PERANCANGAN SISTEM PENGONTROLAN LAMPU LALU LINTAS BERDASARKAN TINGKAT KEPADATAN KENDARAAN MENGGUNAKAN METODE *FUZZY*

Saiful Furqon *), Imam Santoso dan Yosua Alvin Adi Soetrisno.

Departemen Teknik Elektro, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

*)E-mail: furqonsaiful2@gmail.com

Abstrak

Sistem lampu lalu lintas merupakan perangkat yang berfungsi mengatur arus kendaraan yang akan melewati suatu persimpangan. Durasi lampu lalu lintas umumnya dikendalikan secara otomatis dengan menggunakan durasi waktu yang ditetapkan. Pengendalian ini masih kurang efektif dalam mengatur kendaraan yang akan melewati persimpangan, karena jumlah kendaraan setiap saat tidaklah sama. Berdasarkan masalah tersebut dilakukan pengembangan metode pengendalian lampu lalu lintas yang dapat mengatur durasi nyala lampunya agar dapat menyesuaikan dengan jumlah kendaraan yang akan melewati persimpangan. Logika *fuzzy* dapat digunakan untuk mengatur lama durasi penyalaan lampu lalu lintas yang dapat menyesuaikan dengan jumlah kendaraan yang ada, dan mudah untuk diimplementasikan. Dalam penelitian ini menggunakan mikrokontroler Atmega 328p dengan metode logika *fuzzy* Tsukamoto untuk mengendalikan durasi penyalaan lampu lalu lintas. Terdapat beberapa pengujian yang dilakukan, yaitu pengujian pembacaan data, pengujian penyalaan lampu hijau, pengujian ketika ada ruas kosong, dan ketika semua ruas kosong. Hasil pengujian didapatkan durasi penyalaan lampu lalu lintas dapat menyesuaikan dengan jumlah kendaraan yang ada di ruas persimpangan dengan baik.

Kata kunci : *lampu lalu lintas, logika fuzzy, durasi penyalaan.*

Abstract

Traffic light system is a device which works for regulating the flow of vehicles that will pass an intersection. The duration of the traffic lights is generally controlled automatically using the set time duration. This control is still less effective in regulating vehicles that will pass the intersection, because the number of vehicles at any time is not the same. Based on these problems, the method development of controlling traffic lights that can control the duration of lights activation in order to adjust to the number of vehicles that will pass the intersection. Fuzzy logic can be used to control traffic lights activation duration that can adjust to the number of vehicles available, and easy to be implemented. In this final project using the Atmega 328p microcontroller with Tsukamoto's fuzzy logic method to control the traffic lights activation duration. There are several tests conducted, there is data reading test, green light activation test, when there are empty intersection, and when all intersections are empty. The test results obtained the traffic lights activation durarion can adjust to the number of vehicles that are in the intersection well.

Keywords*: traffic light, fuzzy logic, activation duration.*

1. Pendahuluan

Lampu lalu lintas seharusnya diharapkan dapat mengatur kemacetan yang ada sehingga dapat mencegah kemacetan atau kepadatan kendaraan [1]. Saat ini pengendalian lampu lalu lintas yang banyak diterapkan adalah pengendalian dengan nyala lampu merah, kuning dan hijau yang durasi penyalaanya sudah ditetapkan [2].

Pengendalian dengan metode ini masih kurang efektif, sehingga masih sering menimbulkan kemacetan. Seharusnya durasi penyalaan lampu lalu lintas bisa berubah menyesuaikan tingkat kepadatan kendaraan [3].

Metode pengendalian durasi lampu lalu lintas dan alat deteksi tingkat kepadatan kendaraan yang digunakan beraneka ragam. Salah satu metode pengendalian yang biasa digunakan untuk menentukan durasi penyalaan lampu lalu lintas adalah metode logika *fuzzy*. Metode logika *fuzzy* biasa digunakan karena dapat memodelkan

sistem yang kompleks secara kualitatif, yang susah dimodelkan menggunakan persamaan matematika, dan bagus untuk sistem yang memiliki anomali didalamnya [4]. Seperti analisa pada keadaan kendaraan di persimpangan [5].

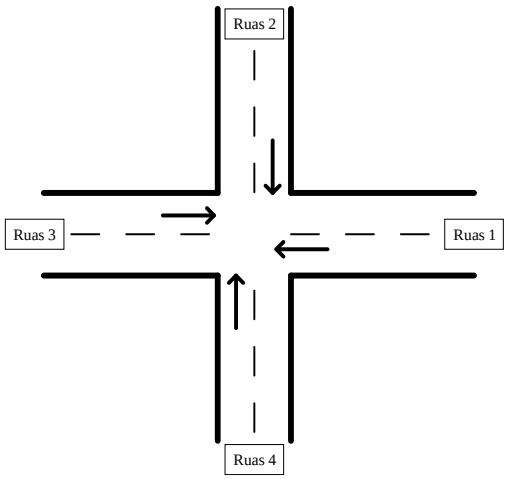
Pada penelitian ini bertujuan untuk merancang purwarupa sistem kendali lampu lalu lintas dengan metode logika *fuzzy* menggunakan mikrokontroler Atmega 328p. Metode logika *fuzzy* ini diharapkan dapat mengatur durasi penyalaan lampu lalu lintas setiap ruasnya berdasarkan tingkat kepadatan kendaraan.

