

## **RANCANG BANGUN SIMULATOR KENDALI LAMPU LALU LINTAS DENGAN LOGIKA FUZZY BERBASIS MIKROKONTROLER**

**RAHMAT TAUFIK, SUPRIYONO, SUKARMAN**

*Sekolah Tinggi Teknologi Nuklir – BATAN*

*Jl. Babarsari Kotak Pos 6101 YKBB Yogyakarta 55281*

*Telp : (0274) 48085, 489716 ; Fax : (0274) 489715*

*E-mail : [sttn@batan.go.id](mailto:sttn@batan.go.id)*

### ***Abstrak***

**RANCANG BANGUN SIMULATOR KENDALI LAMPU LALU LINTAS DENGAN LOGIKA FUZZY BERBASIS MIKROKONTROLER.** Telah dibangun suatu simulator kendali lampu lalu lintas untuk mengatur lamanya waktu lampu hijau menyala pada suatu jalur tergantung dari kepadatannya. Metode yang digunakan untuk mengatur lamanya waktu ini adalah logika fuzzy dengan penalaran fuzzy metode Sugeno menggunakan perangkat lunak (software) MatLab untuk menyelesaikan persoalan tersebut. Dalam rancang bangun ini variabel input adalah kepadatan pada jalur I, jalur II dan jalur III sedangkan variabel output adalah lamanya waktu lampu hijau menyala untuk satu jalur. Variabel input diasumsikan dengan fungsi keanggotaan tidak padat, padat dan sangat padat sedangkan variabel output diasumsikan dengan fungsi keanggotaan cepat, sedang dan lama. Ada 27 rule fuzzy yang digunakan. Output yang didapat dari logika fuzzy menggunakan software MatLab dijadikan sebagai data acuan dalam pembuatan program dengan bahasa assembly untuk menentukan lamanya waktu lampu hijau menyala. Kemudian program tersebut ditanamkan kedalam IC mikrokontroler AT89S52. Hasil rancang bangun simulator lampu lalu lintas menunjukkan, dapat digunakan logika fuzzy sebagai pengambil keputusan untuk menentukan lamanya waktu lampu hijau menyala tergantung dari kepadatan dan mikrokontroler sebagai kendali.

*Kata Kunci : Logika fuzzy, MatLab, Lamanya waktu lampu hijau menyala.*

### ***Abstract***

**DESIGN OF TRAFFIC LIGHT CONTROLLER SIMULATOR USING MICROCONTROLLER BASED FUZZY LOGIC.** It had been built a simulator of traffic light controller to arrange the duration of a green light aflame in a line depended from density. Method applied in arranging the duration was Sugeno method of fuzzy logic uses MatLab software to finalize the problem. In this engineering the input variable is density of line I, line II and line II while the output variable is duration of green light aflame. The input variable is assumed with membership function of Not Massive, Massive and Very Massive while the output variable is assumed with membership function of Fast, Medium and slow. There are 27 rules fuzzy used. Output obtained from fuzzy logic use MatLab software made as reference data in making program using assembly language to determine the duration of a green light aflame. Then the program planted into AT89S52 microcontroller IC. Result of traffic light simulator engineering indicates, can be used fuzzy logic as decision maker to determine the duration of a green light aflame depended from density and microcontroller as controller.

*Keywords : Fuzzy logic, MatLab, The duration of a green light aflame*

## PENDAHULUAN

Perkembangan teknologi di zaman sekarang ini berjalan dengan sangat cepat. Berbagai macam karya teknologi diciptakan untuk memudahkan manusia dalam menjalankan segala macam bentuk aktivitas sehari-hari. Di Indonesia, khususnya pengguna kendaraan bermotor semakin meningkat, akibatnya jumlah kendaraan naik tetapi jumlah jalan tetap sehingga menambah jumlah kepadatan lalu lintas yang mengakibatkan kemacetan. Kemacetan yang muncul tersebut dapat disebabkan dari beberapa faktor, salah satunya adalah faktor pengatur lampu lalu lintas.

