

## **PROTOTYPE LAMPU LALU LINTAS ADAPTIF BERBASIS MULTI AGENT MANGGUNAKAN LOGIKA FUZZY YANG TERTANAM PADA MICROCONTROLLER**

**Adrin. T<sup>1</sup>, H. Himawan<sup>2</sup>, Suprapedi<sup>3</sup>**

<sup>1,2</sup>Pasca Sarjana Teknik Informatika Universitas Dian Nuswantoro

<sup>3</sup>Lembaga Ilmu Pengetahuan Indonesia (LIPI)

### **ABSTRACT**

*Traffic light is a very important tool for urban life in regulating the smooth flow of traffic on the highway. The use of traffic lights are now much more applied using static timing system or a traffic light system which does not know the condition of the crossing are many vehicles or fewer. Timing pattern that is applied from one segment to another segment is set in rotation. In the study conducted to design a traffic light prototype based on multi- agent using fuzzy logic which is planted on the microcontroller. There are two parts and types of microcontrollers are used to design the prototype. The first part, the Slave microcontroller Atmega 32 and the second type, the type of Master mikontroler Atmega 128. Traffic light system which can adapt to the environment made the crossing. If there is a deviation of the queue of vehicles which have very much, then the green light time longer than the deviation that only have a little queue of vehicles. Thus, the traffic light is more adaptive to the dynamic vehicle that will cross the intersection. Traffic lights can also communicate with traffic lights nearest neighbors in both directions. Communication is done through mutual give information about the number of vehicles that left deviation towards each junction nearest neighbors. Results of this study found that the performance of fuzzy logic embedded in the microcontroller can control traffic lights with dynamic adaptive to existing vehicles on Line 1, Line 2 and Line 3. Prototype designed traffic lights represent the environmental conditions that have multiple intersections and each intersection has four traffic lights that can communicate with Multi Agent another intersection nearest neighbors.*

*Keywords: static, prototypes, fuzzy logic, microcontrollers, multi-agent.*

## **1. PENDAHULUAN**

### **1.1. Latar Belakang**

Lampu lalu lintas adalah alat yang sangat penting untuk kehidupan perkotaan dalam mengatur kelancaran arus lalu lintas pada jalan raya. Penggunaan lampu lalu lintas saat ini lebih banyak diterapkan dengan menggunakan sistem pewaktuan yang statis atau sistem lampu lalu lintas yang tidak mengetahui kondisi persimpangan sedang banyak kendaraan ataukah sedikit. Pola pengaturan waktu yang diterapkan dari satu ruas ke ruas lainnya di atur secara bergilir. Dengan pola pengaturan tersebut apabila terdapat satu ruas jalan yang sedang padat tetapi belum mendapat giliran maka harus menunggu sampai tiba giliran untuk mendapatkan nyala lampu hijau. Penerapan sistem lampu lalu lintas seperti ini ternyata tidak menjawab kondisi yang terjadi di jalan raya, sehingga bisa menimbulkan niat masyarakat untuk melakukan pelanggaran terhadap lampu lalu lintas, yang justru berpotensi mengakibatkan kecelakaan. Kondisi tersebut bisa terjadi pada satu persimpangan bahkan di beberapa persimpangan yang ada pada satu perkotaan. Untuk itu, sebaiknya jika dalam satu ruas jalan hanya terdapat sedikit antrian kendaraan maka lamanya lampu hijau lebih cepat dibanding jika dalam ruas tersebut banyak kendaraan. Sehingga pola pengaturan lampu lalu lintas tersebut lebih adaptif dan tidak mudah dibaca oleh masyarakat yang berniat untuk melakukan pelanggaran.

Beberapa penelitian telah memecahkan masalah di atas, antara lain dengan mengaplikasikan *logika fuzzy* controller sebagai pengontrol lampu lalu lintas. Lampu lalu lintas diterapkan pada empat simpang dengan pola pergerakan yang diatur yaitu apabila arah timur dan barat mendapat giliran nyala lampu hijau maka pada arah utara dan selatan tidak bisa melewati simpangan demikian sebaliknya. Untuk mengetahui jumlah kendaraan pada setiap arah simpangan digunakan sensor yang menjadi *inputan* logika *fuzzy*. Penggunaan logika *fuzzy* pada lampu lalu lintas disimulasikan pada tools Matlab dengan metode Mamdani [1]. Rahmat Taufik membuat rancang bangun simulator kendali lampu lalu lintas dengan logika *fuzzy* berbasis *microcontroller*. Simulator yang dirancang terlebih dahulu di uji coba menggunakan Matlab dengan menggunakan metode Sugeno. Pada simulatornya terdapat switch yang digunakan sebagai sensor yang berfungsi sebagai penghitung kepadatan pada satu jalur. Pada simulator digunakan *microcontroller* sebagai pengendali lampu lalu lintasnya [2]. Buana Suhuruddin Putramembuat simulasi penerapan ANFIS pada sistem lampu lalu lintas enam ruas. Penelitian ini dilakukan untuk membandingkan waktu tunggu rata-rata yang dihasilkan dari pengaturan metode statis dengan metode dinamis menggunakan *neuro-fuzzy pada ANFIS* di persimpangan enam ruas[3]. Suhandi dkk, membuat simulasi antrian kendaraan pada persimpangan jalan berbasis *multiagent* menggunakan logika *fuzzy*. Penelitian yang dilakukan dengan membuat Game Simulasi saja yang berbasis *Multi Agent* cerdas, menggunakan logika *fuzzy*, dilengkapi dengan Algoritma Greedy [4].

Peneliti tertarik untuk membuat *prototype* sistem lampu lalu lintas adaptif yang dapat berkomunikasi antarlampu lalu lintas berbasis *multiagent* menggunakan logika *fuzzy* yang ditanam pada *microcontroller*. Sistem lampu lalu lintas yang dibuat dapat beradaptasi terhadap lingkungan persimpangan. Apabila terdapat sisi simpangan yang memiliki antrian kendaraan yang sangat banyak, maka waktu nyala lampu hijau lebih lama dibandingkan dengan sisi simpangan yang hanya memiliki antrian kendaraan yang sedikit. Dengan demikian, maka lampu lalu lintas lebih adaptif terhadap dinamisnya kendaraan yang akan melintasi persimpangan. Lampu lalu lintas juga dapat melakukan komunikasi dengan lampu lalu lintas tetangga terdekatnya secara dua arah. Komunikasi yang dilakukan yaitu saling memberi informasi mengenai jumlah kendaraan yang meninggalkan simpangannya menuju arah masing-masing simpangan tetangga terdekatnya. Dengan pola komunikasi yang terjadi, maka lampu lalu lintas akan bisa lebih dulu mengetahui jumlah kendaraan yang akan datang dari arah masing-masing sisi simpangan selain dari informasi sensor-sensor kedatangan.

## 1.2. Rumusan Masalah

- a. Sistem lampu lalu lintas yang statis dengan pola yang mudah terbaca tidak mampu mengurai kemacetan bahkan memunculkan peluang masyarakat untuk melanggar.
- b. Sistem lampu lalu lintas statis di atur secara bergilir sehingga tidak adaptif terhadap dinamisnya kondisi persimpangan.
- c. Sistem lampu lalu lintas statis belum mampu berkomunikasi antarsesama lampu lalu lintas.

## 1.3. Tujuan Penelitian

- a. Mengembangkan sistem kontrol lampu lalu lintas yang ada dengan sistem kontrol lampu lalu lintas berbasis *multiagent* menggunakan *logikafuzzy* yang ditanam pada *microcontroller*.
- b. Membuat *prototype* sistem lampu lalu lintas adaptif yang dapat berkomunikasi antarlampu lalu lintas berbasis *multiagent* menggunakan logika *fuzzy* yang ditanam pada *microcontroller*.
- c. Menguji kinerja *prototype* sistem lampu lalu lintas adaptif yang dapat berkomunikasi antarlampu lalu lintas berbasis *multiagent* menggunakan logika *fuzzy* yang ditanam pada *microcontroller*.

## 1.4. Manfaat Penelitian

- a. Terciptanya sistem lampu lalu lintas adaptif yang dapat berkomunikasi antarlampu lalu lintas berbasis *multiagent* menggunakan logika *fuzzy* yang ditanam pada *microcontroller*. Sehingga dapat mengurangi niat masyarakat untuk melanggar karena polanya yang selalu berubah-ubah.

- b. Perkembangan iptek yang diharapkan dari penelitian ini dapat menjadi pendukung *Advance Traffic Control System* (ATCS).
- c. Mengurangi jumlah petugas di jalan raya pada setiap persimpangan yang terdapat lampu lalu lintas.