2. Dasar Teori

2.1. Lampu Lalu Lintas

Sistem lampu lalu lintas merupakan perangkat yang berfungsi mengendalikan arus kendaraan yang akan melewati suatu persimpangan. Salah satu cara mengendalikan arus kendaraan, yaitu dengan membuat kendaraan yang terdapat di suatu ruas persimpangan dibolehkan untuk berjalan dengan waktu tertentu, sedangkan kendaraan yang ada di ruas lain harus menunggu sampai waktu yang diberikan di ruas sebelumnya selesai.

Contohnya ketika kendaraan pada ruas satu diberi waktu untuk berjalan, kendaraan ruas dua, tiga, dan empat berhenti. Gambar pesimpangan lampu lalu lintas ditunjukan oleh Gambar 1.



Gambar 1. Lampu lalu lintas pada simpang empat.

Karena pengendalian durasi lampu lalu lintas tidak menyesuaikan jumlah kendaraan yang ada, maka sering terjadi, ketika ruas lain tidak ada kendaraan tetapi kita masih harus menunggu lampu lalu lintas sampai menyala hijau [6]. Berdasarkan masalah tersebut dilakukan pengembangan metode pengendalian lampu lalu lintas yang dapat mengatur durasi nyala lampunya agar dapat menyesuaikan dengan jumlah kendaraan yang akan melewati persimpangan. Salah satu metode pengendalian yang biasa digunakan dalam hal ini adalah metode logika *fuzzy*.

2.2. Logika *Fuzzy*

Logika *fuzzy* adalah logika. Logika merujuk kedalam metode pembelajaran dan prinsip penalaran manusia [6]. Dengan kata lain, logika *fuzzy* telah dibuat agar komputer dapat berpikir seperti manusia [7]. Jika dalam logika klasik, himpunan keanggotaan suatu pernyataan hanya bisa diidentifikasikan sebagai pernyataan benar atau salah. Sedangkan dalam logika *fuzzy*, suatu pernyataan dinyatakan dalam bagian atau perkiraan [8]. Sehingga setiap suatu pernyataan dapat dibagi menjadi beberapa himpunan keanggotaan yang berbeda.

Dalam setiap himpunan keanggotaan logika *fuzzy*, terdiri dari derajat keanggotaan yang nilainya mulai dari 0 hingga 1 [9]. Semakin besar derajat keanggotaan suatu himpunan keanggotaan, maka semakin besar derajat kebenaran dari himpunan tersebut. Dalam pengendalian sistem menggunakan logika *fuzzy* dibagi menjadi tiga bagian, yaitu *fuzzyfication*, *fuzzy reasoning*, dan *defuzzyfication*. *Fuzzyfication* adalah proses untuk mengubah suatu masukan ke dalam bentuk derajat keanggotaan dalam suatu himpunan keanggotaan.

Fuzzy reasoning merupakan proses untuk membuat keputusan dalam menentukan nilai keluaran berdasarkan relasi derajat keanggotaan semua masukan. *Defuzzification* adalah langkah terakhir dalam suatu sistem logika *fuzzy*, tujuannya adalah mengkonversi setiap hasil derajat keanggotaan dari keluaran menjadi suatu nilai yang nyata. Berbeda dari *defuzzification* pada model Sugeno, pada model Tsukamoto tidak menggunakan bobot pada hasil pengambilan keputusan. Proses defuzzifikasi yang digunakan adalah metode *Weight of Average* (WoA), dengan rumus:

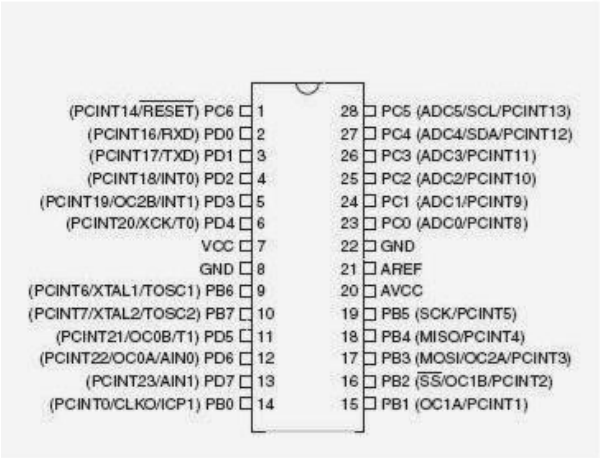
$$z = \frac{W_1 \cdot Z_1 + W_2 \cdot Z_2}{Z_1 + Z_2} \quad (1)$$

Dimana:
w_i – fungsi minimum dari derajat keanggotaan input.
z_i – fungsi *output.*

2.3. Mikrokontroler Atmega 328p

Mikrokontroler Atmega 328p merupakan mikrokontroler AVR 8-bit yang memiliki perfoma tinggi yang bekerja pada tegangan 2,7V hingga 5,5V. Daya yang dikonsumsi rendah dengan konsumsi daya aktif sebesar 1,5mA pada tegangan 3V dan frekuensi 4 MHz. Mikrokontroler Atmega328p juga memiliki 32 x 8 *register* serbaguna,

dengan *clock* 16 MHz dan mencapai kecepatan 16 MIPS. Memori yang dimiliki Atmega 328p adalah 32 KB *flash memory*, 1 KB EEPROM, dan 2 KB SRAM. Pin masukan dan keluaran yang dimiliki oleh Atmega 328p adalah 23 pin. Atmega 328p biasa dibuat dalam bentuk modul, salah satu modul Atmega 328p adalah modul Arduino Nano. Berikut Arduino Nano yang ditunjukan oleh Gambar 2.