Saat ini di Indonesia teknologi kendali lampu lalu lintas terus dikembangkan sedemikian rupa, sehingga peran lampu lalu lintas bukan hanya untuk menghindari kemacetan saja tetapi juga berperan meningkatkan keselamatan lalu lintas. Lampu lalu lintas yang saat ini diterapkan dianggap belum optimal mengatasi kemacetan lalu lintas. Berdasarkan alasan diatas, maka perlu dilakukan penelitian tentang rancang bangun simulator kendali lampu lalu lintas dengan menggunakan logika *fuzzy* berbasis mikrokontroler.

Menangkap kemampuan dan kecerdasan manusia sering kali merupakan cara yang sangat bermanfaat untuk membuat program komputer yang dapat menangani permasalahan nyata yang seringkali rumit. Sedangkan manusia seringkali tidak bertindak berdasarkan aturan oleh karena itu logika *fuzzy* digunakan untuk mendekati kemampuan dan kecerdasan manusia yang selalu berubah – ubah atau tidak tetap dalam menyelesaikan suatu permasalahan. (Kuswadi, Son, 2000).

Rancang bangun yang dibuat pada penelitian ini baru pada tahapan prototype, belum menggunakan sensor yang dipasang di perempatan jalan. Sehingga input jumlah kendaraan yang lewat dimisalkan berupa masukan yang ada dalam fasilitas prototype. Dalam penelitian ini, hasil perhitungan dengan menggunakan logika *fuzzy* digunakan untuk mengatur input output dalam mikrokontroler. Untuk melakukan perhitungan dengan logika *fuzzy*nya digunakan perangkat lunak MatLab versi 7.01. Untuk pengisian program yang

ditanam dalam IC mikrokontroler AT89S52 digunakan bahasa assembly.

Hasil penelitian dapat digunakan sebagai prototype awal pengaturan lampu lalu lintas dengan perhitungan menggunakan logika *fuzzy*, sehingga dalam pengembangannya kelak pemrograman logika *fuzzy*nyapun seraca adaptif dapat langsung ditanam di dalam IC mikrokontroler AT89S52. Pengembangan lebih lanjut adalah penambahan sensor perhitungan jumlah kendaraan yang lewat, sehingga menjadi suatu alat yang terpadu dan dapat langsung di pasang di perempatan jalan sebagai alat kendali lampu pengaturan lalu lintas.

## DASAR TEORI

*Software* MatLab singkatan dari *Matrix Laboratory*, merupakan bahasa pemrograman yang dikembangkan oleh *The Mathwork Inc.* Bahasa pemrograman ini banyak digunakan untuk perhitungan numerik keteknikan, komputasi simbolik, visualisasi, grafis, analisis data matematis, statistika, simulasi, pemodelan dan desain GUI (Thomas & Y. Wahyu Agung Prasetyo, 2004). Ada beberapa alasan mengapa orang menggunakan logika *fuzzy* (Kusumadewi, Sri, 2002), yaitu :

1. Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti ;
2. Logika *fuzzy* sangat flexibel ;
3. Logika *fuzzy* memiliki toleransi terhadap data – data yang tidak tepat ;
4. Logika *fuzzy* mampu memodelkan fungsi – fungsi nonlinear yang sangat kompleks ;
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman – pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan ;
6. Logika *fuzzy* dapat bekerjasama dengan teknik – teknik kendali secara konvensional ;
7. Logika *fuzzy* didasarkan pada bahasa alami.

*Fuzzy* tipe Sugeno atau disebut juga Takagi-Sugeno-Kang (MATLAB, Sugeno-Type Fuzzy Inference :: Tutorial (Fuzzy Logic Toolbox). Diperkenalkan pada tahun 1985, metode Sugeno hampir sama dengan metode Mamdani dalam banyak hal. Pada metode Sugeno dua bagian pertama dari proses penarikan kesimpulan *fuzzy*, fuzzifikasi *input* dan menerapkan operator *fuzzy* semua sama

dengan metode Mamdani. Perbedaan utama antara metode Mamdani dan Sugeno adalah *output membership function* dari metode Sugeno berbentuk linier atau konstan.

