

# Perancangan Simulasi Otomatis *Traffic light* Menggunakan Citra Digital Studi Kasus Persimpangan Toar-Lumimuut

Riansa E. P. Tolah, Rizal Sengkey, ST., MT, Yaulie D. Y. Rindengan, ST., MM., MSc.  
Jurusan Teknik Elektro-FT, UNSRAT, Manado-95115, Email: [funnyghozi@yahoo.com](mailto:funnyghozi@yahoo.com)

**Abstrak** - Lampu lalu lintas (*traffic light*) seiring dengan peningkatan jumlah kendaraan akan sangat di butuhkan dalam pengaturan lalu lintas terutama di kota-kota besar yang pertumbuhan jumlah kendaraannya meningkat cepat. Kemacetan terjadi karena pengaturan pewaktuan lampu lalu lintas yang terpasang masih menggunakan pewaktuan pada kondisi lalu lintas normal. Dengan demikian diperoleh ide untuk membuat simulasi *traffic light* secara otomatis dengan *system* adaptif yaitu *traffic light* menyesuaikan lamanya waktu lampu hijau berdasarkan kepadatan pada ruas jalan.

Perancangan aplikasi menggunakan *Software MATLAB 2009a* dan *Camera Digital* sebagai pengolahan dan *input* citra *traffic light*. Pengambilan *sample* gambar persimpangan *traffic light* masih berupa *prototype* (miniature Persimpangan Toar-Lumimuut) untuk pengolahan citra digital.

Dengan menggunakan alat dan bahan di atas, sistem ini dapat menentukan lama waktu nyala lampu hijau berdasarkan kepadatan pada ruas jalan. Program ini hanya bisa dijalankan pada lokasi persimpangan yang sesuai dengan referensi (Persimpangan Toar-Lumimuut).

**Kata Kunci** : Citra Digital, *MATLAB 2009a*, Persimpangan Toar-Lumimuut, *traffic light*

**Abstract** - *The traffic light along with the increasing number of vehicles will be needed in the regulation of traffic, especially in big cities are growing rapidly increasing number of vehicles. Congestion occurs because setting the timing of traffic lights installed still using the timing in normal traffic conditions. Thus obtained the idea to create a simulation of traffic light automatically with the traffic light system that is adaptive to adjust the length of time the green light by the density of the road.*

*Software application design using MATLAB 2009a and Digital Camera as an input image processing and traffic light. The sample images traffic light intersection is still a prototype (miniature Intersection Toar-Lumimuut) for digital image processing.*

*By using the tools and materials above, the system can determine the length of time green light based on the density of roads. This program can only be run on the location that corresponds to the reference intersection (Intersection Toar-Lumimuut).*

**Keywords:** *Digital Image Processing, MATLAB 2009a, Intersection Toar-Lumimuut, traffic light*

## I. PENDAHULUAN

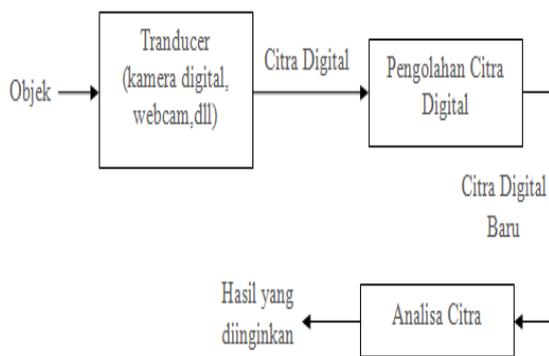
Di era modern ini banyak kita hadapi berbagai piranti otomatis yang bekerja secara mandiri tanpa banyak campur tangan dari manusia, piranti – piranti ini bekerja secara otomatis dengan cara mengambil kesimpulan dari berbagai data yang diperoleh dan kemudian melaksanakan keputusan tersebut sesuai dengan keadaan yang telah ditentukan. Data – data tersebut dapat berupa data yang diambil dari sensor ataupun data yang diberikan oleh pengguna piranti tersebut.

Lampu lalu lintas (*traffic light*) seiring dengan peningkatan jumlah kendaraan akan sangat di butuhkan dalam pengaturan lalu lintas terutama di kota-kota besar yang pertumbuhan jumlah kendaraannya meningkat cepat. Kemacetan lalu lintas merupakan salah satu masalah besar yang terjadi di kota besar dan umumnya terjadi di persimpangan jalan. Lampu lalu lintas di satu kondisi akan sangat membantu dalam kelancaran arus lalu lintas tetapi di lain kondisi malah menjadikan kemacetan semakin parah. Contoh banyak yang terjadi adalah saat jam sibuk dan sore hari. Kemacetan terjadi karena pengaturan pewaktuan lampu lalu lintas yang terpasang masih menggunakan pewaktuan pada kondisi lalu lintas normal. Dengan pewaktuan tersebut akan menyebabkan terjadinya penumpukan jumlah kendaraan di salah satu sisi persimpangan dan sangat rentan menyebabkan terjadinya kemacetan.

## II. LANDASAN TEORI

### A. Pengolahan Citra

Pada dasarnya sistem pengolahan citra terdiri dari objek yang akan diolah, *transducer* yang digunakan untuk merepresentasikan objek dalam bentuk citra digital, pengolahan citra digital untuk menghasilkan suatu citra digital baru untuk dianalisa, serta proses



Gambar 1. Sistem pengolahan citra

analisa citra untuk mendapatkan hasil yang diinginkan. Secara garis besar sistem di atas dapat dilihat pada gambar 1.

**B. Langkah Langkah dalam Pengolahan Citra**

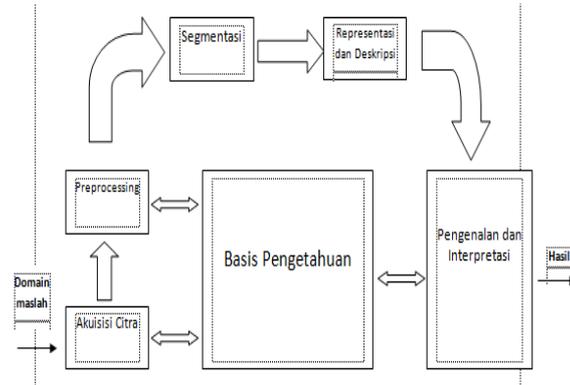
Tahap-tahap pengolahan citra dapat di lihat pada gambar 2. Akuisisi citra, merupakan tahap awal untuk mendapatkan citra digital. Tujuan akuisisi citra adalah untuk menentukan data yang diperlukan dan memilih metode perekaman citra digital. Tahap ini dimulai dari objek yang akan di ambil gambarnya, persiapan alat-alat sampai pada pencitraan. Hasil dari akuisisi citra ini ditentukan oleh kemampuan sensor untuk mendigitalisasi sinyal yang terkumpul pada sensor tersebut, kemampuan digitalisasi alat ditentukan oleh resolusi alat tersebut.

*Pre-processing*, menjamin kelancaran proses berikutnya misalnya dengan menentukan bagian citra yang akan di observasi.