Gambar 2. ***Pin Arduino Nano***

Modul Arduino Nano memiliki dimensi yang kecil. Alokasi pin pada modul Arduino Nano adalah 14 pin digital I/O dan 8 pin analog.

2.4. Aritmatika *Fixpoint*

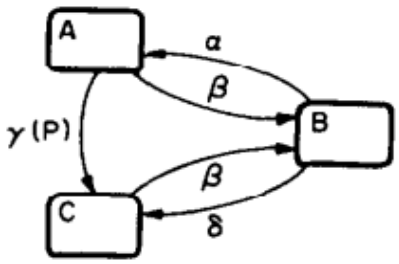
Pada logika *fuzzy* terdapat himpunan keanggotaan dalam *fuzzyfication* yang didalamnya terdapat derajat keanggotaan berupa bilangan desimal. Bilangan desimal perlu didaftarkan ke dalam *register float* agar bisa diolah di mikrokontroler. Akan tetapi pemrosesan *register float* di mikrokontroler membutuhkan memori lebih besar dan durasi yang cukup lama dibandingkan dengan *register* lainya [10]. Untuk mempercepat proses bilangan desimal di mikrokontroler dapat menggunakan aritmatika *fixed point*.

Aritmatika *fixed point* (bilangan bulat) merupakan metode komputasi mikrokontroler yang digunakan untuk mengubah operasi bilangan pecahan menjadi operasi bilangan bulat agar mengurangi penggunaan memori mikrokontroler dan mempecepat waktu komputasi [10]. Pengubahan bilangan pecahan menjadi bilangan bulat pada metode *fixed point*, yaitu dengan mengalikan bilangan pecahan tersebut dengan suatu mantisa. Mantisa adalah kelipatan bilangan dua berpangkat.

Hasil operasi bilangan tidak boleh melebihi kapasitas *register* yang digunakan di mikrokontroler. Contohnya, apabila suatu bilangan di daftarkan dalam *register* *unsigned int* maka nilai hasil hasil operasi maksimal adalah 65535. Nilai mantisa menentukan keakuratan hasil operasi [11]. Semakin besar nilai mantisa, maka keakuratan hasil operasi semakin besar. Akan tetapi, jika nilai mantisa terlalu besar dapat beresiko hasil operasi melebihi kapasitas *register*[12]. Jika hasil operasi melebihi kapasitas *register*, maka akan terjadi *stack overflow* sehingga nilai hasil operasi bisa sangat jauh berbeda dari nilai sebenarnya.

2.5. *State Chart*

State chart, diagram keadaan atau disebut juga *finite state machine* merupakan perpindahan kondisi (*state*) yang dipicu oleh suatu kejadian. Gambar 3. menunjukkan sebuah contoh sederhana mengenai diagram keadaan. Ketika sistem pada kondisi A dan ada kejadian β, maka sistem akan berpindah ke dalam kondisi B. Ketika sistem pada kondisi B dan kejadian α terjadi, maka sistem akan berpindah ke dalam kondisi A [13].



Gambar 3. Contoh diagram keadaan.

Dalam sistem lampu lalu lintas di simpang empat, kita harus membuat keadaan yang membentuk siklus [14]. Dalam penelitian ini digunakan empat siklus ruas lampu lalu lintas yang dinyalakan, dan enam siklus penyalaan lampu pada setiap ruas.

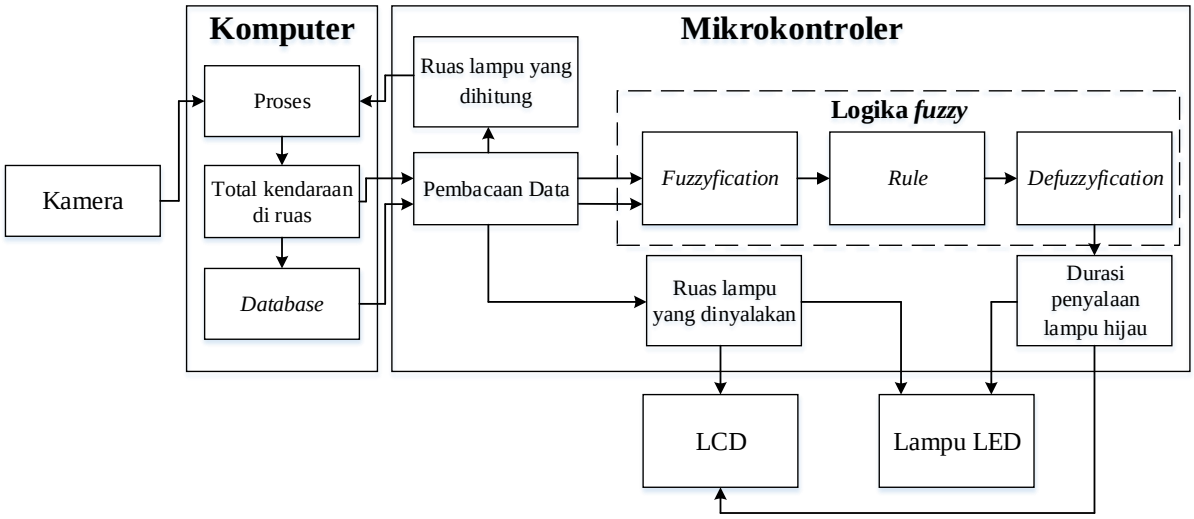
3. Metode

3.1. Perancangan Perangkat Keras

Perancangan perangkat keras dalam penelitian penelitian ini terdiri dari 2 bagian utama, yaitu mikrokontroler Atmega328p, dan modul lampu LED. Diagram blok perancangan perangkat keras yang dirancang pada penelitian ini ditunjukkan pada Gambar 4. Diagram blok pada Gambar 4. menunjukan skema alur dari alat yang akan dibuat pada penelitian ini dengan spesifikasi setiap bloknya sebagai berikut.