## 2. TINJAUAN PUSTAKA

### 2.1. Penelitian Terkait

Simanjuntak N. P[1], membuat aplikasi *fuzzy logic controller* pada pengontrolan lampu lalu lintas memanfaatkan FIS (Fuzzy Inference System) dari perangkat lunak Matlab dengan menggunakan metode Mamdani. Untuk mendeteksi kepadatan atau jumlah mobil pada satu simpangan, diletakan sensor s1 dan s2 dari arah timur, s3 dan s4 dari arah selatan, s5 dan s6 dari barat, s7 dan s8 dari arah utara; s2,s4,s6 dan s8 adalah sensor pertama yang diposisikan di bagian depan. Untuk s1, s3, s5, dan s7 adalah sensor kedua yang diletakan di bagian belakang.

Taufik R, dkk [2],, melakukan penelitian rancang bangun simulator kendali lampu lalu lintas dengan logika *fuzzy* berbasis mikrokoroler. Untuk metode yang digunakan dalam logika *fuzzy* adalah metode Sugeno yang terlebih dahulu di uji cobakan pada Matlab. Pada variabel *input* adalah kepadatan pada jalur I, jalur II, dan jalur III sedangkan variabel *output* adalah lamanya waktu lampu hijau menyala untuk satu jalur.

Suhuruddin Putra. B dkk [3],, membuat simulasi penerapan *ANFIS* pada sistem lampu lalu lintas enam ruas. Penelitian dilakukan untuk membandingkan waktu tunggu rata-rata yang dihasilkan dari pengaturan metode statis dan metode dinamis dengan *neuro-fuzzy* menggunakan *ANFIS* di persimpangan enam ruas Penelitian yang dilakukan dengan membuat suatu model menggunakan perangkat lunak yang menerapkan *neuro-fuzzy* untuk menentukan durasi lampu hijau di persimpangan Pangkalan Jati, Kalimantan, Jakarta Timur.

Suhanda, dkk [4], , membuat game simulasi antrian kendaraan pada persimpangan jalan berbasis *multiagent* menggunakan logika *fuzzy* yang dilengkapi dengan algoritma greedy. Pada *inputnya*, variabel yang digunakan adalah kepadatan pada jalur I, jalur II, dan jalur III. Sedangkan *outputnya*, variabel yang digunakan adalah lamanya lampu hijau menyala untuk satu jalur.

Tomescu. O, dkk [5], , membuat sistem kontrol lampu lalu lintas adaptif menggunakan jaringan komunikasi antarkendaraan (ad hoc). Pada penelitian sistem lampu lalu lintas adaptif ini digunakan variabel seperti volume kendaraan, jenis kendaraan, dan kondisi persimpangan tetangga. Peneliti berfokus pada analisis korelasi antarperilaku pengemudi dan pergerakan kendaraan atau iring-iringan kendaraan. Pada penelitian ini, perhitungan nilai untuk pengaturan yang konstan digunakan simulator logika *fuzzy* pada Matlab, sedangkan sisanya menggunakan perhitungan matematika.

Royani. T, Et. al [6], melakukan penelitian tentang kontrol lampu lalu lintas pada persimpangan terisolasi dengan menggunakan *fuzzy neural network* dan algoritma genetika. Sebuah lampu lalu lintas yang terkontrol berbasis jaringan syaraf tiruan yang digunakan untuk kontrol optimum dari volume lalu lintas yang fluktuatif atau dari kondisi kendaraan jenuh menjadi kondisi biasa. Untuk menentukan interval waktu pada lampu lalu lintas digunakan logika *fuzzy*, untuk rumusnya dengan mengikuti prinsip manusia sebagai operator pada lampu lalu lintas. Untuk menyesuaikan parameter *fuzzy neural network*, digunakan genetika algoritma.

### 2.2. *Prototype* Lampu Lalu Lintas Adaptif Berbasis *Multi Agent* Menggunakan Logika *Fuzzy* yang Tertanam pada *Microcontroller*

#### 2.2.1 Fungsi Lampu Lalu Lintas

Lampu lalu lintas adalah suatu alat kendali (kontrol) dengan menggunakan lampu yang terpasang pada persimpangan dengan tujuan untuk mengatur arus lalu lintas. Pengaturan arus lalu lintas pada persimpangan pada dasarnya dimaksudkan untuk bagaimainama pergerakan kendaraan pada masing-masing kelompok pergerakan kendaraan (vehicle group movements) dapat bergerak secara bergantian sehingga tidak saling mengganggu antar arus yang ada. Ada berbagai jenis kendali dengan menggunakan

lampu lalu lintas. Pertimbangan ini sangat tergantung pada situasi dan kondisi persimpangan yang ada seperti volume, geometri simpang dan sebagainya [7].

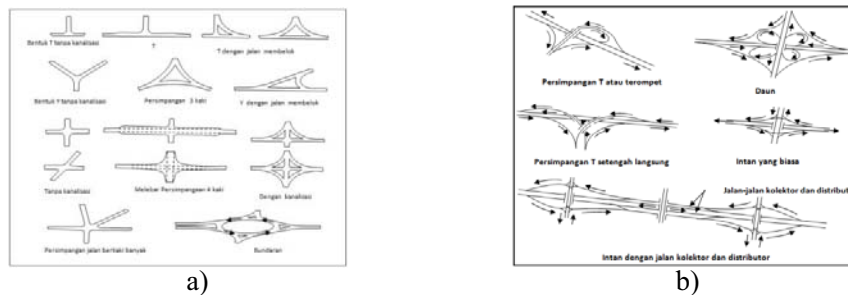
**2.2.2 Kondisi Persimpangan Saat Ini**

Sistem pengendalian lampu lalu lintas saat ini lebih banyak menggunakan pewaktuan yang statis. Pewaktuan statis tersebut telah ditentukan oleh pengontrol secara tetap untuk siklus nyala lampu merah, kuning dan hijau. Lampu lalu lintas tidak mampu mengetahui kondisi/kepadatan disetiap sisi-sisi simpangan. Pengaturan waktu lampu lalu lintas saat ini dilakukan secara bergilir. Akibatnya, dari sisi simpangan tertentu yang tidak memiliki antrian kendaraan, tetap harus ditunggu waktu lampu hijaunya menyala sebelum giliran didapatkan oleh sisi simpangan lain. Sehingga waktu yang dibutuhkan untuk menunggu giliran melewati persimpangan relatif lama. Dengan pola pengaturan yang demikian, masyarakat pengguna jalan dengan mudah membaca situasi dan kondisi yang bisa terjadi untuk pergerakan kendaraan. Akibatnya, memunculkan niat untuk melewati persimpangan atau melanggar isyarat lampu lalu lintas yang telah ditentukan.

a. Jenis Persimpangan

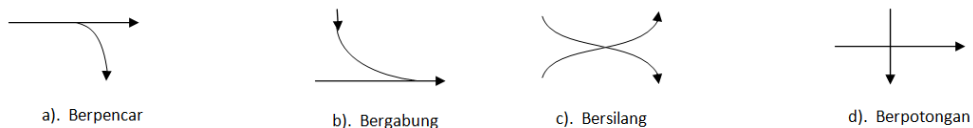
Persimpangan jalan adalah simpul pada jaringan jalan dengan ruas jalan yang bertemu dan lintasan arus kendaraan berpotongan. Lalu lintas pada masing-masing kaki persimpangan menggunakan ruang jalan pada persimpangan secara bersama-sama dengan lalu lintas lainnya. Olehnya itu persimpangan merupakan faktor yang paling penting dalam menentukan kapasitas dan waktu perjalanan pada suatu jaringan jalan khususnya di daerah - daerah perkotaan [8].

Persimpangan merupakan tempat sumber konflik lalu lintas yang rawan terhadap kecelakaan karena terjadi konflik antara kendaraan dengan kendaraan lainnya ataupun antara kendaraan dengan pejalan kaki. Oleh karena itu merupakan aspek penting didalam pengendalian lalu lintas. Secara garis besarnya persimpangan terbagi dalam 2 bagian yaitu : Pertama, persimpangan sebidang dan kedua persimpangan tak sebidang [8].