Segmentasi, tahap ini bertujuan mempartisi citra menjadi bagian-bagian pokok yang mengandung informasi penting. Misalnya memisahkan objek dari latar belakang. Segmentasi adalah salah satu metode penting untuk mengubah citra *input* kedalam citra *output* berdasarkan atribut yang diambil.

Representasi dan deskripsi, bagian ini untuk mempresentasikan suatu wilayah sebagai suatu daftar titik-titik koordinat. Selanjutnya melakukan deskripsi dengan cara seleksi ciri yakni untuk memilih informasi kuantitatif dari ciri yang ada untuk membedakan kelas-kelas objek secara baik dan ekstrasi ciri yakni untuk mengukur besaran kuantitatif ciri setiap *pixel* misalnya rata-rata, standar deviasi, koevisien variasi.

Pengenalan dan interpretasi, tahap ini bertujuan untuk memberi label pada objek yang informasinya disediakan oleh *descriptor*, kemudian interpretasi untuk memberi arti pada objek-objek yang dikenali.



Gambar 2. Tahap-tahap Pengolahan Citra

Basis Pengetahuan, Sebagai basis data pengetahuan berguna untuk memandu operasi dari masing-masing modul proses dan mengontrol interaksi antar modul-modul tersebut.

**C. Thresholding**

Karena properti intuitif dan kesederhanaannya dalam implementasi, thresholding citra menjadi titik pusat dalam aplikasi segmentasi citra. Metode ini menggunakan nilai ambang T sebagai patokan untuk memutuskan sebuah *pixel* di ubah menjadi hitam atau putih. Biasanya T dihitung dengan persamaan (1).

$$T = \frac{f_{max} + f_{min}}{2} \tag{1}$$

Dimana *f<sub>max</sub>* adalah nilai intensitas maksimum pada citra dan *f<sub>min</sub>* adalah nilai intensitas minimum pada citra. Jika *f(x,y)* adalah nilai intensitas *pixel* pada posisi (x,y) maka *pixel* tersebut diganti putih atau hitam tergantung kondisi berikut :

$$f(x,y) = 1, \text{ jika } f(x,y) > T$$

$$f(x,y) = 0, \text{ jika } f(x,y) < T$$

**D. Sifat Geometri**

Sifat geometri dari objek bisa dipakai sebagai ciri objek tersebut di dalam citra, yang meliputi ukuran, lokasi, bentuk, orientasi (sudut kemiringan objek terhadap garis acuan yang digunakan) dan ciri lain yang berguna untuk mengenali objek dalam citra yang dianalisis.

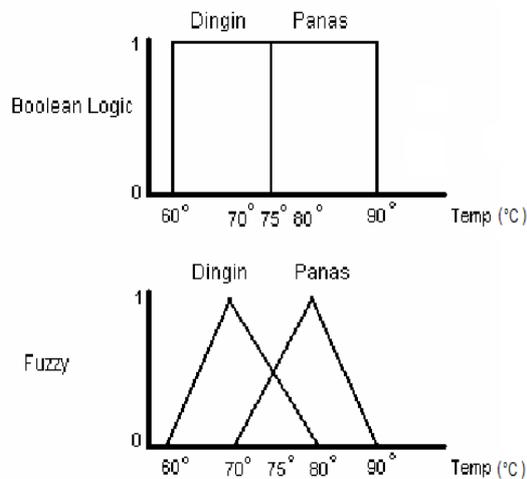
Pemotongan (*cropping*), Adalah pengolahan citra dengan kegiatan memotong satu bagian citra. Menggunakan rumus 2 & 3.

$$x' = x - xL \text{ untuk } x = xL \text{ sampai } xR \tag{2}$$

$$y' = y - yT \text{ untuk } y = yT \text{ sampai } yB \tag{3}$$

**E. Fuzzy Logic**

Sistem *Fuzzy* ditemukan pertama kali oleh Prof. Lotfi Zadeh pada pertengahan tahun 1960 di Universitas



Gambar 3. Perbedaan *Fuzzy Logic* dan *Boolean Logic*

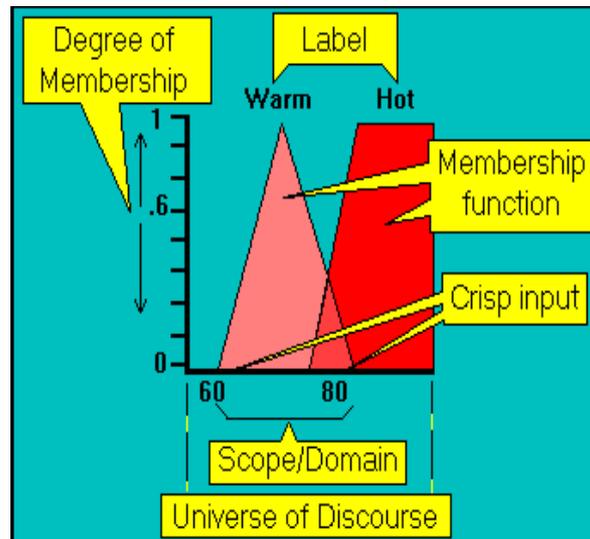
California. Sistem ini diciptakan karena *boolean logic* tidak mempunyai ketelitian yang tinggi, hanya mempunyai logika 0 dan 1 saja. Ia berpendapat bahwa logika benar dan salah dari logika Boolean tidak dapat mengatasi masalah gradasi yang berada pada dunia nyata. Sehingga untuk membuat sistem yang mempunyai ketelitian yang tinggi maka kita tidak dapat menggunakan *boolean logic*. Bedanya *Fuzzy* dengan *boolean logic* dapat diilustrasikan pada gambar 3.

Untuk mengatasi masalah gradasi yang tidak terhingga tersebut, Zadeh mengembangkan sebuah himpunan *Fuzzy*. Tidak seperti logika boolean, logika *Fuzzy* mempunyai nilai yang kontinu.

*Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh karena itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama. Terkadang kemiripan antara keanggotaan *Fuzzy* dengan probabilitas menimbulkan kerancuan. Keduanya memiliki nilai antara [0,1], namun interpretasi nilainya sangat berbeda. Keanggotaan *Fuzzy* memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang.

Ada beberapa alasan mengapa orang menggunakan logika *Fuzzy* antara lain lima hal yang dapat diuraikan.

Konsep logika *Fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *Fuzzy* sangat sederhana dan mudah dimengerti. Dapat mengekspresikan konsep yang sulit untuk dirumuskan, seperti misalnya "suhu ruangan yang nyaman". Pemakaian membership - *function* memungkinkan *Fuzzy* untuk melakukan



Gambar 4. Istilah yang digunakan dalam *Fuzzy logic*

observasi obyektif terhadap nilai-nilai yang subyektif. Selanjutnya *membership-function* ini dapat dikombinasikan untuk membuat pengungkapan konsep yang lebih jelas. Logika *Fuzzy* dapat membangun dan mengaplikasikan pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan. Logika *Fuzzy* sangat fleksibel.