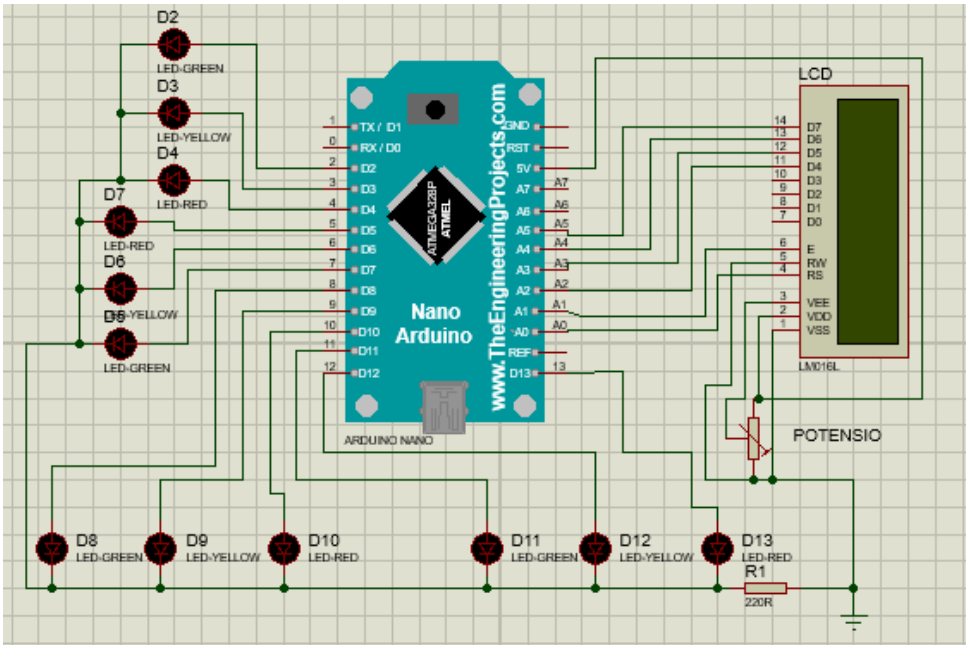
1. Mikrokontroler mengirimkan data berupa ruas lampu yang akan dihitung ke komputer.
2. Selesai menghitung, komputer mengirim bobot jumlah kendaraan dan data kendaraan per-jam sebagai masukan untuk logika *fuzzy*.

3. Dalam logika *fuzzy*, bobot jumlah kendaraan dan data kendaraan per-jam masuk fungsi *fuzzyfication* untuk diklasifikasikan dalam himpunan keanggotaan.
4. Keluaran dari fungsi *fuzzyfication* digunakan untuk menentukan keputusan di fungsi aturan.
5. *Defuzzyfication* bertujuan untuk mengubah hasil keputusan menjadi nilai sesungguhnya yaitu durasi penyalaan lampu hijau dalam bentuk milidetik.
6. Durasi penyalaan lampu hijau digunakan sebagai waktu penyalaan lampu hijau LED, dan masukan LCD.
7. Selain untuk masukan logika *fuzzy*, bobot jumlah kendaraan juga memiliki saluran pengiriman yang digunakan oleh mikrokontroler untuk menentukan ruas lampu LED yang dinyalakan dan sebagai masukan data LCD.
8. Mikrokontroler juga berperan untuk mengatur durasi penyalaan lampu merah dan kuning, serta mengatur tampilan yang ada di LCD.



Gambar 4. Diagram blok perancangan perangkat keras

Koneksi pin modul Arduino Nano ditunjukan oleh Gambar 5.



Gambar 5. Alokasi pin modul mikrokontroler

Alokasi pin modul mikrokontroler tidak menggunakan pin digital D0 dan D1 karena pin tersebut dipakai untuk berkomunikasi dengan komputer menggunakan kabel *serial*.

Sisa 12 pin digital untuk mengendalikan penyalaan lampu LED dengan rincian pin D2, D3, dan D4 untuk modul satu , D5, D6, dan D7 untuk modul dua, B0, B1, dan B2 untuk modul tiga, B3, B4, dan B5 untuk modul empat. Ke empat modul LED dihubungkan dengan satu *resistor* 220Ω yang terhubung ke *ground* di mikrokontroler. Ada enam pin analog yang digunakan untuk memgendalikan LCD yaitu pin A0, A1, A2, A3, A4, dan A5.

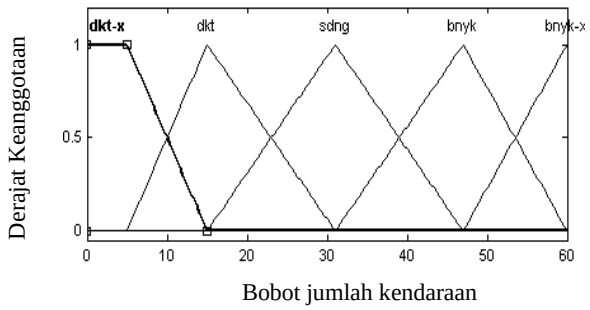
3.2. Perancangan Metode Kontrol *Fuzzy* Tsukamoto pada Pengendalian Durasi Lampu Hijau

Algoritma kontrol yang diterapkan pada sistem pengendali durasi lampu hijau adalah metode kontrol *fuzzy* Tsukamoto. Proses kontrol dilakukan ketika komputer telah mengirimkan jumlah kendaraan dari hasil perhitungan dan jumlah kendaraan per-jam dari data *database*. Berikut ini akan dijelaskan algoritma kontrol *fuzzy* dalam penentuan lama durasi pada lampu hijau.