Gambar 1. a). Jenis Persimpangan Sebidang b). Persimpangan Tak Sebidang

Pergerakan arus lalu lintas pada persimpangan juga membentuk suatu manuver yang menyebabkan sering terjadi konflik dan tabrakan kendaraan. Pada dasarnya manuver dari kendaraan dapat dibagi atas 4 jenis, yaitu [10]:

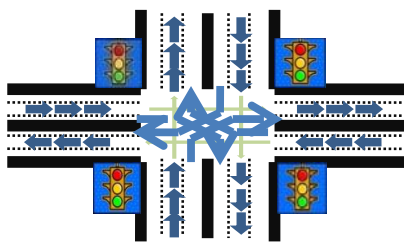


Gambar 2. Manuver Kendaraan

Penelitian dilakukan pada persimpangan sebidang dengan setiap simpang memiliki empat ruas. Menurut jenis fasilitasnya, pada persimpangan sebidang digunakan simpang bersinyal (signalised intersection) untuk mensimulasi *prototype* lampu lalu lintas adaptif berbasis *Multi Agent* dengan menggunakan logika *fuzzy* yang ditanam pada *microcontroller*.

b. Pola Pergerakan Arus Lalu Lintas

Berdasarkan jenis persimpangan yang digunakan yaitu persimpangan sebidang, untuk pergerakan arus lalu lintas di persimpangan pada *prototype* yang dibuat diperlihatkan pada gambar di bawah ini.



Gambar 3. Pola Pergerakan Arus Lalu Lintas Pada Persimpangan

Pergerakan arus lalu lintas pada persimpangan memiliki dua jalur pada setiap sisi simpangan. Setiap sisi terdapat dua jalur yang berlawanan arah, baik arah menuju simpangan maupun arah meninggalkan simpangan. Ada tiga manuver yang bisa dilakukan oleh kendaraan pada setiap melintasi simpangan. Untuk setiap kendaraan pada sisi persimpangan yang mendapatkan giliran melewati simpangan akan bisa melakukan manuver kekiri, lurus dan kekanan. Ketiga manuver tersebut hanya bisa dilakukan apabila mendapat isyarat nyala lampu hijau pada lampu lalu lintas dari masing-masing sisi simpangannya.

Agar pergerakan kendaraan pada persimpangan dua jalur tersebut lancar, maka diperlukan lampu lalu lintas adaptif. Supaya sistem yang dibuat dapat adaptif, maka sistem harus mampu belajar sendiri atau mempunyai kecerdasan. Sisten lampu lalu lintas yang polanya mampu menyesuaikan keadaan lingkungannya disebut dengan sistem cerdas.

Sistem cerdas atau kecerdasan buatan yang dimaksud disini merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia. Sementara pengertian cerdas adalah memiliki pengetahuan, pengalaman, dan penalaran untuk membuat keputusan dan mengambil tindakan. Jadi, agar mesin bisa cerdas (bertindak seperti manusia) maka harus diberi bekal pengetahuan dan diberi kemampuan untuk menalar [10]. Pada penelitian ini, algoritma cerdas yang digunakan yaitu logika *fuzzy* dan ditambahkan dengan sistem komunikasi *Multi Agent* sistem.

Logika *fuzzy* adalah metode sistem kontrol pemacahan masalah, yang cocok untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan PC, *multi-channel* atau *workstation* berbasis akuisisi data, dan sistem kontrol. Metode ini dapat diterapkan pada perangkat keras, perangkat lunak atau kombinasi keduanya [10].

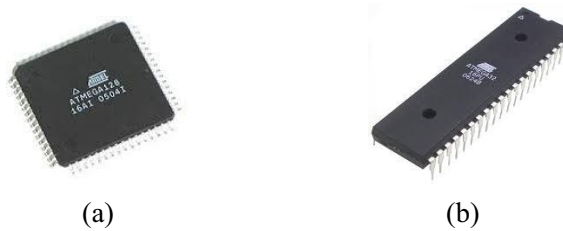
### 2.2.3 *Microcontroller*

Untuk membangun *prototype* sistem lampu lalu lintas yang adaptif dan relatif murah, maka diperlukan *microcontroller* sebagai pemrosesnya. Berbeda dengan *microprocessor*, *microcontroller* menggabungkan memori dan peripheral didalam sebuah kemasan (*Single chip*). *Microcontroller* dapat digunakan untuk kontrol aplikasi. Seiring dengan perkembangan teknologi dan kebutuhan yang tinggi akan mikrokontroller, maka perusahaan membuat mikrokontroller dengan kemampuan melakukan

pekerjaan yang rumit seperti pengolahan sinyal. Gambar di bawah ini merepresentasikan sebuah *microcontroller* atau singel chip yang di dalamnya terdapat sebuah mikroprosesor, rom, ram dan jalur *input output data*.

Kelebihan yang dapat dilihat jelas dari *microcontroller* adalah pada kekuatan program yang disimpan didalamnya. Tanpa *microcontroller* untuk melakukan perubahan disain suatu rangkaian kemungkinan besar dengan cara mengganti atau menyolder ulang komponen yang tepat. Tetapi dengan *microcontroller*, cukup merubah programnya. Sebuah system *microcontroller* dapat digunakan untuk melakukan berbagai macam pekerjaan, hanya dengan menyesuaikan program yang tertanam di dalamnya. Beberapa perusahaan yang membuat mikrokontroler antara lain adalah Atmel, Intel, Reinesance, Motorola, dan Zilog.

Berdasarkan banyaknya perusahaan yang membuat dan memproduksi *microcontroller*, maka ada banyak jenis *microcontroller* yang telah digunakan untuk berbagai macam keperluan perancangan sistem khususnya sistem kontrol. Atmel yang merupakan salah satu perusahaan yang telah banyak memproduksi dan memasarkan *microcontroller*. Salah satu jenis *microcontroller*nya yang populer saat ini adalah jenis AVR. *Microcontroller* AVR memiliki kelebihan-kelebihan dibandingkan dengan *microcontroller* lainnya. AVR menganut intruksi RISC (*Reduced Instruction Set Computer*) dimana sebagian besar kode instruksi dilakukan dengan hanya satu siklus clock. AVR dikelompokkan dalam 4 kelas. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral dan fungsinya. Keempat kelas tersebut adalah keluarga ATTiny, keluarga AT90Sxx, keluarga ATmega dan AT86RFxx. Pada setiap kelas AVR terdiri dari berbagai macam type. Pada penelitian ini, *microcontroller* yang digunakan adalah jenis AVR dengan kelas ATmega untuk type ATmega 128 dan ATmega32. Secara fisik *microcontroller* ATmega 128 dan ATmega 32 diperlihatkan pada gambar di bawah ini [11][12].



Gambar 4. *Microcontroller* (a) ATmega 128 (b) ATmega 32

#### 2.2.4 *Multi Agent System*

*Agent* adalah suatu entitas *Software* komputer yang memungkinkan *user* (pengguna) untuk mendelegasikan tugas kepadanya secara mandiri (autinomously) [14]. Sebuah agen adalah sebuah sistem komputer yang berada dalam suatu lingkungan dan memiliki kemampuan bertindak secara otonomos didalam situasi lingkungan tersebut sesuai dengan sasaran yang dirancang [13]. Sedangkan *Multi Agent* sistem adalah pengembangan sistem yang dalam suatu komunitas sistem terdapat beberapa *agent*, yang saling berinteraksi, bernegosiasi dan berkoordinasi satu sama lain dalam menjalankan pekerjaan [15]. Ada beberapa karakteristik dan atribut *agent* [15][16] sebagai berikut.

a. *Autonomy*

*Agent* dapat melakukan tugas secara mandiri dan tidak dipengaruhi secara langsung oleh *user*, *agent* lain ataupun oleh lingkungan (environment). Untuk mencapai tujuan dalam melakukan tugasnya secara mandiri, *agent* harus memiliki kemampuan kontrol terhadap setiap aksi yang mereka perbuat, baik aksi keluar maupun kedalam dan hal penting yang mendukung *Autonomy* adalah masalah intelegensi dari *agent*.

b. *Intelligence, reasoning, dan learning*

Standar minimum untuk bisa disebut *agent* yaitu intelegensia dan dalam konsep intelegensia, ada tiga komponen yang harus dimiliki yaitu internal *knowledge base*, kemampuan *reasoning* berdasar



pada *knowledge base* yang dimiliki, dan kemampuan *learning* untuk beradaptasi dalam perubahan lingkungan.

- c. *Mobility* dan *stationary*  
Khusus untuk *mobile agent*, dia harus memiliki kemampuan yang merupakan karakteristik tertinggi yang dimiliki yaitu mobilitas. Berkebalikan dari hal tersebut adalah *stationary agent*. Tetapi keduanya tetap harus memiliki kemampuan untuk mengirim pesan dan berkomunikasi dengan *agent* lain.
- d. *Delegation*  
*Agent* bergerak dalam kerangka menjalankan tugas yang diperintahkan oleh *user*. Proses pendelegasian ini adalah karakteristik utama suatu program disebut *agent*.
- e. *Reactivity*  
Karakteristik lain dari *agent* adalah kemampuan untuk bisa cepat beradaptasi dengan adanya perubahan informasi yang ada dalam suatu lingkungan. Lingkungan itu bisa mencakup : *agent* lain, *user*, adanya informasi dari luar, dsb.
- f. *Proactivity* dan *goal-oriented*  
*Agent* tidak hanya dituntut bisa beradaptasi terhadap perubahan lingkungan, tetapi juga harus mengambil inisiatif langkah penyelesaian apa yang harus diambil. Untuk itu *agent* harus didesain memiliki tujuan yang jelas, dan selalu berorientasi kepada tujuan yang diembannya.
- g. *Communication* dan *Coordination Capability*  
*Agent* harus mampu memiliki kemampuan berkomunikasi dengan *user* dan juga *agent* lain. Masalah komunikasi dengan *user* adalah masuk ke masalah *user interface* dan perangkatnya, sedangkan masalah komunikasi, koordinasi, dan kolaborasi dengan *agent* lain adalah masalah sentral penelitian *Multi Agent system* (MAS).