Dari gambar 3, pada saat suhu berada pada 75°C maka sistem yang pertama akan bingung karena batas kondisi dingin < 75°C dan kondisi panas > 75°C, pada *Fuzzy logic*, suhu 75°C dapat dinyatakan dengan 0.50 dingin dan 0.50 panas. Pengambilan nilai 0.50 berasal dari proses *fuzzifikasi* yang akan diterangkan pada proses *fuzzification*.

Pada gambar 4 dapat dilihat enam istilah yang digunakan dalam *Fuzzy* dan keterangannya adalah sebagai berikut :

*Degree of membership* (derajat keanggotaan), Fungsi dari *degree of membership* ini adalah untuk memberikan bobot pada suatu *input* yang telah kita berikan, sehingga *input* tadi dapat dinyatakan dengan nilai. Misalnya suhu adalah dingin, dengan adanya *degree of membership* maka suhu dingin tersebut dapat mempunyai suatu nilai misal 0,5. Batas dari *degree of membership* adalah dari 0 – 1.

*Scope / Domain*, merupakan suatu batas dari kumpulan *input* tertentu. Misalnya suhu dingin adalah dari 10 – 50 derajat, sangat cepat adalah dari 200 – 500 rpm.

Label, adalah kata – kata untuk memberikan suatu keterangan pada *Scope / Domain*. Contohnya : panas, dingin, cepat, sangat cepat, dan lain-lain.

Membership *Function* (fungsi keanggotaan), suatu bentuk bangun yang merepresentasikan suatu batas dari *scope / domain*.

*Crisp Input*, nilai *input* analog yang kita berikan untuk mencari *degree of membership*.

*Universe of discourse*, batas *input* yang telah kita berikan dalam merancang suatu *Fuzzy system*. Batas ini berbeda dengan batas *scope / domain*. *Universe of discourse* adalah batas *input* yang akan diberikan sedangkan *scope / domain* adalah suatu batas yang menentukan bahwa *input* tersebut dinyatakan panas, dingin, cepat, dan lain-lain.

#### F. Metode Mamdani

Metode Mamdani sering juga dikenal dengan nama metode Max-Min. metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975 (Kusuma Dewi, 2003). Untuk mendapatkan *output* diperlukan beberapa tahapan.

Pembentukan himpunan *Fuzzy* pada Metode Mamdani, baik variable *input* maupun variable *output* dibagi menjadi satu atau lebih himpunan *Fuzzy*.

Aplikasi fungsi implikasi pada Metode Mamdani, fungsi implikasi yang digunakan adalah Min.

Komposisi aturan tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan maka inferensi diperoleh dari kumpulan dan kolerasi antar aturan.

#### G. Bwarea

*Bwarea* merupakan teknik pengolahan citra yang memperkirakan jumlah daerah dalam *pixel* yang terkandung pada sebuah gambar yang sudah *dibinerisasi*. Suatu daerah adalah pengukuran ukuran latar depan (*foreground*) dari citra atau dapat juga dikatakan sebagai jumlah *pixel* pada citra. Fungsi ini tidak secara sederhana hanya menghitung jumlah *pixel*, tetapi juga melakukan pembobotan pola *pixel* yang berbeda ketika menghitung suatu area.

Total = *bwarea* (BW) memperkirakan wilayah objek dalam citra *biner* BW. total adalah skalar yang nilainya sesuai dengan jumlah total pada *pixel* dalam gambar, tapi mungkin tidak persis sama karena pola yang berbeda dari bobot *pixel*.

BW dapat berupa numerik atau logis. Untuk *input* numerik, setiap *pixel* nol dianggap ada.

Algoritma *bwarea* memperkirakan luas dari semua *pixel* di dalam gambar dengan menjumlahkan bidang setiap *pixel* dalam gambar. Wilayah dimana *pixel* individu ditentukan dengan melihat 2-by-2 tetangganya. Ada enam pola yang berbeda, masing-masing mewakili daerah yang berbeda:

Pola dengan nol pada *pixel* (daerah = 0)

Pola dengan satu pada *pixel* (daerah = 1/4)

Pola dengan dua berdekatan pada *pixel* (daerah = 1/2)

Pola dengan dua diagonal pada *pixel* (daerah = 3/4)

Pola dengan tiga pada *pixel* (daerah = 7/8)

Pola dengan keempat pada *pixel* (daerah = 1)

*Bwarea* sebagai alat ukur membantu *visualisasi* hasil dengan menghasilkan gambar dimana setiap bentuk diberi label dengan nilai wilayah. *bwarea* tidak hanya menghitung jumlah *pixel* diaktifkan, namun. Sebaliknya, bobot *bwarea* pola *pixel* di hitung tidak merata. Bobot ini mengkompensasi distorsi yang melekat dalam merepresentasikan sebuah gambar terus menerus dengan *pixel* diskrit. Misalnya, garis diagonal dari 50 *pixel* lebih panjang dari garis horizontal 50 *pixel*. Sebagai hasil dari *bwarea* pembobotan menggunakan, garis horizontal memiliki luas wilayah 50, tetapi garis diagonal memiliki luas wilayah 62,5.

#### H. Kamera Digital

Kamera Digital adalah alat untuk membuat gambar dari objek untuk selanjutnya dibiarkan melalui lensa kepada sensor CCD (*charge coupled device*) yang hasilnya kemudian direkam dalam format digital ke dalam media penyimpanan. Karena hasilnya disimpan secara digital maka hasil rekam gambar ini harus diolah menggunakan pengelola digital pula semacam komputer atau mesin cetak yang dapat membaca media simpan digital tersebut.

Kemudahan dari kamera digital adalah hasil gambar yang dengan cepat diketahui hasilnya secara instan, kemudahan memindahkan hasil (*transfer*), dan penyuntingan warna, ketajaman, kecerahan dan ukuran yang dapat dilakukan dengan *relative* lebih mudah.

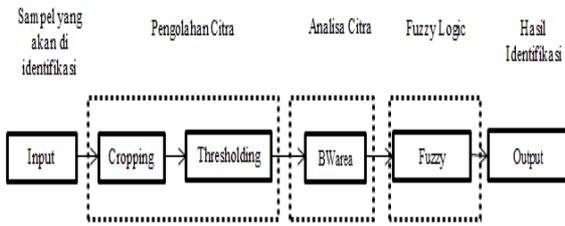
#### I. Matlab (Matrix Laboratory)

Matlab merupakan bahasa pemrograman dengan kemampuan tinggi dalam bidang komputasi. Saat ini, bahasa pemrograman tidak hanya dituntut memiliki kemampuan dari segi komputasi, tetapi juga kemampuan *visualisasi* yang baik. Matlab memiliki kemampuan untuk mengintegrasikan komputasi, *visualisasi*, dan pemrograman. Dalam *visualisasi* sebuah objek, matlab memiliki kemampuan merotasi obyek tanpa mengubah programnya.