3.2.1. *Fuzzyfication*

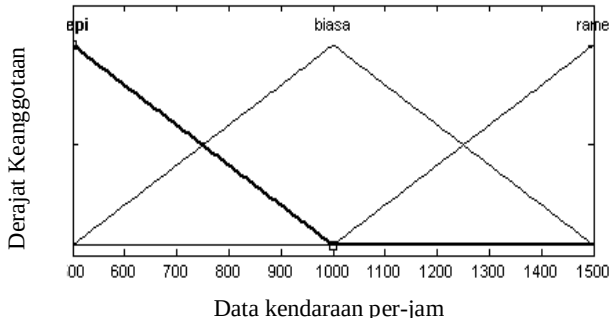
Fuzzyfication adalah proses pemasukan nilai masukan ke dalam himpunan keanggotaan agar dapat memiliki nilai derajat keanggotaan. Himpunan keanggotaan pada bobot jumlah kendaraan dibagi menjadi lima bagian, yaitu sedikit sekali (dkt-x), sedikit (dkt), sedang (sdng), banyak (bnyk), dan banyak sekali (bnyk-x).

Sedangkan himpunan keanggotaan untuk masukan data kendaraan per-jam dibagi menjadi tiga, yaitu kendaraan sedikit (sepi), kendaraan sedang (biasa), dan kendaraan banyak (rame). Masing –masing masukan seperti yang ditunjukkan pada Gambar 6. dan Gambar 7. berikut.



Gambar 6. Himpunan keanggotaan masukan bobot jumlah kendaraan

Diasumsikan deteksi bobot jumlah kendaraan paling banyak yang terdeteksi dalam satu ruas adalah 60. Asumsi ini mengikuti spesifikasi dari kemampuan pendeteksian kendaraan.



Gambar 7. Himpunan keanggotaan masukan kendaraan per-jam

Untuk menentukan parameter himpunan keanggotaan pada masukan data kendaraan per-jam, jumlah kendaraan didapat dengan menghitung kendaraan yang ada pada rekaman CCTV pada salah satu persimpangan di kota Semarang. perhitungan dilakukan dengan asumsi jumlah kendaraan yang diam di depan lampu lalu lintas saja. Didapat jumlah kendaraan sepi 324 yang dibulatkan menjadi 500 dan jumlah kendaraan pada saat ramai mencapai 1328 yang dibulatkan 1500.

3.2.2. *Fuzzy Reasoning*

Fuzzy reasoning atau pengambilan keputusan *fuzzy* pada penelitian ini dirancang. Penentuan *fuzzy reasoning* dilakukan dengan melihat dari nilai kedua masukan yang kemudian diambil keputusan agar keluaran sesuai dengan durasi lampu hijau yang diharapkan. Berdasarkan pengamatan karakteristik sistem yang telah dilakukan, diperoleh *rule base fuzzy* yang dapat dilihat pada Tabel 3.1.

Tabel 1. *Rule base fuzzy*

	Sedikit Sekali	**Sedikit**	**Sedang**	**Banyak**	**Banyak Sekali**
Sepi	Z1	Z2	Z3	Z4	Z5
Biasa	Z1	Z2	Z3	Z4	Z6
Rame	Z1	Z2	Z3	Z4	Z7

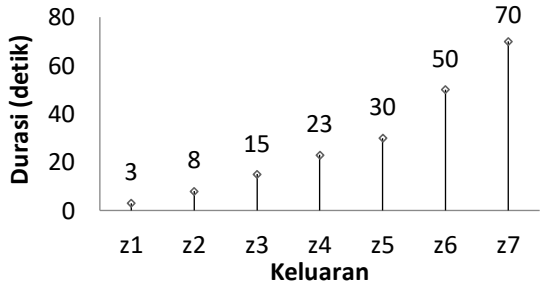
Penentuan parameter *rule base* didapatkan melalui pertimbangan waktu kendaraan yang lewat. Untuk hasil keluaran selain himpunan keanggotaan banyak sekali, tidak dipengaruhi oleh perubahan parameter data kendaraan per-jam. Z1 hingga Z7 adalah nilai keluaran (durasi lampu hijau).
Sedangkan untuk himpunan keanggotaan banyak sekali dipengaruhi oleh parameter tersebut karena bisa saja tidak

semua kendaraan terdeteksi oleh kamera karena sangat banyak jumlahnya. Oleh karena itu penting melakukan pencatatan *database* kendaraan per-jam agar dapat diketahui bahwa ruas jalan pada saat itu dalam keadaan ramai.

3.2.3. *Defuzzyfication*

Defuzzyfication merupakan tahap akhir dari perancangan metode kontrol *fuzzy*. Perancangan menggunakan metode *fuzzy logic* Tsukamoto dengan metode *weight of average* agar didapatkan nilai keluaran *fuzzy* untuk semua nilai keluaran dari setiap aturan.

Himpunan keanggotaan keluaran (durasi lampu hijau) memilki bentuk *singleton* yang ditunjukkan pada Gambar 8. Berikut.