Aturan pada model *fuzzy* Sugeno mempunyai bentuk :

If Input 1 = x and Input 2 = y

then Output is  $z = ax + by + c$

Untuk model Sugeno orde-Nol, *Output level* z adalah konstan ( $a=b=0$ ).

*Output level*  $z_i$  dari setiap aturan merupakan berat dari aturan  $w_i$  (*firing strength*). Sebagai contoh, untuk aturan AND dengan *Input 1* = x dan *Input 2* = y, maka *firing strength* adalah :  $w_i = \text{AndMethod}(F1(X), F2(Y))$

dimana  $F_{1,2}(\cdot)$  adalah *membership function* untuk *Input 1* dan 2.

Keuntungan metode Sugeno :

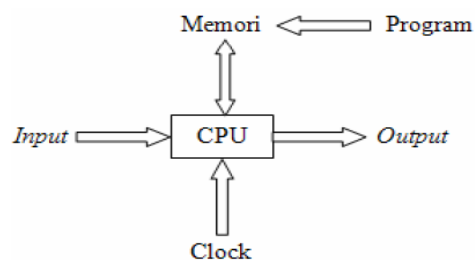
1. Komputasinya lebih efisien ;
2. Bekerja paling baik untuk teknik linear (kontrol PID, dll) ;
3. Bekerja paling baik untuk teknik optimasi dan adaptif ;
4. Menjamin kontinuitas permukaan *output* ;
5. Lebih cocok untuk analisis secara matematis.

Secara umum di dalam logika *fuzzy* ada 5 langkah dalam melakukan penalaran, yaitu :

1. Memasukkan *input fuzzy* ;
2. Mengaplikasikan operator *fuzzy* ;
3. Mengaplikasikan metode implikasi ;
4. Komposisi semua *output* ;
5. Defuzifikasi.

Mikrokontroler, sebagai suatu terobosan teknologi mikroprosesor dan mikrokomputer, hadir memenuhi kebutuhan pasar (*market need*) dan teknologi baru. Sebagai teknologi baru, yaitu teknologi semikonduktor dengan kandungan transistor yang lebih banyak namun hanya membutuhkan ruang yang kecil serta dapat diproduksi secara massal (dalam jumlah banyak) membuat harganya menjadi lebih murah (dibandingkan mikroprosesor). Sebagai kebutuhan pasar, mikrokontroler hadir untuk memenuhi selera industri dan para konsumen akan kebutuhan dan keinginan alat – alat bantu bahkan mainan yang lebih baik dan canggih.

Tidak seperti sistem komputer, yang mampu menangani berbagai macam program aplikasi (misalnya pengolahan kata, pengolahan angka dan lain sebagainya), mikrokontroler hanya bisa digunakan untuk suatu aplikasi tertentu saja (hanya satu program saja yang bisa disimpan), komponen sistem komputer ditunjukkan pada Gambar 1. Perbedaan lainnya terletak pada perbandingan RAM dan ROM. Pada sistem komputer perbandingan RAM dan ROM-nya besar, sedangkan rutin-rutin antarmuka perangkat keras disimpan dalam ruang ROM yang kecil. Sedangkan pada mikrokontroler, perbandingan ROM dan RAM-nya yang besar, artinya program kontrol disimpan dalam ROM (bisa *Masked ROM* atau *Flash ROM*) yang ukurannya relatif lebih besar, sedangkan RAM digunakan sebagai tempat penyimpanan sementara, termasuk register – register yang digunakan pada mikrokontroler yang bersangkutan.