Ada banyak arsitektur *agent* yang telah dikemukakan dari berbagai peneliti. Secara umum ada beberapa arsitektur *agent*. *Agent* secara *black box*, *agent* mendapatkan *input* atau *perception* terhadap suatu masalah, kemudian bagian *intelligent processing* mengolah *input* tersebut sehingga bisa menghasilkan *output* berupa *action*.

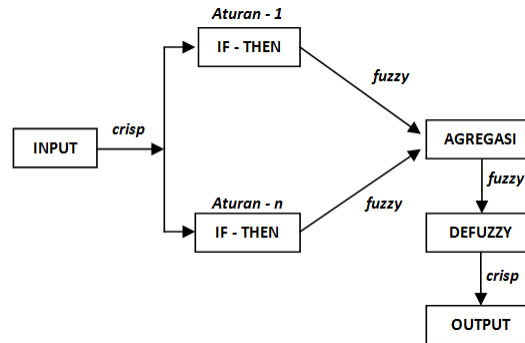


Gambar 5. Arsitektur *Software Agent* Secara *Black-Box*

### 2.2.5 Logika *Fuzzy*

Logika *fuzzy* merupakan salah satu cabang dari ilmu komputer yang mempelajari tentang nilai kebenaran yang bernilai banyak. Berbeda dengan logika klasik yang hanya memiliki nilai 0 (salah) atau 1 (benar). Logika *fuzzy* mempunyai nilai kebenaran real dalam rentang nilai (0-1) [1].

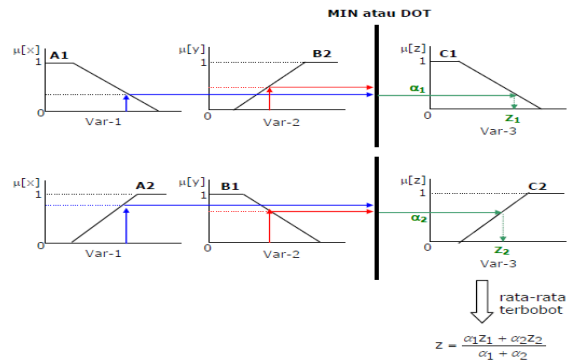
Untuk proses komputasi logika *fuzzy* digunakan *Fuzzy Inference System* (FIS). *Fuzzy Inference System* merupakan suatu kerangka komputasi yang didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* berbentuk IF – THEN, dan penalaran *fuzzy*. Dari gambaran di atas tampak pada diagram blok di bawah ini [18].



Gambar 6. Diagram Blok Sistem Inferensi Fuzzy

Sistem inferensi fuzzy menerima *input crisp*. *Input* ini kemudian dikirim ke basis pengetahuan yang berisi *n* aturan fuzzy dalam bentuk IF – THEN. *Fire strength* akan dicari pada setiap aturan. Apabila jumlah aturan lebih dari satu, maka akan dilakukan agregasi dari semua aturan. Selanjutnya, pada hasil agregasi akan dilakukan defuzzy untuk mendapatkan nilai *crisp* sebagai *output* sistem. Metode sistem inferensi dalam logika fuzzy yang digunakan dalam penelitian ini adalah metode Tsukamoto dengan metode rata-rata terbobot.

Pada Metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk IF-Then harus direpresentasikan dengan suatu himpunan fuzzy dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, *output* hasil inferensi dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan  $\alpha$ - predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.



Gambar 7. Inferensi dengan Menggunakan Metode Tsukamoto

### 3. METODE PENELITIAN

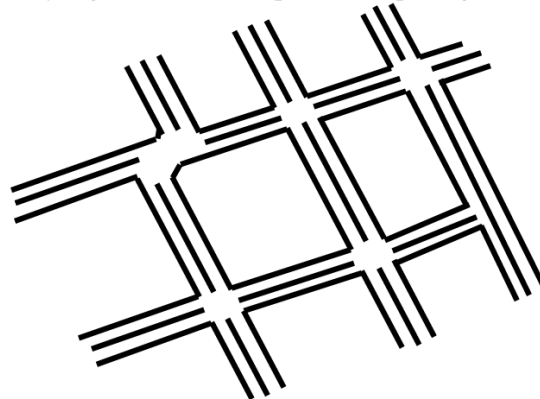
#### 3.1. Perancangan Prototype

##### a. Rancangan Prototype Lampu Lalu Lintas dan Jalan

Prototype yang dirancang akan merepresentasikan bentuk jalan-jalan persimpangan pada Kota. Pada prototype ini diletakan masing-masing lampu lalu lintas pada setiap simpangan empat, dengan asumsi bahwa pada setiap persimpangan Kota memiliki lampu lalu lintas. Setiap lampu lalu lintas yang ada pada persimpangan, digunakan Led sebagai penanda atau sinyal yang mengatur pergerakan kendaraan pada masing-masing sisi persimpangan. Warna Led yang digunakan



merepresentasikan warna lampu lalu lintas pada umumnya yaitu merah, kuning dan hijau. Dalam mengatur nyala Led tersebut, *microcontroller* akan mengirimkan sinyal ke led berdasarkan hasil pengolahan *inputan* sensor atau banyak sedikitnya kendaraan pada sisi-sisi persimpangan. Adapun gambar *prototype* jalan yang akan dibuat dapat dilihat pada gambar di bawah ini.



Gambar 8. Rancangan *Prototype* Jalan

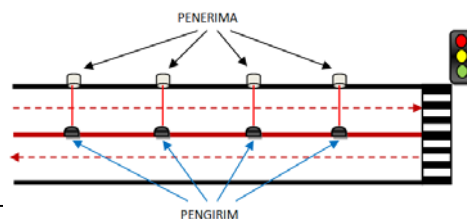
b. Rancangan Sensor Kedatangan dan Kepergian

Untuk mendeteksi kendaraan dari setiap sisi-sisi persimpangan digunakan sensor. Pada *prototype* lampu lalu lintas yang dibuat terdapat dua posisi penempatan sensor. Posisi pertama, sensor diletakan pada jalur kedatangan kendaraan pada setiap sisi simpangan. Posisi kedua, sensor diletakan pada jalur kepergian kendaraan untuk setiap sisi simpangan. Sensor yang digunakan adalah sensor proximity atau sensor berbasis cahaya. Terdapat dua bagian pada sensor yaitu pemancar dan penerima. Pada bagian pemancarnya digunakan sinar laser yang berfungsi mengirim cahaya dan pada bagian penerimanya digunakan Light Dependent Resistor (LDR) yang akan menerima cahaya.



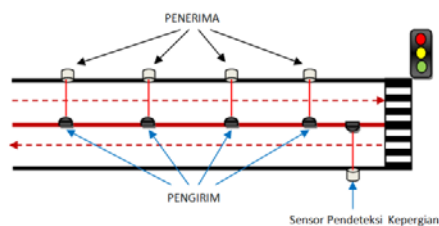
Gambar 9. Ilustrasi Pengirim Sinar Laser dan Penerima LDR

Pada saat kendaraan melintas dan menghalangi cahaya dalam waktu yang relatif lama maka akan dianggap ada kendaraan yang sedang mengantri. Untuk mengetahui seberapa panjang antrian kendaraan pada setiap sisi simpangan, maka akan diletakan beberapa sensor pada setiap sisi jalan pada jalur kedatangan kendaraan di persimpangan. Jumlah sensor yang terpasang menyatakan banyaknya himpunan *fuzzy* yang digunakan. Sensor yang digunakan sebanyak empat untuk mewakili masing-masing himpunan *fuzzy*. Pada himpunan *fuzzy* yang dipakai adalah sedikit, sedang, banyak, dan sangat banyak. Sensor akan memberikan informasi untuk dijadikan *inputan* nilai *crisp* pada proses awal *fuzzyfikasi*. Sensor akan diletakan dengan posisi urutan dari jarak terdekat dengan lampu lalu lintas untuk informasi jumlah kendaraan pada himpunan *fuzzy* sedikit sampai jarak terjauh dari lampu lalu lintas untuk informasi jumlah kendaraan yang padat pada himpunan *fuzzy* sangat banyak. Adapun posisi penempatan sensor pada salah satu sisi simpangan diperlihatkan pada gambar di bawah ini.