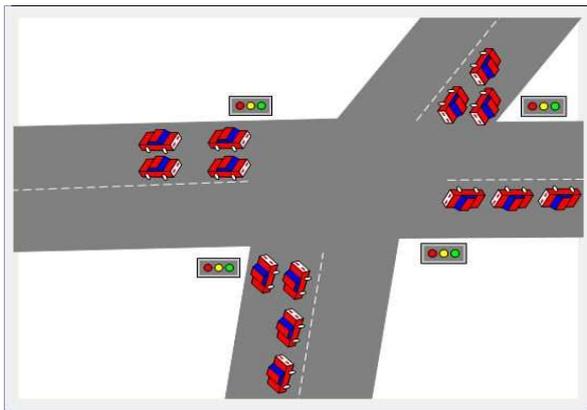
### III. PERANCANGAN SISTEM

#### A. Konsep Dasar Perancangan Otomatis Traffic light

Konsep dasar merupakan pedoman untuk merencanakan sesuatu dalam melakukan rancangan, dimana konsep ini memuat langkah – langkah dan



Gambar 5. Diagram blok program



Gambar 6. Skema Miniatur Persimpangan Toar-Lumimuut



Gambar 7. Miniatur Persimpangan Toar-Lumimuut

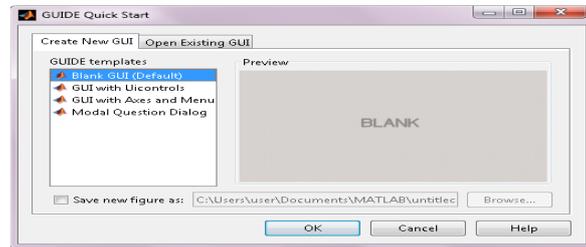
petunjuk untuk menentukan sesuatu penunjang yang dibutuhkan dalam mendesain (lihat gambar 5).

Perancangan sistem ini dibuat dalam dua bagian, yaitu bagian pemasukan referensi serta identifikasi kepadatan berdasarkan hasil *input* kamera pada sebuah *miniature* persimpangan *traffic light*.

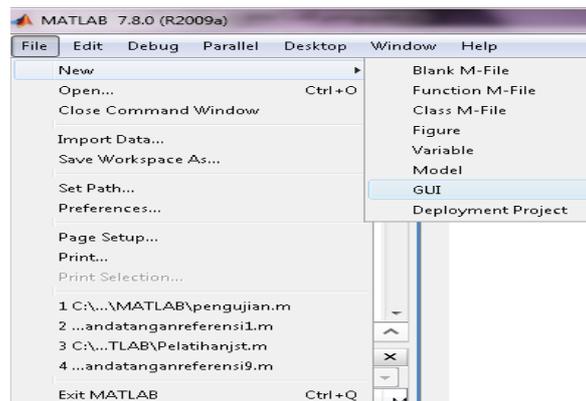
Sistem ini merupakan *loop* terbuka karena tidak ada umpan balik, *output* tidak berpengaruh pada *input*. Sistem ini bekerja saat pengguna menekan *pushbutton*.

**B. Perancangan Alat (Maket)**

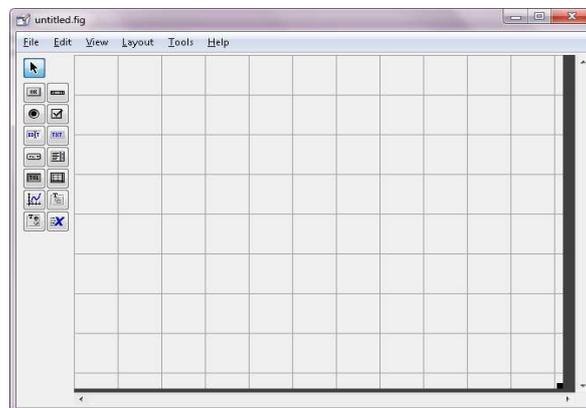
Skema perancangan Maket persimpangan Toar-Lumimuut untuk otomatis *traffic light* dapat dilihat pada



Gambar 9. GUIDE Quick Star



Gambar 8. Membuat GUI



Gambar 9. GUIDE Form

gambar 6 dan maket persimpangan yang digunakan pada gambar 7.

**C. Perancangan Program Pengolahan Citra Digital**

Untuk pengolahan citra digital dibuat dalam program Matlab 2009a dengan memanfaatkan fasilitas *GUI* (*Graphical User Interface*).

Jalankan *software* Matlab 2009a, kemudian klik *File > New*, pilih *GUI*. (Lihat gambar 8).

Muncul jendela “*GUIDE Quick Start*” (seperti pada gambar 9). Klik *Blank GUI (Default)* dan akan terbuka

jendela *GUI* yang berisi desain *visual* program (seperti pada gambar 10).

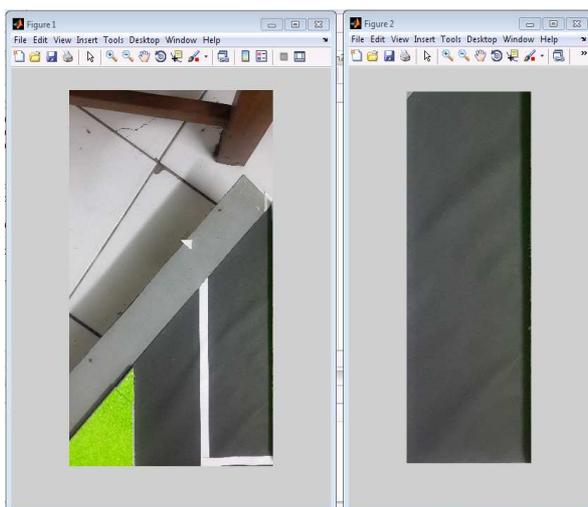
Desain *visual* program pada *GUI* Matlab dirancang menggunakan 1 *push button*, 4 *axes*, 32 *edit text*, 8 *static text* dan 5 *panel*. *Push button* digunakan untuk membuka citra, proses *grayscale*, proses *binerisasi*, dan analisa. *Axes* digunakan untuk menampilkan citra. *Edit text* digunakan untuk menampilkan hasil eksekusi program, *Static text* dan *Panel* digunakan sebagai keterangan (seperti pada gambar 11).

#### D. Proses Pengambilan Citra

Proses pengambilan citra dilakukan dalam mode *offline*. Proses pengambilan citra persimpangan *traffic light* ini menggunakan *camera digital*, dengan cara memposisikan kamera pada tempat yang sama dari tempat yang lebih tinggi, pengambilan citra ini dilakukan pada sebuah *miniatur* persimpangan *traffic light*. Citra yang diambil kemudian disalin kedalam memori penyimpanan program dan akan digunakan nanti pada proses selanjutnya.