Gambar 1. Komponen Sistem Komputer

Walaupun bahasa pemrograman tingkat tinggi terus berkembang dengan segala fasilitas dan kemudahannya, peranan bahasa pemrograman tingkat rendah tetap tidak dapat digantikan. Bahasa assembly mempunyai keunggulan yang tidak mungkin diikuti oleh bahasa tingkat apapun dalam hal kecepatan, ukuran file yang kecil serta kemudahan dalam manipulasi sistem komputer. (S'to, 2001)

## METODE PENELITIAN

Langkah kerja yang dilakukan untuk merancang bangun penelitian ini terdiri dari :

### Proses Pengaturan Lampu Lalu Lintas

Pada umumnya arah perpindahan lampu lalu lintas dapat diatur sesuai dengan arah jarum jam (*clockwise*) atau berlawanan arah jarum jam (*counter clockwise*). Proses pengaturan perpindahan lampu lalu lintas pada penelitian ini mengikuti arah seperti ditunjukkan pada

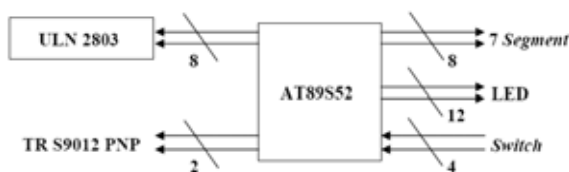
Gambar 2. Lampu lalu lintas bekerja secara bergantian pada tiap jalur sesuai dengan waktu yang sudah ditentukan dengan urutan menyala lampu hijau, lampu kuning dan lampu merah.



Gambar 2. Proses Pengaturan Lampu Lalu Lintas

### Rancang Bangun Hardware

Langkah yang dilakukan pada rancang bangun hardware penelitian ini adalah membuat rangkaian sesuai dengan Gambar 3.



Gambar 3. Skematik Kendali Lampu Lalu Lintas

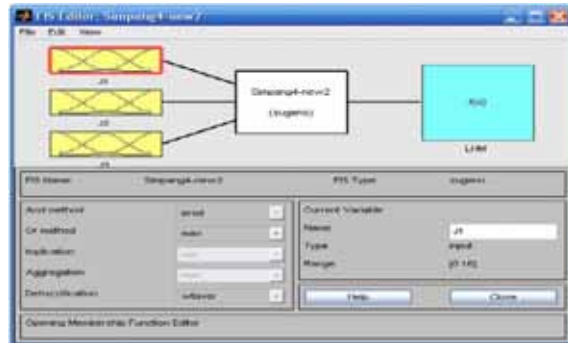
### Rancangan Software MATLAB

Pembuatan program pada penelitian ini berdasarkan data output yang diperoleh dari simulasi lampu lalu lintas simpang empat menggunakan logika fuzzy Software MatLab. Langkah awal yang dilakukan untuk pembuatan simulasi tersebut adalah menjalankan toolbox fuzzy software MatLab kemudian menentukan

nilai untuk setiap parameter yang disediakan sebagai berikut :

1. Membership Function Editor ;
2. FIS Editor ;

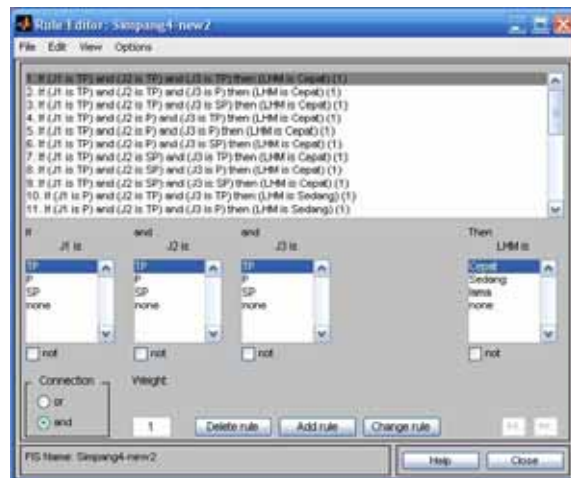
Bentuk FIS Editor yang dihasilkan dengan program Matlab Versi 7.01 ditampilkan pada gambar 4 berikut ini :



Gambar 4. FIS Editor

### Rule Editor

Adapun aturan fuzzy yang digunakan ditampilkan pada gambar 5 berikut :



Gambar 5. Aturan Fuzzy

1. Rule Viewer ;
2. Surface Viewer ;
3. Output Fuzzy Menggunakan Software MatLab.

Simulasi lampu lalu lintas simpang empat menggunakan fuzzy software MatLab diperoleh data yang ditunjukkan pada Tabel 1.