Gambar 8. Keluaran *singleton* sistem *fuzzy* durasi lampu hijau

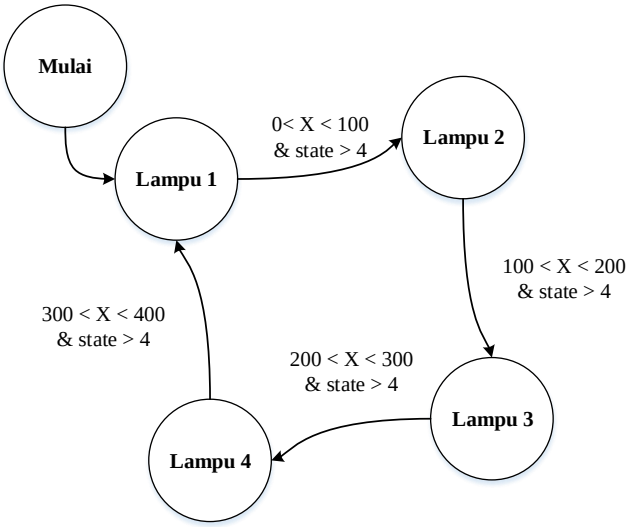
Penentuan nilai keluaran durasi dengan melihat CCTV kota Semarang di jalan kaligarang pada hari Senin, 2 Desember 2019 pada jam 10 hingga 11 siang. Didapat untuk keluaran pertama (z1) adalah tiga detik, keluaran ke-dua (z2) adalah 8 detik, keluaran ke-tiga (z3) adalah 15 detik, keluaran ke-empat (z4) adalah 23 detik, keluaran ke-lima (z5) adalah 30 detik, keluaran ke-enam (z6) adalah 50 detik, dan keluaran ke-tujuh (z7) adalah 70 detik.

3.3. Perancangan Program Pengendali Durasi Lampu Lalu Lintas

Perancangan program pada penelitian ini dibuat menggunakan *software* dan *compiler Arduino IDE* v1.8.10 Proses pengunggahan program ke mikrokontroler Atmega328p menggunakan konektor *serial*. Algoritma sistem pengendalian durasi lampu lalu lintas sebagai berikut.

1. Mulai.
2. Inisialisasi I/O *register* dan variabel.
3. Penjadwalan program yang akan dilaksanakan.
4. Pengiriman data melalui komunikasi *serial*.
5. Pembacaan data dari komputer.
6. Jika ada kendaraan dalam ruas, lakukan kontrol durasi lampu hijau dengan kontrol logika *fuzzy*.
7. Jika tidak ada kendaraan yang terdeteksi, nyalakan lampu merah.
8. Jika tidak ada kendaraan selama satu menit disetiap ruas, nyalakan lampu kuning berkedip disetiap ruas.
9. Proses penyalaan lampu LED dan menampilkan proses di LCD.
10. Mengulangi langkah empat hingga 10 selama proses kontrol berjalan.

Pada pembacaan data dari komputer terdapat beberapa kondisi untuk menentukan ruas lampu yang akan dinyalakan, berdasarkan data yang dikirim dari komputer seperti yang ditunjukan Gambar 3. berikut.



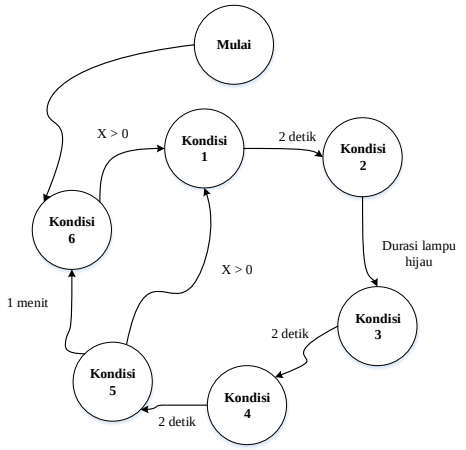
Gambar 9. Penentuan ruas lampu yang akan dinyakan

Pada Gambar 9. menunjukan urutan penyalaan ruas lampu lalu lintas, apabila setiap ruas terdapat kendaraan. Penjelasan setiap kondisi penyalaan ruas adalah sebagai berikut.

1. Ketika x (bobot jumlah kendaraan) < 100, maka ruas lampu yang dinyalakan adalah ruas satu.
2. x ≥ 100 dan x < 200, maka ruas lampu yang dinyalakan adalah ruas dua.
3. x ≥ 200 dan x < 300, maka ruas lampu yang dinyalakan adalah ruas tiga.
4. x ≥ 300 dan x < 400, maka ruas lampu yang dinyalakan adalah ruas empat.

Karena sistem dirancang agar mikrokontroler dapat berkomunikasi dengan komputer dan menyalakan lampu secara bersamaan, maka perlu dilakukan penjadwalan pelakasanaan program [15]. Dalam proses penyalaan

lampu lalu lintas juga perlu dibuat kedalam beberapa kondisi penyalaan sebagai berikut.



Gambar 10. Kondisi penyalaan lampu di setiap ruas

Pada Gambar 3. kondisi penyalaan lampu ini berubah sesuai dengan waktu yang telah ditentukan. Pada kondisi satu, tiga, dan empat, durasi waktunya adalah dua detik. Pada durasi kondisi dua, durasi penyalaan lampu ditentukan oleh hasil perhitungan logika *fuzzy*. Kondisi lima selama bertahan selama satu menit apabila tidak ada kendaraan yang terdeteksi. Sedangkan pada kondisi enam tidak ada batas waktu, akan tetapi berubah apabila ada kendaraan yang terdeteksi.