#### E. Proses Cropping

Proses selanjutnya adalah proses pemotongan gambar atau *cropping* dengan menentukan batas-batas daerah yang akan di analisa dengan cara memanggil citra yang sudah disimpan lewat *m-file* yang ada di program matlab. Citra ditampilkan lewat *figure*, kemudian tentukan batas-batas daerah yang akan dianalisa lewat titik koordinat yang ditunjukkan pada *data cursor*. Ambil nilai-nilai yang didapatkan untuk dimasukkan pada proses pemotongan seperti pada gambar 10 & 11.



Gambar 10. Menentukan batas-batas daerah yang akan di analisa pada lokasi parkir yang kosong

Potong bagian dari koordinat yang telah di tentukan pada titik  $[X1\ Y1\ X2\ Y2]/[1200\ 1470\ 700\ 1520]$ , sehingga hanya daerah yang akan di pakai saja yang di ambil seperti pada gambar 10 dan 11.

Selanjutnya potong (*cropping*) menjadi 6 bagian seperti pada gambar 12 & 13. Akan ada 6 area yang diidentifikasi, area ini berada pada koordinat yang berbeda-beda. Seperti pada contoh *cropping* area A1 dan seterusnya. Contoh *listing* program yang digunakan untuk hasil *cropping* di atas :

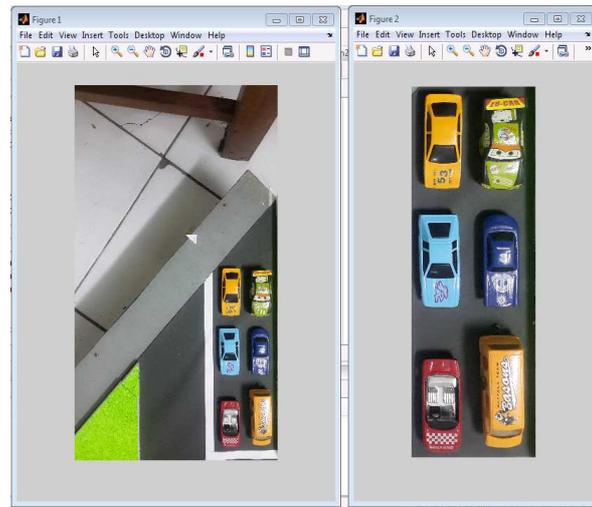
```
G=imread('D:\Skripsi\Citra\Gambar\utara.jpg');
A=imcrop(A, [1200 1470 700 1520]);
A1=imcrop(A1, [260 1030 240 480]);
A2=imcrop(A1, [260 480 240 480]);
A3=imcrop(A1, [260 10 240 480]);
A4=imcrop(A1, [10 1030 240 480]);
A5=imcrop(A1, [10 480 240 480]);
A6=imcrop(A1, [10 10 240 480]);
```

Hasil proses *cropping* citra Jalan kemudian dimasukan ke dalam program aplikasi yang dibuat menggunakan bahasa pemrograman *GUI* Matlab. Setelah itu data citra yang diambil di jadikan sebagai citra referensi ataupun sebagai citra yang akan diidentifikasi. Kemudian dilakukan proses *Thresholding*.

#### F. Proses Thresholding

Setelah melakukan *cropping* dilakukan proses *Thresholding*. Proses ini dilakukan untuk mengubah citra warna menjadi hitam putih.

Pengolahan citra dengan *thresholding* ini karena *pixel* objek dan *background* mempunyai level intensitas yang berbeda dalam dua *mode* dominan dari citra yang sudah di *binerisasi*. Pada citra yang ada *pixel* yang diberinilai 1 berkaitan dengan latar belakang



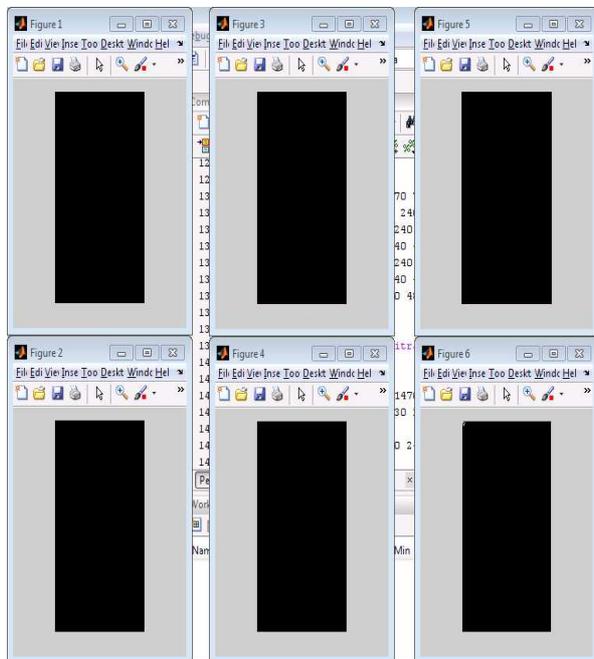
Gambar 11. Menentukan batas-batas daerah yang akan di analisa pada lokasi parkir yang penuh



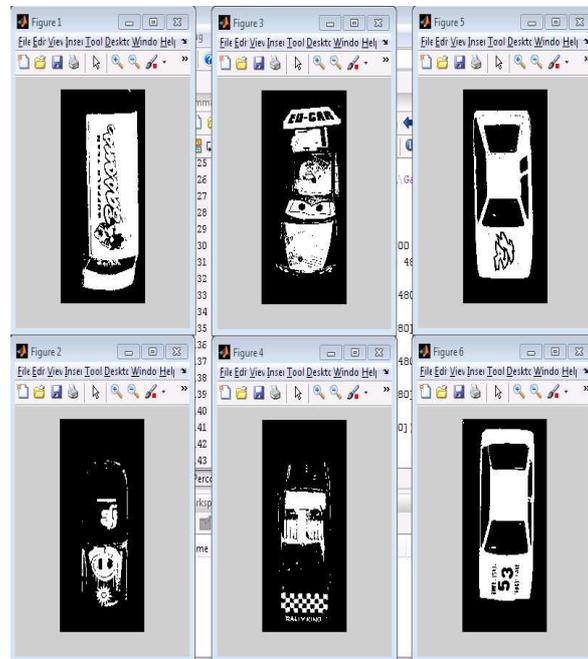
Gambar 12. Area A1 (kosong) pada koordinat [260 1030 240 480]



Gambar 13. Area A1 (terisi) pada koordinat [260 1030 240 480]