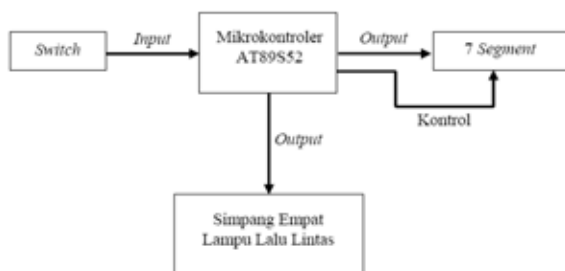
Tabel 1. *Output Fuzzy Software*

Input 1/Jumlah Kepadatan Ji	Input 2/Jumlah Kepadatan JII	Input 3/Jumlah Kepadatan JIII	oUTPUT//Ilhmi (DETIK)
0-3	x	x	3
4	x	x	5
5	x	x	7
6	x	x	9
7	x	x	11
8	x	x	13
9	x	x	15
10	x	x	17
11	x	x	19
12	x	x	21
13-16	x	x	23

Pada Tabel 1. dapat dilihat *input* 1 data 0-3 dan 13-16 mempunyai *output* yang nilainya masing-masing 3 dan 23, hal ini dikarenakan dalam menentukan parameter pada *membership function editor* bagian *ouput* oleh peneliti diberikan nilai untuk fungsi keanggotaan cepat adalah 3 dan fungsi keanggotaan lama adalah 23. Nilai tersebut tidaklah mutlak, nilai tersebut disesuaikan dengan fungsinya pada penelitian ini sebagai simulator. Nilai *input* maupun nilai *output* dapat dirubah sesuai dengan kebutuhan.

### Rancangan Program Mikrokontroler

Setelah algoritma, rancang bangun *hardware* dan rancangan *software* MatLab selesai dibangun, langkah berikut yang dilakukan adalah membuat program dengan bahasa assembly dengan menggunakan Tabel 1. sebagai acuan. Hal penting yang tidak boleh dilupakan yaitu program haruslah dibuat sesuai dengan rancang bangun *hardware* karena berhubungan dengan cara kerja dari rangkaian, seperti yang ditunjukkan Gambar 6.



Gambar 6. Blok Diagram Kerja Rangkaian

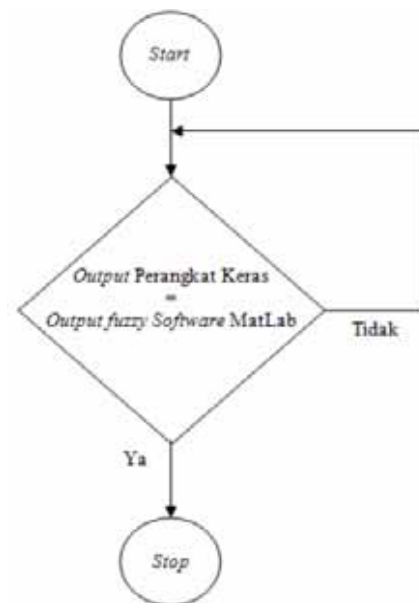
Karena program dengan bahasa assembler sangat panjang, maka dalam makalah ini, program assembly tidak ditampilkan.

### Pengujian

Langkah yang dilakukan untuk menguji penelitian ini dapat dikategorikan menjadi beberapa bagian, antara lain :

1. Menguji rangkaian *driver 7 segment* ;
2. Menguji rangkaian IC ULN 2803 ;
3. Menguji rangkaian *switch* ;
4. Menguji rangkaian LED ;
5. Menguji rangkaian IC mikrokontroler AT89S52 ;
6. Menguji keluaran LED yang menyala ;
7. Menguji keluaran lamanya waktu lampu hijau menyala.