Gambar 10. Urutan Posisi Sensor Pada Salah Satu Sisi Simpangan

Selain kendaraan yang datang pada persimpangan, kendaraan yang meninggalkan persimpangan juga akan dideteksi oleh sensor. Jenis sensor yang dipakai sama dengan sensor yang digunakan pada bagian jalur kedatangan kendaraan. Posisi peletakan sensor untuk mendeteksi kendaraan yang meninggalkan persimpangan berada pada jalur yang bersebelahan dengan jalur kedatangan kendaraan. Setiap jalur kepergian kendaraan atau yang meninggalkan persimpangan terdapat minimal satu sensor yang dipasang. Sehingga dengan adanya sensor ini akan bisa diketahui berapa dan kemana arah kendaraan tersebut meninggalkan persimpangan. Untuk posisi peletakan sensor kepergian kendaraan ini dapat dilihat pada gambar di bawah ini.



Gambar 11. Posisi Sensor Pendeteksi Kepergian Kendaraan

c. Rancangan Kendali *Microcontroller Slave* dan *Master*

Pada penelitian ini, untuk pengaturan lampu lalu lintas pada setiap persimpangan digunakan *microcontroller* sebagai pemrosesnya. *Microcontroller* akan mengolah informasi yang ada berdasarkan hasil pendeteksian sensor, baik sensor yang terpasang pada jalur kedatangan kendaraan maupun sensor yang ada pada jalur kepergian kendaraan. Agar hasil pendeteksian sensor bisa diketahui secara akurat, maka penggunaan *microcontroller* sebagai pengolah dibagi menjadi dua bagian, yaitu bagian *Slave* dan *Master*. Jenis *microcontroller Master* digunakan ATmega 128 dan jenis *microcontroller Slave* digunakan ATmega 32. *Microcontroller Slave* akan berfungsi menerima dan mengolah informasi dari sensor-sensor yang ada pada setiap sisi persimpangan, baik pada jalur kedatangan maupun kepergian kendaraan. Untuk hubungan *microcontroller* dengan sensor yang mendeteksi kedatangan dan kepergian kendaraan pada setiap sisi-sisi persimpangan tampak pada gambar di bawah ini.



Gambar 12. Hubungan Sensor dengan *Microcontroller*

*Microcontroller* akan menerima data dari sensor, baik kendaraan yang datang maupun yang pergi. Data yang diperoleh terlebih dahulu dihitung dan disimpan lalu dikirim ke *microcontroller Master*. Pada *microcontroller Master* akan mengolah hasil informasi dari *microcontroller Slave* untuk memutuskan berapa lama durasi untuk lampu hijau, merah dan kuningnya. Untuk hubungan komunikasi antara *microcontroller Master* dengan *microcontroller Slave* diperlihatkan pada gambar berikut ini.



Gambar 13. Bagan Komunikasi Mikrokotroler *Master* dan *Slave*

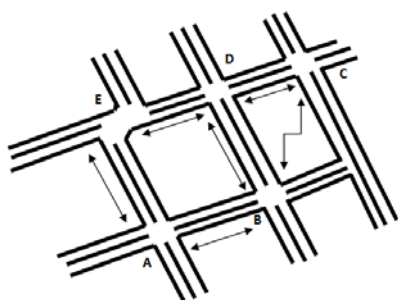
*Microcontroller Slave* pada setiap sisi-sisi persimpangan akan mengirimkan data informasi yang terdeteksi oleh sensor kedatangan maupun sensor kepergian kendaraan. *Microcontroller Master* mengolah informasi data tersebut untuk menentukan durasi lampu hijau dan merah serta sisi mana yang akan dilewatkan dari kondisi yang dianggap memiliki antrian panjang. Selain itu, *microcontroller Master* akan mengetahui berapa jumlah dan kemana arah pergerakan kendaraan meninggalkan persimpangan, lalu hasil dari perhitungan tersebut akan dikirimkan ke setiap simpangan tetangga terdekat dari empat arah yang ada di simpangan dengan sistem *Multi Agent*. Untuk sistem secara keseluruhan dapat dilihat pada gambar di bawah ini.



Gambar 14. Bagan Sistem Lampu Lalu Lintas Adaptif

d. Pola Komunikasi *Agent* AntarLampu Lalu Lintas

Untuk satu persimpangan diketahui oleh persimpangan lain dalam kondisi padat atau tidak, maka lampu lalu lintas satu dengan lampu lalu lintas tetangga akan melakukan komunikasi. Sistem komunikasi yang dilakukan berbasis *multiagent*. *Agent* yang ada pada setiap lampu lalu lintas akan saling berkomunikasi dan berkoordinasi dengan prinsip satu lampu lalu lintas hanya bisa berkomunikasi dengan maksimal empat lampu lalu lintas tetangganya. Jadi, lingkup *agent* yang ada pada satu lampu lalu lintas tidak dapat melakukan komunikasi lebih dari empat *agent* pada lampu lalu lintas lain. Komunikasi antar*agent* dibatasi hanya pada prinsip ketetanggaan terdekat dari setiap arah jalur persimpangan atau lampu lalu lintas tidak dapat berkomunikasi dengan lampu lalu lintas yang ada setelah tetangganya.



Tabel 1. Komunikasi *Agent* AntarTitik Lampu Lalu Lintas

Titik Lampu Lalu Lintas	Titik Komunikasi Agent
A	B, E
B	A, C, D
C	B, D
D	C, E
E	D, A

Gambar 15. Pola Komunikasi *Agent* Pada Lampu Lalu Lintas

Seperti yang diperlihatkan pada gambar di atas, hubungan komunikasi yang dilakukan antarpersimpangan secara dua arah. Pola hubungan komunikasi antarlampu lalu lintas pada *prototype* jalan yang dibuat diperlihatkan pada tabel di atas.

### 3.2. Penerapan dan Pengujian *Fuzzy Inferensi Sistem (FIS)*

Pada penelitian ini, untuk penerapan *fuzzy* digunakan *Fuzzy Inference System Metode Tsukamoto*, sementara untuk metode pengujiannya dengan cara membandingkan hasil kinerja yang dilakukan oleh *prototype* lampu lalu lintas dalam menentukan durasi nyala lampu hijau dengan perhitungan menggunakan aplikasi Microsoft Excel 2007.

*Inputan* nilai tegas didapatkan dari sensor kedatangan yang menjadi data untuk proses *fuzzyfikasi* pada *microcontroller Slave* dengan himpunan *fuzzy* yang sama pada setiap jalur. Fungsi keanggotaan yang digunakan adalah segitiga. Untuk jalur 1, jalur 2, dan jalur 3 memiliki himpunan sedikit, sedang, banyak dan sangat banyak. Adapun domain nilai masing-masing himpunan sebagai berikut.

a. Proses Fuzifikasi

Himpunan Sedikit

$$\mu[z_0]_{Sedikit} = \begin{cases} (10 - x)/(10 - 0); & 0 \leq x < 10 \\ 0; & x \geq 10 \end{cases}$$

Himpunan Sedang

$$\mu[z_1]_{Sedang} = \begin{cases} 0; & x \leq 2.5 \text{ atau } x \geq 17.5 \\ (x - 2.5)/(10 - 2.5); & 2.5 \leq x \leq 10 \\ (10 - x)/(17.5 - 10); & 10 \leq x \leq 17.5 \end{cases}$$

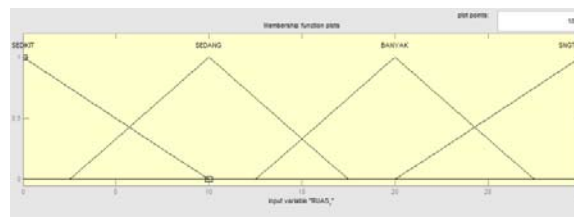
Himpunan Banyak

$$\mu[z_2]_{Banyak} = \begin{cases} 0; & x \leq 12.5 \text{ atau } x \geq 27.5 \\ (x - 12.5)/(20 - 12.5); & 12.5 \leq x \leq 20 \\ (20 - x)/(20 - 27.5); & 20 \leq x \leq 27.5 \end{cases}$$

Himpunan Sangat Banyak

$$\mu[z_3]_{Sangat\_Banyak} = \begin{cases} 0; & x \leq 20 \\ (x - 20)/(30 - 20); & 20 \leq x \leq 30 \\ 1; & x \geq 30 \end{cases}$$

Grafik fungsi keanggotaan dari setiap himpunan diperlihatkan pada gambar berikut.