Gambar 14. Hasil Thresholding data referensi dengan nilai 0.6



Gambar 15. Hasil Thresholding data referensi dengan nilai 0.6

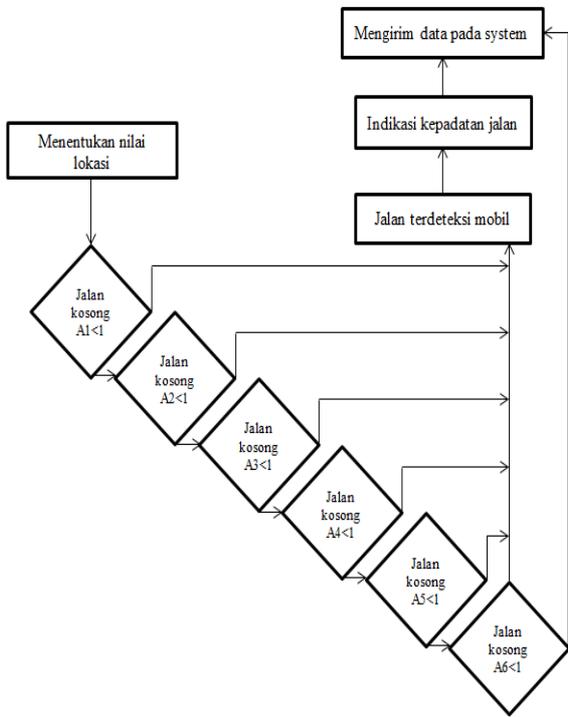
sedangkan *pixel* yang diberinilai 0 berkaitan dengan objek. Dalam memilih nilai *threshold* yang tepat pada tugas akhir ini digunakan metode *trial and error*, yaitu dengan mengambil beberapa *threshold* yang berbeda sampai mendapatkan 1 nilai *threshold* yang tepat dan memberikan hasil yang terbaik. Nilai ambang yang di dapat adalah [0.6]. jadi citra diubah ke *biner* dengan *threshold* [0.6] setelah itu lanjut ke proses berikutnya untuk membedakan nilai daerah yang berobjek dan tidak berobjek seperti pada gambar 14 & 15.

Citra referensi dan citra yang akan diidentifikasi ini kemudian akan disimpan setiap gambar yang di *crop* untuk dihitung nilai wilayahnya pada *bwarea* nanti.

### G. Metode Bwarea

Setelah melalui proses *cropping* dan *thresholding*, maka citra akan dianalisa. Proses analisa citra menggunakan metode *bwarea*.

*Bwarea* digunakan agar dapat membedakan nilai citra refrensi dengan nilai citra yang akan diidentifikasi. Dengan *bwarea* ini kita bisa mengetahui nilai wilayah yang terkandung dalam suatu gambar, sehingga bisa ditentukan dengan sendirinya batas nilai area yang kosong dengan batas nilai area yang sudah terisi mobil, nilai-nilai area tersebut di ambil rata-ratanya kemudian menjadi pembanding antara citra sekarang dengan citra refrensi seperti pada TABEL I.

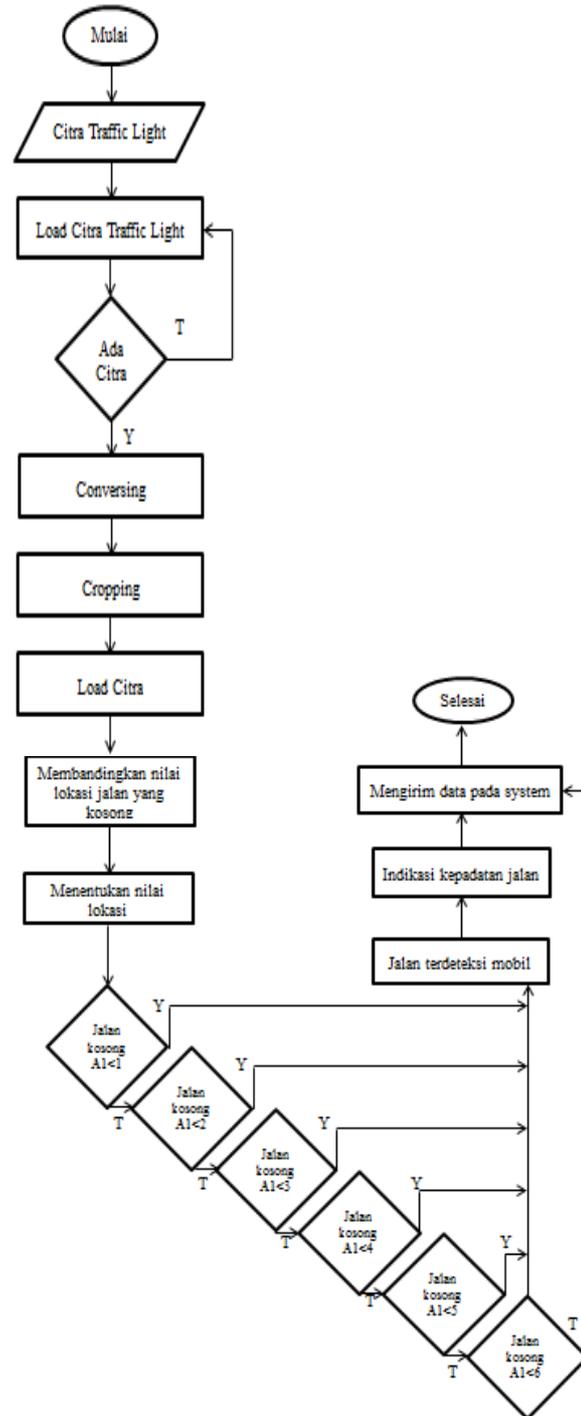


Gambar 16. Bagan proses analisa citra pada setiap jalur

TABEL I. PERBANDINGAN NILAI *BWAREA*

Area	Referensi	1	2	3	4	5	6
A1	115921	115921	115921	94976*	115921	112985*	113863*
A2	115916	115921	115921	115921	115921	115921	105747*
A3	115912	115911	115919	115906	115909	115900	100479*
A4	115921	112950*	113225*	111707*	112718*	112132*	112356*
A5	115921	115921	113688*	115921	113400*	113107*	113132*
A6	115921	115921	115920	115920	110996*	109690*	110296*

Note: '\*' = area tenisi



Gambar 17. Diagram Alir indikasi lampu hijau berdasarkan kepadatan kendaraan pada persimpangan *Traffic light*

Dari nilai-nilai yang di dapatkan di atas, setiap nilai referensi di kurangkan dengan nilai sample yang akan diidentifikasi kemudian di ambil nilai maksimum untuk lokasi yang kosong dan di tetapkan sebagai nilai acuan, contohnya :

Untuk area A1-A6 menggunakan data 1 & 4, citra referensi – Citra yang akan diidentifikasi = .....

115921–115921= 0	115921–115921=0
115916–115921= -5	115916–115921=5
115912–115911= 1	115912–115909=3
115921–112950 = 2971	115921–112718=3203
115921–115921= 0	115921–113400=2521
115921–115921= 0	115921–110996=4925

Berdasarkan perhitungan di atas keputusan nilai untuk lokasi yang kosong adalah  $<5$ , di ambil nilai maksimal yang mendekati agar program bisa berjalan dengan baik. Untuk posisi area yang lain dilakukan juga perhitungan dengan cara yang sama dan hasilnya sama namun ada beberapa lokasi yang berbeda mungkin karena perbedaan jarak yang mempengaruhi intensitas cahayanya. Diagram alir untuk metode *bwarea* bisa dilihat pada gambar 16.

#### H. Perancangan Perangkat Lunak

Pada perancangan perangkat lunak untuk otomatis *traffic light* bahasa pemrograman yang digunakan adalah bahasa pemrograman MATLAB 2009a.