Setelah perangkat keras dapat dibuktikan bahwa *output* yang dihasilkan sesuai dengan *output fuzzy software* MatLab seperti yang ditunjukkan Gambar 6. selanjutnya dilakukan pengambilan data.



Gambar 7. Algoritma Pengujian

### HASIL DAN PEMBAHASAN

Pembuatan rancang bangun simulator kendali lampu lalu lintas dengan logika *fuzzy* berbasis mikrokontroler dilakukan melalui beberapa tahap. Pada penelitian ini proses pergantian lampu lalu lintas dimulai dengan kondisi awal lampu hijau menyala pada jalur utara dan lampu merah pada jalur barat, jalur selatan dan jalur timur. Lampu merah ditunjukkan dengan LED warna merah, lampu kuning ditunjukkan dengan LED warna kuning dan lampu hijau ditunjukkan dengan LED warna hijau. Untuk dapat menjalankan

simulator haruslah diberi *input* tegangan 5 Volt pada rangkaian kendali. Pergantian lampu lalu lintas pada penelitian ini berjalan secara terus menerus selama 24 jam. Lampu lalu lintas pada penelitian ini bekerja layaknya lampu lalu lintas dengan keadaan yang sudah diterapkan saat ini.

Pada penelitian ini rancang bangun *hardware* dibangun sesuai dengan tujuan agar berfungsi sebagai simulator lampu lalu lintas. *Switch* berfungsi sebagai penghitung kepadatan pada saat menyala lampu merah pada jalur dan merupakan *input* pada mikrokontroler. Data kepadatan tersebut menentukan lamanya waktu lampu hijau menyala pada jalur yang ditunjukkan oleh 7 *segment* sebagai pewaktu. *Switch* dihubungkan dengan Vcc sehingga apabila *switch* ditekan akan memberikan logika 1 pada mikrokontroler. 7 *segment* dapat menunjukkan angka maksimum 25 dan angka minimum 5 yang bekerja menghitung mundur (*countdown*).

Peraga 7 *segment* yang digunakan pada penelitian ini adalah tipe anoda yaitu pada kaki 3 dan kaki 8 dihubungkan dengan Vcc (5Volt), dapat juga digunakan 7 *segment* tipe katoda dengan syarat pada kaki 3 dan kaki 8 dihubungkan dengan *ground* (GND). Masukan 7 *segment* berasal dari IC ULN 2803 yang merupakan gerbang logika NOT berfungsi merubah logika 1 menjadi logika 0 begitu pula sebaliknya. Pada penelitian ini digunakan IC ULN 2803 dikarenakan 7 *segment* tipe anoda merupakan aktif low sehingga untuk menghidupkan LED pada 7 *segment* tipe anoda harus diberi *input* logika 0.

Pada penelitian ini rancangan *software* MatLab digunakan untuk mensimulasikan simpang empat lampu lalu lintas yang mempunyai *switch* pada jalur yang berfungsi sebagai penghitung kepadatan sehingga lamanya waktu lampu hijau menyala pada jalur tergantung dari kepadatan jalur. Pada *membership function editor* bagian *input* fungsi keanggotaan tidak padat (TP) dan sangat padat (SP) menggunakan kurva trapesium (*trapmf*) dan fungsi keanggotaan padat (P) menggunakan kurva segitiga (*trimf*). Pada *membership function editor* bagian *output* menggunakan tipe *constant* dengan tujuan agar data *output* yang dihasilkan sudah merupakan bilangan bulat.

*Rule* yang digunakan sebanyak 27 buah, banyaknya *rule* ini adalah jumlah maksimum. Dengan menerapkan *rule* maksimum

diharapkan agar didapat nilai *output fuzzy* dengan kondisi semakin besar jumlah kepadatan pada jalur maka lamanya waktu lampu hijau menyala pada jalur semakin lama dan semakin kecil jumlah kepadatan pada jalur maka lamanya waktu lampu hijau menyala pada jalur semakin cepat.