Gambar 16. Grafik Fungsi Keanggotaan Segitiga Jalur 1

b. *Output* waktu lampu hijau

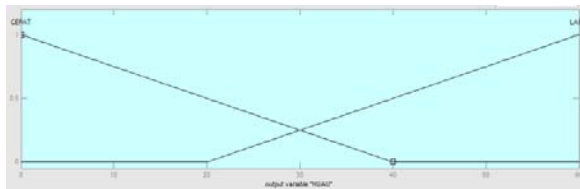
Himpunan Cepat

$$\mu[z_0]_{Cepat} = \begin{cases} (40 - x)/(40 - 0); & 0 \leq x < 40 \\ 0; & x \geq 40 \end{cases}$$

Himpunan Lama

$$\mu[z_1]_{Lama} = \begin{cases} 0; & x \leq 20 \\ (x - 20)/(60 - 20); & 20 \leq x \leq 60 \\ 1; & x \geq 60 \end{cases}$$

Grafik fungsi keanggotaan untuk variabel waktu untuk masing-masing himpunan dapat dilihat pada gambar di bawah ini.



Gambar 17. Grafik Fungsi Keanggotaan Segitiga Untuk Variabel Waktu

c. *Agregasi Rule*

Proses agregasi yang dilakukan berdasarkan kemungkinan pembacaan sensor pada setiap sisi simpangan. Dimana pembacaan sensor merupakan anteseden dari setiap jalur dan konsekuennya adalah variabel waktu yang dihasilkan untuk lamanya lampu hijau menyala. Terdapat 40 aturan yang digunakan. Untuk proses agregasi mengacu pada aturan berikut ini.

- 1) If (Jalur 1 is Sedikit) dan (Jalur 2 is Sedikit) dan (Jalur 3 is Sedikit) Then (Hijau is Cepat)
- 2) If (Jalur 1 is Sedikit) dan (Jalur 2 is Sedikit) dan (Jalur 3 is Sedang) Then (Hijau is Cepat)
- 3) If (Jalur 1 is Sedikit) dan (Jalur 2 is Sedikit) dan (Jalur 3 is Banyak) Then (Hijau is Lama)
- 4) If (Jalur 1 is Sedikit) dan (Jalur 2 is Sedikit) dan (Jalur 3 is Sangat\_Banyak) Then (Hijau is Lama)
- 5) If (Jalur 1 is Sedang) dan (Jalur 2 is Sedikit) dan (Jalur 3 is Sedikit) Then (Hijau is Cepat)
- 6) If (Jalur 1 is Banyak) dan (Jalur 2 is Sedikit) dan (Jalur 3 is Sedikit) Then (Hijau is Lama)
- 7) If (Jalur 1 is Sangat\_Banyak) dan (Jalur 2 is Sedikit) dan (Jalur 3 is Sedikit) Then (Hijau is Lama)
- 8) If (Jalur 1 is Sedikit) dan (Jalur 2 is Sedang) dan (Jalur 3 is Sedikit) Then (Hijau is Cepat)
- 9) If (Jalur 1 is Sedikit) dan (Jalur 2 is Banyak) dan (Jalur 3 is Sedikit) Then (Hijau is Lama)
- 10) If (Jalur 1 is Sedikit) dan (Jalur 2 is Sangat\_Banyak) dan (Jalur 3 is Sedikit) Then (Hijau is Lama)
- 11) If (Jalur 1 is Sedang) dan (Jalur 2 is Sedang) dan (Jalur 3 is Sedikit) Then (Hijau is Cepat)
- 12) If (Jalur 1 is Sedang) dan (Jalur 2 is Sedang) dan (Jalur 3 is Sedang) Then (Hijau is Cepat)
- 13) If (Jalur 1 is Sedang) dan (Jalur 2 is Sedang) dan (Jalur 3 is Banyak) Then (Hijau is Lama)

- 14) If (Jalur 1 is Sedang) dan (Jalur 2 is Sedang) dan (Jalur 3 is Sangat\_Banyak) Then (Hijau is Lama)
- 15) If (Jalur 1 is Sedikit) dan (Jalur 2 is Sedang) dan (Jalur 3 is Sedang) Then (Hijau is Cepat)
- 16) If (Jalur 1 is Banyak) dan (Jalur 2 is Sedang) dan (Jalur 3 is Sedang) Then (Hijau is Lama)
- 17) If (Jalur 1 is Sangat\_Banyak) dan (Jalur 2 is Sedang) dan (Jalur 3 is Sedang) Then (Hijau is Lama)
- 18) If (Jalur 1 is Sedang) dan (Jalur 2 is Sedikit) dan (Jalur 3 is Sedang) Then (Hijau is Cepat)
- 19) If (Jalur 1 is Sedang) dan (Jalur 2 is Banyak) dan (Jalur 3 is Sedang) Then (Hijau is Lama)
- 20) If (Jalur 1 is Sedang) dan (Jalur 2 is Sangat\_Banyak) dan (Jalur 3 is Sedang) Then (Hijau is Lama)
- 21) If (Jalur 1 is Banyak) dan (Jalur 2 is Banyak) dan (Jalur 3 is Sedikit) Then (Hijau is Lama)
- 22) If (Jalur 1 is Banyak) dan (Jalur 2 is Banyak) dan (Jalur 3 is Sedang) Then (Hijau is Lama)
- 23) If (Jalur 1 is Banyak) dan (Jalur 2 is Banyak) dan (Jalur 3 is Banyak) Then (Hijau is Lama)
- 24) If (Jalur 1 is Banyak) dan (Jalur 2 is Banyak) dan (Jalur 3 is Sangat\_Banyak) Then (Hijau is Lama)
- 25) If (Jalur 1 is Sedikit) dan (Jalur 2 is Banyak) dan (Jalur 3 is Banyak) Then (Hijau is Lama)
- 26) If (Jalur 1 is Sedang) dan (Jalur 2 is Banyak) dan (Jalur 3 is Banyak) Then (Hijau is Lama)
- 27) If (Jalur 1 is Sangat\_Banyak) dan (Jalur 2 is Banyak) dan (Jalur 3 is Banyak) Then (Hijau is Lama)
- 28) If (Jalur 1 is Banyak) dan (Jalur 2 is Sedikit) dan (Jalur 3 is Banyak) Then (Hijau is Lama)
- 29) If (Jalur 1 is Banyak) dan (Jalur 2 is Sedang) dan (Jalur 3 is Banyak) Then (Hijau is Lama)
- 30) If (Jalur 1 is Banyak) dan (Jalur 2 is Sangat\_Banyak) dan (Jalur 3 is Banyak) Then (Hijau is Lama)
- 31) If (Jalur 1 is Sangat\_Banyak) dan (Jalur 2 is Sangat\_Banyak) dan (Jalur 3 is Sedikit) Then (Hijau is Lama)
- 32) If (Jalur 1 is Sangat\_Banyak) dan (Jalur 2 is Sangat\_Banyak) dan (Jalur 3 is Sedang) Then (Hijau is Lama)
- 33) If (Jalur 1 is Sangat\_Banyak) dan (Jalur 2 is Sangat\_Banyak) dan (Jalur 3 is Banyak) Then (Hijau is Lama)
- 34) If (Jalur 1 is Sangat\_Banyak) dan (Jalur 2 is Sangat\_Banyak) dan (Jalur 3 is Sangat\_Banyak) Then (Hijau is Lama)
- 35) If (Jalur 1 is Sedikit) dan (Jalur 2 is Sangat\_Banyak) dan (Jalur 3 is Sangat\_Banyak) Then (Hijau is Lama)
- 36) If (Jalur 1 is Sedang) dan (Jalur 2 is Sangat\_Banyak) dan (Jalur 3 is Sangat\_Banyak) Then (Hijau is Lama)
- 37) If (Jalur 1 is Banyak) dan (Jalur 2 is Sangat\_Banyak) dan (Jalur 3 is Sangat\_Banyak) Then (Hijau is Lama)
- 38) If (Jalur 1 is Sangat\_Banyak) dan (Jalur 2 is Sedikit) dan (Jalur 3 is Sangat\_Banyak) Then (Hijau is Lama)



- 39) If (Jalur 1 is Sangat\_Banyak) dan (Jalur 2 is Sedang) dan (Jalur 3 is Sangat\_Banyak) Then (Hijau is Lama)
- 40) If (Jalur 1 is Sangat\_Banyak) dan (Jalur 2 is Banyak) dan (Jalur 3 is Sangat\_Banyak) Then (Hijau is Lama)

d. Defuzifikasi

Proses defuzifikasi dilakukan dengan menggunakan metode Tsukamoto. Hasil inference Tsukamoto diperoleh dengan menggunakan rata-rata terbobot. Z merupakan lamanya lampu hijau menyala pada jalur yang terbuka.

$$Z = \frac{z1*apred1+z2*apred2+z3*apred3+.....+z40*apred40}{apred1+apred2+apred3+.....+apred40}$$

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengujian *Prototype* Jalan dan Lampu Lalu Lintas

Berdasarkan penelitian yang telah dilakukan, gambar di bawah ini merupakan hasil rancangan *prototype* jalan dan lampu lalu lintas adaptif berbasis *multiagent* menggunakan logika *fuzzy*.