Sistem otomatis *traffic light* dirancang agar dapat menentukan lama nyala lampu hijau berdasarkan kepadatan kendaraan untuk mengurangi kemacetan pada persimpangan *traffic light*. Diagram alir perancangan perangkat lunak otomatis *traffic light* dapat dilihat pada gambar 17.

## IV. HASIL DAN PEMBAHASAN

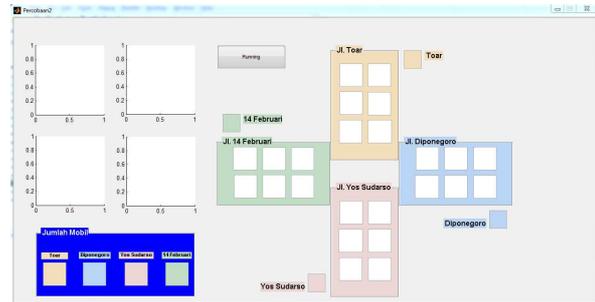
Proses pengujian otomatis *traffic light* dilakukan pada sebuah *miniatur* Persimpangan Toar-Lumimuut sebagai *input* pengambilan citra. *Input citra* diambil menggunakan kamera digital, sedangkan *output* pengolahan citra di tampilkan pada *software* MATLAB.

#### A. Pengenalan Objek

Proses pengujian untuk mengidentifikasi kepadatan jalan pada tiap jalur dilakukan dengan program yang dibuat dengan bahasa pemrograman GUI Matlab.

Program yang dibuat terbagi atas dua bagian yaitu bagian referensi dan bagian untuk pengidentifikasian. Citra referensi merupakan citra area parkir yang masih kosong dan dijadikan sebagai patokan nilai awal dari citra jalan.

Sedangkan citra yang lainnya akan diidentifikasi untuk penentuan kepadatan jalan yang kosong dan terisi mobil ketika program dijalankan nanti.



Gambar 18. Pemasukan Citra pada GUI Matlab

#### B. Penentuan Keputusan

Citra referensi telah disimpan kedalam program terlebih dahulu, sehingga saat *pushbutton* 'camera' di tekan akan muncul gambar persimpangan jalan. Sehingga saat menekan tombol *running* akan langsung memulai proses pengolahan data lokasi jalan yang akan diidentifikasi seperti terlihat pada gambar 18.

Semua prosesnya berada dalam program sehingga GUI Matlab tinggal menampilkan informasi dari hasil eksekusi program.

#### C. Penetapan Identitas Objek

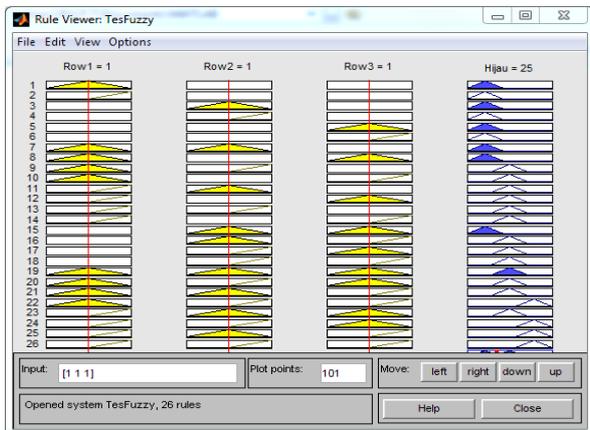
Setelah mendapatkan nilai – nilai *bwarea* dari citra referensi dan dari citra yang lain sudah dapat dibedakan batas jalan yang kosong dan yang terdeteksi mobil. Dikatakan area kosong dan bernilai 1, kita dapat mengambil contoh pada jalur Utara jika  $A1 < 1$ ,  $A2 < 1$ ,  $A3 < 1$ ,  $A4 < 1$ ,  $A5 < 1$ ,  $A6 < 1$ .

Sedangkan jika lebih besar dari nilai – nilai itu maka area itu sudah terisi mobil. *Threshold* digunakan agar bisa dengan tepat mendapatkan perbedaan nilai *pixel* dari objek dan nilai *pixel* dari latar belakang.

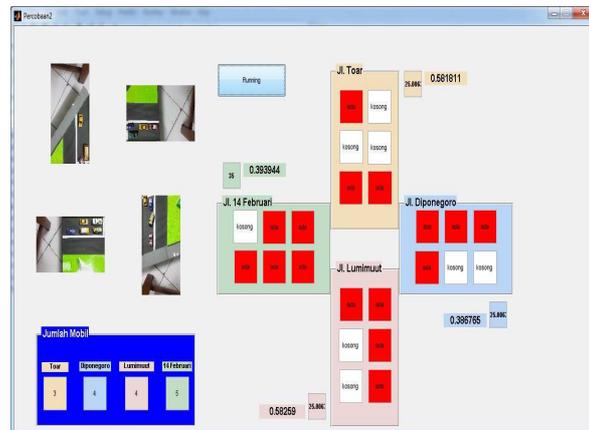
Kemudian untuk proses pengidentifikasiannya akan dilakukan dengan mengurangi nilai *bwarea* citra yang diidentifikasi dengan nilai *bwarea* citra referensi, jika hasilnya lebih besar dari nilai yang di tentukan maka area tersebut dinyatakan 0 yang artinya ada objek, sedangkan jika nilai pengurangannya lebih kecil maka area tersebut dinyatakan dengan 1 yang artinya kosong. Saat tombol *Running* di tekan maka program akan berjalan dengan sendirinya mengeksekusi hasil pemantau dan penentuan citra *traffic light*.

#### D. Pengujian Program Untuk Logika Fuzzy

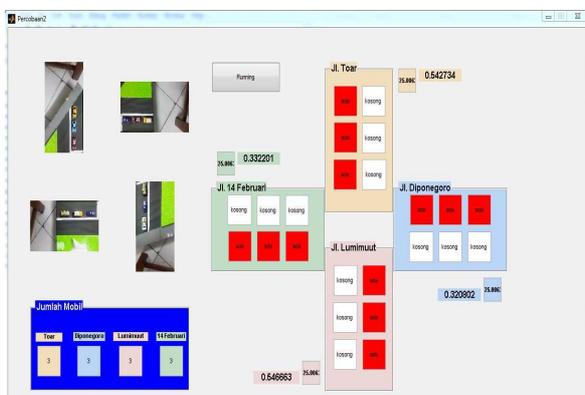
Yang diproses dalam logika *Fuzzy* adalah persimpangan jalan yang didapatkan lewat proses pengolahan citra digital. System logika *Fuzzy* yang digunakan menggunakan metode *mamdani* yang mempunyai 4 *input* yaitu Utara; Timur; Selatan; Barat sebagai indikasi kepadatan pada tiap persimpangan jalan, dan memiliki 1 *output* yaitu Hijau sebagai lama lampu hijau.