Pada penelitian ini langkah kerja program yang dibuat dapat disimpulkan sebagai berikut :

1. Kondisi awal yaitu menghidupkan lampu hijau pada jalur utara dan lampu merah pada jalur yang lain dengan 7 *segment* menunjukkan angka 5 yaitu lampu hijau menyala selama 3 detik dan lampu kuning menyala selama 2 detik kemudian lampu hijau menyala pada jalur barat dan lampu merah pada jalur yang lain dengan lamanya waktu lampu hijau menyala pada jalur barat tergantung dari jumlah kepadatan dengan maksimal jumlah kepadatan adalah 5. Hal ini disebabkan program dibuat untuk setiap detik hanya dapat di isi oleh 1 kepadatan ;
2. *Switch* akan memberikan logika 1 pada saat ditekan dan menyimpannya di register masing-masing jalur yang telah ditentukan peneliti, dengan demikian pada saat jalur akan mendapat giliran lampu hijau menyala mikrokontroler sudah dapat memproses berapa lamanya waktu lampu hijau menyala yang akan dikeluarkan pada jalur ;
3. Lamanya waktu lampu kuning menyala dibuat selama 2 detik sebelum lampu merah menyala dan lamanya waktu lampu merah menyala pada jalur tergantung dari jumlah lamanya waktu lampu hijau pada jalur lain.

Pengujian yang dilakukan untuk rancang bangun *hardware* dengan cara memastikan tiap jalur rangkaian terhubung dengan semestinya sesuai dengan *datasheet* menggunakan multimeter. Pengujian tidak hanya dilakukan dengan menggunakan ohmmeter untuk mengetahui jalur terhubung atau tidaknya tetapi juga dengan diberi tegangan pada tiap titik rangkaian untuk mengetahui dari setiap komponen bekerja dengan baik atau tidak.

Pengujian program yang ditanam ke dalam mikrokontroler dilakukan secara bertahap dari program yang sederhana sampai yang paling sulit dari penelitian ini. Program yang sederhana berupa memberikan logika 1 di setiap kaki mikrokontroler AT89S52 untuk membuktikan bahwa mikrokontroler bekerja

dengan baik kemudian menghidupkan LED di tiap jalur secara bergantian untuk membuktikan bahwa LED bekerja dan mengaktifkan *switch*. Untuk menghidupkan 2 buah 7 *segment* yang menunjukkan angka waktu tundaan akan menghitung mundur (*countdown*) dari angka maksimum 25 sampai 0 berdasarkan rancangan nilai tundaan tersebut sudah sesuai dengan hasil yang ditunjukkan pada peraga.

Setelah semua langkah pengujian dilakukan dan rancang bangun *hardware* sudah bekerja dengan baik sesuai dengan program yang dibuat kemudian diperoleh data yang ditunjukkan pada Tabel 2.

Tabel 2. *Output* Rancang Bangun Simulator

Jumlah Kepadatan Jalur I	Jumlah Kepadatan Jalur II	Jumlah Kepadatan Jalur III	Lamanya Waktu Lampu Hijau Menyala Jalur I
0	X	X	3
1	X	X	3
2	X	X	3
3	X	X	3
4	X	X	5
5	X	X	7
6	X	X	9
7	X	X	11
8	X	X	13
9	X	X	15
10	X	X	17
11	X	X	19
12	X	X	21
13	X	X	23
14	X	X	23
15	X	X	23
16	X	X	23

Seperti yang terlihat pada Tabel 2. Jalur I dianggap sebagai jalur yang akan mendapat giliran lampu hijau menyala sedangkan jalur II merupakan jalur yang akan mendapat giliran lampu hijau menyala setelah jalur I dan jalur III merupakan jalur yang akan mendapat giliran lampu hijau menyala setelah Jalur II. Nilai X adalah jumlah kepadatan pada jalur dengan nilai sembarang yaitu antara 0 dan 16. Jalur yang akan mendapat giliran lampu hijau menyala dikatakan sebagai Jalur I. Lamanya waktu lampu hijau menyala yang ditunjukkan Tabel 2. Sesuai dengan lamanya waktu lampu hijau menyala yang ditunjukkan Tabel 1.

