%)
	A	B	C	Data Master			Cabang B		
	Jumlah	Jumlah	Jumlah	A	B	C	Waktu		
1	0	0	0	0	1	0	4	4	0.00
2	2	0	3	2	0	3	6.5	6.5	0.00
3	1	2	6	1	2	6	6	6	0.00
4	1	0	8	1	0	8	6	6	0.00
5	2	4	0	2	4	0	14	14	0.00
6	2	5	1	2	5	1	14	14	0.00
7	2	4	4	2	4	4	14	14	0.00
8	0	3	7	0	3	7	14	14	0.00
9	0	6	3	0	6	3	14	14	0.00
10	1	5	7	1	5	7	14	14	0.00
11	1	4	9	1	4	9	14	14	0.00

Tabel 6 Hasil Percobaan Cabang C

Perc.	SLAVE			MASTER			Perhit.	err. (%)	
	A	B	C	Data Master					Cabang C
	Jumlah	Jumlah	Jumlah	A	B	C			Waktu
1	0	0	0	0	0	0	4	4	0.00
2	2	0	3	2	0	3	14	14	0.00
3	1	2	6	1	2	6	14	14	0.00
4	1	0	8	1	0	8	22	22	0.00
5	2	4	0	2	4	0	6	6	0.00
6	2	5	1	2	5	1	6	6	0.00
7	2	4	4	2	4	4	14	14	0.00
8	0	3	7	0	3	7	14	14	0.00
9	0	6	3	0	6	3	14	14	0.00
10	1	5	7	1	5	7	14	14	0.00
11	1	4	9	1	4	9	22.7	22.75	0.22

Dari pengujian *prototype* dapat dikatakan berhasil, sensor inframerah dan photodiode pada masing-masing *slave* dapat bekerja dengan mendeteksi kendaraan yang masuk dan keluar. Kemudian data jumlah kendaraan dikirimkan ke *Master* dengan komunikasi serial I2C dan diolah oleh mikrokontroler ATmega32. Lampu lalu lintas juga dapat bekerja dengan baik yaitu dengan menyala sesuai dengan lama lampu hijau yang ditentukan oleh program *Fuzzy Logic*.

Persentase error pada perbandingan pengujian dan perhitungan secara teoritis adalah dengan menggunakan persamaan,

$$\text{Persentase error} = \frac{\text{Error Cab.A} + \text{Error Cab.B} + \text{Error Cab.C}}{3}$$

$$\text{Persentase error} = \frac{0.04\% + 0.05\% + 0.06\%}{3}$$

$$\text{Persentase error} = 0.05\%$$

Persentase keberhasilan alat didapatkan dengan menggunakan persamaan

$$\text{Persentase Keberhasilan Alat} = 100\% - \text{Persentase error}$$

$$\text{Persentase Keberhasilan Alat} = 100\% - 0.053\%$$

$$\text{Persentase Keberhasilan Alat} = 99.95\%$$

Kekurangan dari prototipe ini adalah jika ada perubahan rentang jumlah kendaraan yang diamati dan rentang waktu lampu hijau maka program tahap fuzzifikasi dan defuzzifikasi pada *Fuzzy Logic* akan berubah. Kelebihan dari alat ini adalah perhitungan dan pewaktuan memiliki nilai yang sama namun pada alat terdapat pembulatan sehingga alat memiliki error.

#### 4. Kesimpulan

Setelah melakukan perancangan, pembuatan dan pengujian pada *prototype* rekayasa pengaturan lampu lalu lintas. Dapat diambil kesimpulan sebagai berikut:

1. Algoritma *Fuzzy Logic* dapat digunakan untuk mengatur nyala lampu hijau pada lampu lalu lintas berdasarkan jumlah kendaraan.
2. *Prototype* rekayasa lampu lalu lintas dapat berjalan dengan baik persentase error pengujian dengan perhitungan teoritis adalah 0.05% dan persentase keberhasilan alat dengan Algoritma *Fuzzy Logic* adalah 99.95%.
3. Pengiriman data dari *slave* ke *master* dapat dikirimkan secara baik dengan komunikasi I2C

**Ucapan Terimakasih**

Peneliti mengucapkan terimakasih kepada teman-teman Teknik Elektro Universitas Sanata Dharma Angkatan 2014, dosen pembimbing atas semangat, bantuan, nasihat dan saran yang diberikan kepada peneliti.

**Daftar Pustaka**

- [1] Manto. --. Perangkat Pengatur Timer Lampu Lalu Lintas Berdasarkan Antrian Kendaraan. Depok: Jurnal
- [2] Nugroho, Emmanuel Agung.2017. Sistem Pengendali Lampu Lalulintas Berbasis Logika Fuzzy. Purwakarta: Jurnal
- [3] Jatmika,Sunu dan Indra Andiko. 2014. Simulasi Pengaturan Lampu Lalu Lintas Berdasarkan Data Image Processing Kepadatan Kendaraan Berbasis Mikrokontroler Atmega16. Malang: Jurnal
- [4] Alamsyah. 2012. Pengaturan Lampu Lalu Lintas Berbasis Mikrokontroler Atmega8535. Palu: Jurnal
- [5] Riyadi,Kris Paryanto Dkk. 2014. Simulasi Lampu Lalu Lintas Dengan Sensor Di Simpang Empat Menggunakan Software Automation Studio 5.0. Yogyakarta: Jurnal.