Proses yang terjadi pada setiap kondisi adalah sebagai berikut.

1. Nyalakan lampu kuning dan matikan lampu merah.
2. Nyalakan lampu hijau dan matikan lampu kuning.
3. Nyalakan lampu kuning dan matikan lampu hijau.
4. Nyalakan lampu merah dan matikan lampu kuning.
5. Nyalakan lampu merah di setiap ruas karena tidak ada kendaran yang terdeteksi.
6. Nyalakan lampu kuning berkedip pada setiap ruas.

4. Pengujian dan Analisis

4.1. Pengujian dan Pengiriman Data

Pengujian ini akan membahas mengenai hasil pengujian pada pembacaan dan pengiriman data dari komputer yang dilakukan oleh mikrokontroler. Pengujian pembacaan data terdapat dua tahap yaitu pembacaan bobot jumlah kendaraan dan pembacaan data kendaraan per-jam.

4.1.1. Pembacaan Bobot Jumlah Kendaraan

Pembacaan bobot jumlah kendaraan di setiap ruas lampu dilakukan dengan memberikan data masukan dari komputer untuk setiap ruas dengan nilai bobot jumlah kendaraan 10.

Tabel 2. Masukan bobot jumlah kendaraan

Masukan	Ruas 1	Ruas 2	Ruas 3	Ruas 4
Data yang dibaca	10	110	210	310
Bobot kendaraan	10	10	10	10

Tabel 2. menunjukan hasil pembacaan data yang dibaca dari komputer yang ada di ruas satu, dua, tiga, dan empat dengan disertai bobot jumlah kendaraan yang dibaca mikrokontroler. Data yang dibaca dari komputer memiliki bobot jumlah kendaraan 10.

4.1.2. Pembacaan Data Kendaraan Per-Jam

Pembacaan data kendaraan per-jam dilakukan dengan dua variasi masukan data kendaraan per-jam yang dikirim dari komputer.

Variasi ini ketika *database* menunjukan ruas persimpangan dalam keadaan sepi dengan nilai 202 dan ketika *database* menunjukan ruas salah satu persimpangan dalam keadaan biasa dengan nilai 740.

Tabel 3. Masukan data kendaraan per-jam

Masukan	Variasi 202	Variasi 740
Data yang dibaca	702	1240
Data kendaraan per-jam	202	740

Tabel 3. menunjukan hasil data yang dibaca sebesar 702 dan 1240. Kedua data tersebut merupakan nilai yang dikirim oleh komputer, sedangkan data kendaraan perjam sebesar 202 dan 740.

4.2. Pengujian Durasi Penyalaan Lampu Hijau

Tabel 4. Perbandingan total durasi penyalaan lampu dalam satu siklus

Kondisi Ramai	Durasi Lampu (detik)				
	Ruas 1	Ruas 2	Ruas 3	Ruas 4	Total
1 Ruas Ramai	44,11	12,81	8,437	3	92,359
2 Ruas Ramai	39,1	65,23	3,57	15,63	147,53
3 Ruas Ramai	27,93	46,4	70	14,5	182,83
4 Ruas Ramai	23	41,2	31,6	60,2	180

Pengujian ini dilakukan dengan memberikan masukan berupa bobot jumlah kendaraan dan data kendaraan perjam dari *database* secara langsung.dari komputer kemudian hasilnya ditampilkan menggunakan komunikasi *serial*. Pengujian ini dilakukan dengan beberapa variasi keramaian pada setiap ruas lampu lalu lintas yang

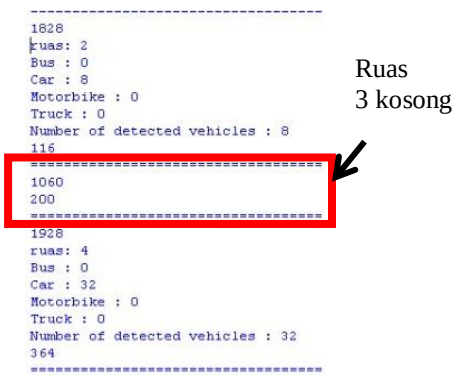
ditampilkan Total durasi lampu adalah total durasi lampu hijau setiap ruas ditambah total durasi lampu kuning dan merah pada setiap ruas sebesar 24 detik.

Pada Tabel 4. menunjukan apabila semakin banyak kendaraan yang terdeteksi maka durasi penyalaan lampu lalu lintas semakin lama. Jika ruas di persimpangan lampu lalu lintas banyak yang ramai, maka waktu siklus penyalaan lampu lalu lintas dari satu ruas lampu ke ruas lampu lain semakin lama. Hasil ini menunjukan semakin banyak ruas yang ramai, maka waktu tunggu pengendara pada setiap ruas juga semakin bertambah lama.

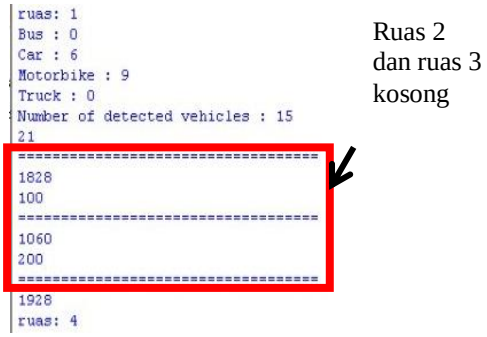
4.3. Pengujian Ketika Ada Ruas Kosong

Pengujian ini dilakukan untuk mengetahui respon lampu lalu lintas apabila ada ruas lampu yang tidak ada kendaraanya. Jika ada kendaraan kosong maka respon yang diharapkan adalah mikrokontroler tetap menyalakan lampu merah di ruas tersebut, dan mengirimkan sinyal ke komputer untuk menghitung bobot jumlah kendaraan di ruas selanjutnya. Sehingga tidak ada perubahan penyalaan lampu pada ruas lampu yang kosong tersebut.