Gambar 18. a) *Prototype* Jalan b) Lampu Lalu Lintas Pada Persimpangan

Tabel 2. Hasil Pengujian Lampu Lalu Lintas

Sisi Simpangan	Nyala Lampu			Waktu
	Merah	Kuning	Hijau	
Jalur Terbuka			Nyala	20
Jalur 1	Nyala			40
Jalur 2	Nyala			20

4.2. Hasil Pengujian Sensor

Dari rancangan sensor yang telah dibuat, sensor dapat mendeteksi kendaraan baik yang datang maupun yang pergi. Kendaraan dapat dideteksi dengan baik oleh sensor karena kemampuan sensitifitasnya dapat diatur. Posisi sensor diletakan di jalur yang berbeda pada setiap sisi simpangan. Sehingga kendaraan dapat diketahui baik yang datang maupun yang pergi. Dari hasil, sensor diperlihatkan pada gambar di bawah ini.



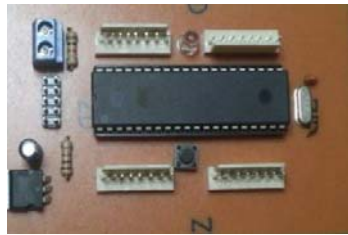
a) b) c)  
Gambar 19. a) Modul Sensor, b) Posisi Sensor Kedatangan, c) Posisi Sensor Kepergian

Tabel 3. Hasil Uji Coba Sensor Kedatangan Berdasarkan Himpunan *Fuzzy* Pada Satu Sisi Simpangan

Titik Sensor	Jumlah Kendaraan Yang Terdeteksi	Himpunan <i>Fuzzy</i>	Derajat Keanggotaan
1	7	Sedikit	0.30
2	10	Sedang	1.00
3	15	Banyak	0.33
4	25	Sangat banyak	0.50

#### 4.3. Hasil Pengujian Kendali *Microcontroller*

*Microcontroller* yang digunakan pada setiap titik lampu lalu lintas ada dua. *Microcontroller* yang berfungsi sebagai *Slave* adalah jenis *microcontroller* AVR Atmega 32 dan *microcontroller* yang berfungsi sebagai *Master* adalah jenis mikorokontroler AVR Atmega 128. Pada *microcontroller* Atmega 32 yang terintegrasi dengan sensor kedatangan maupun dengan sensor kepergian dan diprogram untuk melakukan proses fuzifikasi, mengirim dan menerima informasi dari *agent* pada titik lampu lalu lintas tetangga. Berikut gambar modul *microcontroller Slave* (Atmega 32) dan *microcontroller Master* (Atmega 128).



a)



b)

Gambar 20. a) Modul *Microcontroller Slave* (Atmega 32), b) *Microcontroller Master* (Atmega 128)

Tabel 4. Hasil Pengujian *Microcontroller Slave* (Atmega 32)

Sensor Pada Satu Sisi Simpangan	Jumlah Yang Terdeteksi	Hasil Fuzifikasi (derajat keanggotaan)	Pengiriman Informasi ( <i>Agent</i> )	Penerimaan Informasi ( <i>Agent</i> )
A. Kedatangan				
1	10	0.00		
2	15	0.33		
3	30	1.00		
B. Kepergian				
1	5		5	10
2	10		10	6
3	20		20	15

Tabel 5. Hasil Pengujian *Microcontroller Master* (Atmega 128)

C. Sisi simpangan	D. Himpunan Fuzzy dan Derajat Keanggotaan Pada Masing-Masing Jalur												E. Waktu F. Hijau
	G. edikit	H. edang	I. banyak	J. angkat K. banyak	L. edikit	M. edang	N. banyak	O. angkat P. banyak	Q. edikit	R. edang	S. banyak	T. angkat U. banyak	
V. Jalur 1	W. .0	X. .87	Y. .0	Z. .0	AA.	BB.	CC.	DD.	EE.	FF.	GG.	HH.	II.
JJ. Jalur 2	KK.	LL.	MM.	NN.	OO. .0	PP. .0	QQ. .0	RR. .0	SS.	TT.	UU.	VV.	WW.
XX. alur 3	YY.	ZZ.	AAA.	BBB.	CCC.	DDD.	EEE.	FFF.	GGG. .0	HHH. .0	III. .0	JJJ. .0	KKK.
LLL. alur Yang Dibuka	MMM.	NNN.	OOO.	PPP.	QQQ.	RRR.	SSS.	TTT.	UUU.	VVV.	WWW.	XXX.	YYY. 0. 456
ZZ. alur 1	AAAA. .0	BBBB. .0	CCCC. .07	DDDD. .0	EEEE.	FFFF.	GGGG.	HHHH.	IIII.	JJJJ.	KKKK.	LLLL.	MMMM.
NNNN. alur 2	OOOO.	PPPP.	QQQQ.	RRRR.	SSSS. .0	TTTT. .0	UUUU.	VVVV. .0	WWWW.	XXXX.	YYYY.	ZZZZ.	AAAA.
BBBB. alur 3	CCCC.	DDDD.	EEEE.	FFFF.	GGGG.	HHHH.	IIII.	JJJJ.	KKKK. .0	LLLL. .0	MMMM. .0	NNNN. .0	OOOO.
PPPP. alur Yang Dibuka	QQQQ.	RRRR.	SSSS.	TTTT.	UUUU.	VVVV.	WWWW.	XXXX.	YYYY.	ZZZZ.	AAAA.	BBBB.	CCCC. 3. 604
DDDD. alur 1	EEEE. .0	FFFF. .0	GGGG. .0	HHHH. .8	IIII.	JJJJ.	KKKK.	LLLL.	MMMM.	NNNN.	OOOO.	PPPP.	QQQQ.
RRRR. alur 2	SSSS.	TTTT.	UUUU.	VVVV. .0	WWWW.	XXXX. .0	YYYY. .0	ZZZZ. .0	AAAA.	BBBB.	CCCC.	DDDD.	EEEE.
FFFF. alur 3	GGGG.	HHHH.	IIII.	JJJJ.	KKKK.	LLLL.	MMMM.	NNNN.	OOOO. .0	PPPP. .0	QQQQ. .0	RRRR. .0	SSSS.
TTTT. alur Yang Dibuka	UUUU.	VVVV.	WWWW.	XXXX.	YYYY.	ZZZZ.	AAAA.	BBBB.	CCCC.	DDDD.	EEEE.	FFFF.	GGGG. 7. 000
HHHH. alur 1	IIII. .0	JJJJ. .2	KKKK. .0	LLLL. .0	MMMM.	NNNN.	OOOO.	PPPP.	QQQQ.	RRRR.	SSSS.	TTTT.	UUUU.
VVVV. alur 2	WWWW.	XXXX.	YYYY.	ZZZZ.	AAAA. .0	BBBB. .0	CCCC. .6	DDDD. .0	EEEE.	FFFF.	GGGG.	HHHH.	IIII.
JJJJ. alur 3	KKKK.	LLLL.	MMMM.	NNNN.	OOOO.	PPPP.	QQQQ.	RRRR.	SSSS. .0	TTTT. .0	UUUU. .87	VVVV. .0	WWWW.
XXXX. alur Yang Dibuka	YYYY.	ZZZZ.	AAAA.	BBBB.	CCCC.	DDDD.	EEEE.	FFFF.	GGGG.	HHHH.	IIII.	JJJJ.	KKKK. 2. 358
LLLL. alur 1	MMMM. .0	NNNN. .73	OOOO. .0	PPPP. .0	QQQQ.	RRRR.	SSSS.	TTTT.	UUUU.	VVVV.	WWWW.	XXXX.	YYYY.
ZZZZ. alur 2	AAAA.	BBBB.	CCCC.	DDDD.	EEEE. .0	FFFF. .87	GGGG. .0	HHHH. .0	IIII.	JJJJ.	KKKK.	LLLL.	MMMM.
NNNN. alur 3	OOOO.	PPPP.	QQQQ.	RRRR.	SSSS.	TTTT.	UUUU.	VVVV.	WWWW. .7	XXXX. .0	YYYY. .0	ZZZZ. .0	AAAA.
BBBB. alur Yang Dibuka	CCCC.	DDDD.	EEEE.	FFFF.	GGGG.	HHHH.	IIII.	JJJJ.	KKKK.	LLLL.	MMMM.	NNNN.	OOOO. 2. 000

4.4. Hasil Pengujian Komunikasi *Agent* AntarLampu Lalu Lintas

Dari pengujian yang dilakukan, diperoleh hasil komunikasi *agent* antarlampu lalu lintas seperti pada tabel berikut.