## KESIMPULAN

1. Simulasi simpang empat lampu lalu lintas yang dilakukan menggunakan logika *fuzzy software* MatLab metode Sugeno diperoleh nilai lamanya waktu lampu hijau menyala tergantung dari jumlah kepadatan ;
2. Lamanya waktu lampu hijau menyala di suatu jalur tergantung dari jumlah kepadatan di jalur tersebut. Semakin besar jumlah kepadatan di suatu jalur maka semakin lama lampu hijau di jalur tersebut begitu juga sebaliknya ;
3. Besar ataupun kecilnya jumlah kepadatan di suatu jalur ditentukan oleh banyak ataupun sedikitnya *switch* pada jalur ditekan.

## DAFTAR PUSTAKA

1. EKO PUTRA, AGFIANTO, 2004, *Belajar Mikrokontroler AT89C51/52/55 (Teori dan Aplikasi)*. Yogyakarta, Gava Media.
2. MATLAB, "Sugeno-Type Fuzzy Inference :: Tutorial (Fuzzy Logic Toolbox)"
3. KUSUMADEWI, SRI, 2002, *Analisis & Desain Sistem Fuzzy Menggunakan Toolbox MATLAB*. Yogyakarta, GRAHA ILMU.
4. KUSWADI, SON, 2000, *KENDALI CERDAS (INTELLIGENT CONTROL)*, Surabaya, EEPIS Press.
5. S'TO, 2001. "Pemrograman Dengan Bahasa Assembly Edisi Online Versi 1.0", Jasakom
6. WAHYU DWI HARTANTO, THOMAS & Y. WAHYU AGUNG PRASETYO, 2004. *Analisis dan Desain Sistem Kontrol dengan MATLAB*, Yogyakarta, ANDI offset.
7. <http://alfi.blogs.ie/2007/01/09/sistem-pengaturan-lampu-lalu-lintas-logika-fuzzy/> (01 Maret 2008)

## TANYA JAWAB

### Pertanyaan

1. Apakah rangkaian hardware telah menggunakan ON/OFF relay logic ke sistem 220 VAC dan berapa jumlah saklar/switch sebagai sensor tiap jalur dalam simulasi? Subari Santoso-STTN
2. Berapa jumlah saklar yang paling efektif untuk 1 jalur harus terpasang sebagai sensor

pada variabelnya ? (untuk realita). Subari Santoso-STTN

3. Sensor apa yang digunakan untuk mengetahui kepadatan lalu lintas? Wayan W (PPR - BATAN)
4. Mungkinkah pada kenyataannya setelah dipasang sensor tersebut akan terjadi “kepadatan semu” akibat salah deteksi. Wayan W (PPR - BATAN)

#### **Jawaban**

1. Rangkaian ini tidak menggunakan ON/OFF relay logic dan jumlah saklar sebagai sensor tiap jalurnya ada 4 saklar/switch
2. Yang paling efektif untuk 1 jalur cukup 1 saklar/switch.
3. Kelak sensor yang digunakan adalah sensor cahaya dengan asumsi jika ada kendaraan ada di daerah deteksi, maka akan muncul luasan dan luasan tersebut dibagi dengan satuan kendaraan, sehingga akan dihasilkan jumlah kendaraan yang sedang antri di traffic light.
4. Jika distribusi kepadatan lalu lintas normal, kemungkinan muncul “kepadatan semu” kecil, tetapi jika terjadi kondisi lalu lintas yang mendadak, misalkan ada iring-iringan rombongan kendaraan yang lewat, maka dimungkinkan “kepadatan semu” akan muncul.