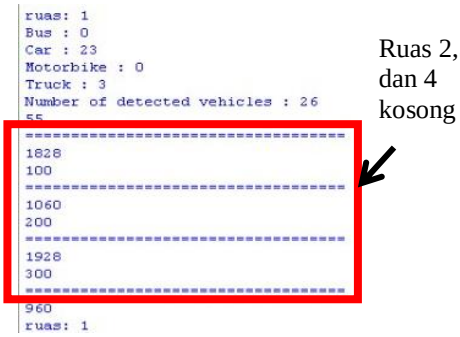
Pengujian ini dilakukan dengan menggunakan *software python* 3.6.8 yang ditampilkan Gambar 11. hingga Gambar 13.



Gambar 11. Pengujian ketika salah satu ruas kosong



Gambar 12. Pengujian ketika dua ruas kosong



Gambar 13. Pengujian ketika tiga ruas kosong

Pada Gambar 11. hingga Gambar 13. menunjukan tidak terjadi perhitungan. Lampu di ruas yang kosong tetap menyala merah.

4.4. Pengujian Ketika Semua Ruas Kosong

Pengujian ini dilakukan untuk mengetahui respon mikrokontroler apabila tidak ada kendaraan pada setiap ruas lampu lalu lintas. Jika ke-empat ruas kosong mikrokontroler tetap menyalakan lampu merah di setiap ruas selama satu menit. Ketika setelah satu menit tetap tidak ada kendaraan yang terdeteksi, maka mikrokontroler akan menyalakan lampu kuning pada setiap ruas lampu secara berkedip dengan jeda kedipan, satu detik

5. Kesimpulan

Metode kontrol *fuzzy* sebagai pengendali durasi penyalaan lampu lalu lintas telah berhasil direalisasikan dengan menggunakan mikrokontroler Atmega 328p. Dari hasil pengujian mikrokontroler dapat berkomunikasi secara *real-time* dengan komputer dalam mengirim dan menerima data. Pada pengujian durasi penyalaan lampu hijau dengan menggunakan kontrol *fuzzy*, durasi penyalaan dapat berubah menyesuaikan masukan yang ada dari komputer. Pengendalian penyalaan lampu lalu lintas tetap menyalakan lampu merah pada ruas yang kosong. Penyalaan lampu lalu lintas akan diatur menjadi lampu kuning ber-kedip apabila semua ruas kosong dalam satu menit.

Saran untuk pengembangan penelitian ini yaitu. Perlu dilakukan pengujian durasi lampu hijau yang digunakan agar dapat diketahui apakah bisa diterapkan. Penambahan pengendalian lampu lalu lintas secara manual. Penambahan fitur *Graphical User Interface* (GUI) yang ditampilkan di komputer sehingga dapat dilakukan monitoring dari server, serta untuk mengubah mode pengendalian dari otomatis ke manual, atau sebaliknya.

Referensi

[1] M. Maslim, B. Y. Dwiandiyanta and N. V. Susilo, "Implementasi Metode Logika Fuzzy dalam Pembangunan Sistem Optimalisasi Lampu Lalu Lintas," *Jurnal Buana Informatika*, Vol 9, pp 11-20, 2018.

[2] S. Chandel, S. Yadav and S. Yadav, "Modern Traffic Control System," Malaya Journal of Matematik, vol. S, pp. 22-25, 2018.

[3] N. Diaz, J. Guerra and J. Nicola, "Smart Traffic Light Control System," *IEEE*, 2018.

[4] S. Mehan, "Introduction of Traffic Light Controller with Fuzzy Control System," International Journal of Electronics & Communication Technology, vol. 2, no. 3, September 2011.

[5] R. Hoyer, "Fuzzy Control of Traffic Lights," *IEEE*, 1994.

[6] G. Chen, Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems, CRC Press, 2001.

[7] A. M. Ibrahim, Fuzzy Logic for Embedded Systems Applications, Elsevier Science, 2004.

[8] T. J. Ross, Fuzzy Logic with Engineering Applications, Wiley, 2010.

[9] V. B. Rao, C++ Neural Network and Fuzzy Logic, IDG Books Worldwide, 1995.

[10] E. L. Oberstar, "Fixed-Point Representation & Fractional Math," 2007.

[11] R. Nehmeh, Quality Evaluation in Fixed-point Systems with Selective Simulation, INSA de Rennes, 2017.

[12] A. Cervin, Fix Point Implementation of Control Algorithms, Lund University, 2009.

[13] D. Harel, "Statecharts: A Visual Formalism For Complex Systems," *Science of Computer Programming*, vol. 8, pp. 231-274, 1987.

[14] J.-H. Lee and H. Lee-Kwang, "Distributed and Cooperative Fuzzy Controllers for Traffic Intersections Group," *IEEE*, vol. 29, pp. 263-271, May 1999.

[15] S. Natarajan and D. Broman, "Timed C: An Extension to the C Programming Language for Real-Time Systems," *IEEE*, 2018.