Tabel 6. Hasil Pengujian Komunikasi *Agent* AntarLampu Lalu Lintas

Titik Lampu Lalu Lintas	Titik Komunikasi <i>Agent</i>	Pengiriman Informasi ( <i>agent</i> )	Penerimaan Informasi ( <i>agent</i> )
"A" ke	B	10	7
	E	14	16
"B" ke	A	7	10
	C	9	21
	D	10	17
"C" ke	B	13	27

	D	3	7
"D" ke	B	5	18
	C	12	16
	E	22	19
"E" ke	D	10	28
	A	11	10

4.5. Hasil Pengujian Kinerja Logika *Fuzzy*

Tabel 7. Hasil Pengujian Logika *Fuzzy*  
Berdasarkan Jumlah Kendaraan Pada Jalur 1, Jalur 2 dan Jalur 3

No	Jalur 1	Jalur 2	Jalur 3	Waktu Hijau Pada Excel	Waktu Hijau <i>Microcontroller</i>
1	9	20	20	40.465	40
2	13	20	20	43.604	43
3	28	20	20	47.000	47
4	8	9	3	12.000	12
5	16	9	3	16.444	16
6	16	21	3	17.200	17
7	16	29	3	30.250	30
8	16	17	19	32.358	32

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil penelitian yang dilakukan, berdasarkan *prototype* yang sudah dirancang dan di uji, *prototype* lampu lalu lintas adaptif berbasis *multiagent* menggunakan logika *fuzzy* dapat disimpulkan :

- Bekerja dengan baik dan adaptif terhadap jumlah kendaraan pada sisi-sisi simpangan.
- Mengurai kemacetan pada simpangan yang memiliki arus kendaraan yang dinamis.
- Pola nyala lampu merah dan hijau menyesuaikan dengan kondisi kendaraan pada sisi-sisi simpangan sehingga tidak mudah terbaca oleh pengendara (masyarakat/pengguna jalan).
- Kinerja logika *fuzzy* yang ditanam pada *microcontroller* dapat berfungsi sebagaimana mestinya dan dibandingkan dengan perhitungan pada Excel didapatkan nilai yang sama.
- Multiagent* sistem yang diterapkan dapat berkomunikasi dan saling memberi informasi pada setiap titik simpangan dengan simpangan yang lain yang merupakan simpangan tetangga terdekat.

5.2. Saran

Dari pengujian *prototype* lampu lalu lintas adaptif berbasis *multiagent* menggunakan logika *fuzzy* ini terdapat kekurangan yang perlu disempurnakan guna mendapatkan hasil yang lebih baik apabila akan di implementasikan pada lingkungan persimpangan yang sebenarnya. Adapun kekurangan tersebut, disarankan untuk dikembangkan khusus pada bagian berikut.

- Sensor kedatangan dan kepergian kendaraan yang digunakan masih bersifat on dan off ( ada atau tidak ada/dalam nilai biner 0 dan 1). Disarankan untuk perancangan selanjutnya agar dapat mendeteksi kendaraan secara presisi terhadap jumlah kendaraan pada sisi-sisi simpangan. Misalnya dengan menggunakan sensor kamera yang mampu mendeteksi kendaraan secara citra. Dengan menggunakan kamera dapat diketahui melalui pergerakan kendaraan (motion detection). Sehingga keakuratan dari sisi jumlah kendaraan yang datang maupun yang pergi diperoleh dengan baik. Namun dari sisi pengolahan, penggunaan kamera lebih membutuhkan sistem pemroses yang cukup tinggi sehingga secara ekonomi lebih mahal.
- Microcontroller* yang digunakan pada satu persimpangan terdapat dua bagian yaitu *Master* dan *Slave*, untuk lebih ekonomisnya agar dapat dibuat mejadi satu *microcontroller* yang dapat mengatur semua bagian yang ada pada satu titik persimpangan.

- c. Untuk pengembangan sistem lampu lalu lintas adaptif berikutnya diharapkan dapat memasukan variabel kendaraan yang merupakan iring-iringan pejabat atau konvoi khusus dan kendaraan darurat seperti ambulans serta variabel pengalihan jalur secara otomatis bagi jalur yang sedang dalam kondisi terisolasi.

### PERNYATAAN ORIGINALITAS

“Saya menyatakan dan bertanggung jawab dengan sebenarnya bahwa artikel ini adalah hasil karya saya sendiri kecuali cuplikan dan ringkasan yang masing-masing telah saya jelaskan sumbernya” [Adrin. T P31.191100961].

### DAFTAR PUSTAKA

- [1] N. P. Simanjuntak, "Aplikasi *Fuzzy Logic controller* pada Pengontrolan Lampu Lalu Lintas," *Makalah IF4058 Topik Khusus Informatika I*, 2012.
- [2] Taufik, R, Supriyono, and Sukarman, "Rancang Bangun Simulator Kendali Lampu Lalu Lintas Dengan Logika *Fuzzy* Berbasis *Microcontroller*," *SEMINAR NASIONAL IV SDM TEKNOLOGI NUKLIR*, 2008, pp. 25-26.
- [3] Putra, B. S, "Simulasi Penerapan ANFIS Pada Sistem Lampu Lalu Lintas Enam Ruas," *Kursor*, vol. 6, 2011, pp. 77-82.
- [4] Suhandi, *dkk* "Simulasi antrian kendaraan pada persimpangan jalan berbasis *Multi Agent* menggunakan logika *fuzzy*," *Jurnal Teknologi Informasi*, vol. 7, 2011, pp. 1.
- [5] Tomescu, O. E "Adaptive Traffic Light Control System Using Adhoc Vehicular Communication Network," *U. P. B. Sci. Bull, Series D*, vol. 74, 2012, pp. 67-78.
- [6] Royani, T. E. "Control of Traffic Light in Isolated Intersections Using *Fuzzy* Neural Network and Genetic Algorithm," *International Journal of Computer and Electrical Engineering*, vol. 5, 2013, pp. 142-146.
- [7] Universitas Widyagama Malang, "*Lampu Lalu Lintas*", Diktat Kuliah :Rekayasa Lalu Lintas, Malang, 2008.
- [8] Harianto. J. Ir, "*Perencanaan Persimpangan Tidak Sebidang Pada Jalan Raya*", Digitized by USU digital library, 2004.
- [9] Direktorat Pembinaan Jalan Kota, "Tata Cara Perencanaan Persimpangan Sebidang Jalan Perkotaan," Direktorat Jendral Bina Marga, Jakarta, 1993.
- [10] Sutojo, T, *dkk*, "*Kecerdasan Buatan*", Edisi 1, Yogyakarta, Penerbit Andi, 2010, P 211-282.
- [11] Atmel Corp, "*8-bit Atmel Microcontroller Programmable ATmega128L*" [www. datasheet. com](http://www.datasheet.com), [www. atmel. com](http://www.atmel.com), 2013.
- [12] Atmel Corp, "*Programmable ATmega32 ( L )*" [www. datasheet. com](http://www.datasheet.com), [www. atmel. com](http://www.atmel.com), 2012.
- [13] Azhari, *dkk* "Sistem Agen Cerdas Berbasis Web Untuk Komputasi Informasi Kinerja dari Pelaksanaan Proyek-Proyek" *Juti*, vol. 8, 2009, pp. 39-46.
- [14] Hirankitti, V et. al, "A Multi-Agent Approach for Intelligent Traffic-Light Control," *Proceedings of The Congress on Engineering*, London: 2007, pp. 1-6
- [15] Wahono, R. S. "*Multi Agent System : Beberapa Isu , Pendekatan dan Tantangan*," *IECI Japan Refreshing Seminar 2001 (IJS-2001)*, vol. 3, 2001, pp. 22-37.
- [16] Sycara, K. P. "*Multi Agent Systems*", *American Association for Artificial Intelligence*, 1998, pp. 79-93.
- [17] Widodo P. P dan Handayanto, R. T. "*Penerapan Soft Computing dengan Matlab*", Bandung: Rekayasa Sains, 2009.
- [18] Kusumadewi, S dan Hartati, Sri, "*Neuro Fuzzy : Integrasi Sistem Fuzzy dengan Jaringan Syaraf Tiruan*", Edisi 2, Yogyakarta, Graha Ilmu, 2010.