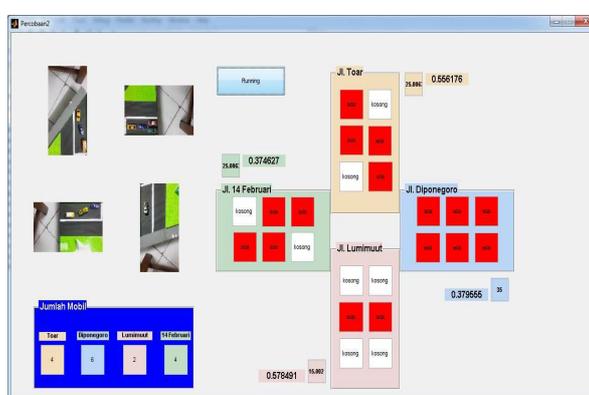
Gambar 19. Rule Viewer Sebagai Proses Penalaran Output



Gambar 21. Pengujian Program Berhasil



Gambar 20. Pengujian Program Berhasil



Gambar 22. Pengujian Program Gagal

Program akan melakukan perhitungan *Fuzzy* apabila data variabel dimasukkan. Data yang telah dimasukkan, dibaca untuk kemudian dikonversi dari nilai tegas menjadi nilai derajat keanggotaan *Fuzzy*. Nilai-nilai ini akan menjadi *input* untuk proses deFuzzyfikasi dengan menggunakan aturan (*Rule*) logika *Fuzzy* yang sudah ditentukan.

Dengan menggunakan rule pada gambar 19 bisa diambil kesimpulan hasil yang di dapati ketika *input* (jumlah mobil) bertambah maka *output* (lama lampu hijau) juga bertambah. Sebagai contoh dimasukan *input* 3 kendaraan pada tiap jalur dapat dilihat pada label *Input* [Utara; Timur; Selatan; Barat] dari kiri ke kanan. Sehingga *output* pada label Hijau (lama lampu hijau) = 35.

#### E. Pengujian Hasil

Pada pengujian sample seperti pada gambar 20 dan 21, posisi serta jumlah mobil yang terdeteksi sesuai seperti pada gambar, sehingga pengujian program berhasil. Sedangkan pada gambar 22, jumlah mobil yang terdeteksi berbeda dengan jumlah mobil pada

gambar, sehingga pengujian program tidak berhasil (mengalami *error*).

Telah dilakukan juga *Time Sampling* untuk pengujian lama waktu saat pengolahan citra ketika membuka (*load*) data citra (gambar) dari sebuah sample untuk di analisa, hasilnya lama waktu untuk membuka sebuah gambar bukan tergantung berdasarkan jumlah *pixel*, tapi tergantung pada resolusi dari sebuah gambar. Spesifikasi Laptop/PC juga berpengaruh pada lama waktu untuk membuka gambar untuk analisa citra.

Keakuratan pemantau dan penentuan citra dengan pengolahan citra digital menggunakan metode *Bwarea*, Yang didapat dengan rumus seperti pada gambar 23.

Berdasarkan hasil yang ada tingkat keakuratan mencapai 73,3333%, dapat dilihat bahwa program ini di jalankan masih secara *offline* pada *miniatur* persimpangan Toar-Lumimuut berdasarkan referensi yang ada. Masih belum efektif jika di jalankan pada lokasi persimpangan Toar-Lumimuut yang sebenarnya, harus di lakukan observasi untuk menentukan titik koordinat (wilayah) terlebih dahulu, agar bisa

$$\text{Keakuratan sistem} = \frac{\text{Jumlah pemantau dan penentuan yang benar}}{\text{Jumlah data uji}} \times 100\%$$

$$\text{Keakuratan sistem} = \frac{22}{30} \times 100\%$$

$$\text{Keakuratan sistem} = 73,3333\%$$

Gambar 23. Rumus Keakuratan Sistem

menyesuaikan dengan *system* yang dibuat. Karena *system* yang dibuat masih bekerja secara *offline*, pada persimpangan Toar-Lumimuut yang sebenarnya telah dilakukan penelitian dan observasi untuk dijadikan perbandingan lama nyala lampu hijau pada persimpangan yang sebenarnya dengan *miniature* persimpangan yang dibuat sesuai referensi berdasarkan jumlah kepadatan kendaraan.

Dari hasil penelitian dan observasi yang dilakukan untuk membandingkan persimpangan Toar-Lumimuut yang sebenarnya dengan *miniatur* persimpangan yang dibuat sesuai referensi, kepadatan kendaraan pada persimpangan Toar-Lumimuut sering terjadi pada jalur Jl. Lumimuut dan Jl. 14 Februari, akan tetapi pewaktuan lama nyala lampu hijau pada setiap jalur masih menggunakan normal *time*, dengan kata lain berapapun jumlah kendaraan yang terdapat pada tiap jalur tersebut, lama lampu hijau yang diberikan sama atau tidak berdasarkan kepadatan jumlah kendaraan, sehingga sering terjadi penumpukan kendaraan pada 2 jalur tersebut. Sedangkan pengujian pada *miniatur* persimpangan yang digunakan pada Tugas Akhir ini, lama lampu hijau tidak berdasarkan normal *time* melainkan berdasarkan jumlah kendaraan pada tiap jalur, sehingga dapat mengurangi kemungkinan terjadinya penumpukan kendaraan, karena waktu yang diberikan pada tiap jalur di atur berdasarkan kepadatan kendaraan yang ada pada jalur tersebut.

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Berdasarkan pengujian yang dilakukan maka dapat disimpulkan sebagai berikut:

Keakuratan sistem mencapai 73,3333% dari citra yang ada menggunakan logika *Fuzzy* dan pengolahan citra dengan metode *bwarea*. Masih terdapat *error* pada sistem yang dibuat, karena posisi mobil tidak selamanya akan berada pada batas-batas daerah (koordinat) seperti yang telah di analisa, misalnya posisi mobil berada di tengah jalan/miring sehingga menyebabkan *sample* yang terbaca oleh sistem berbeda dengan yang tertangkap oleh kamera.

Pengendalian sistem menggunakan Matlab lebih mudah.

Hasil penalaran untuk logika *Fuzzy* sangat mempengaruhi pemasukan *input* nilai pada *bwarea*, dimana *output* pada *bwarea* sama dengan *output* pada logika *Fuzzy*.

Posisi kamera mempengaruhi pembagian area saat pengolahan citra.

Intensitas cahaya berpengaruh pada nilai *bwarea*.

### B. Saran

Sistem dapat dikembangkan untuk bekerja secara *realtime* dan jaringan

Sistem dapat ditingkatkan lebih baik lagi dengan menambahkan metode-metode analisa citra yang lain dalam proses analisa.

Sistem dapat ditingkatkan keandalannya dengan menambahkan lebih banyak lagi *sample* citra yang digunakan.

Dapat dikembangkan dan diintegrasikan dengan sistem *traffic light* yang sudah ada sehingga nantinya sistem *traffic light* bisa menjadi lebih kompleks dan efisien.

## DAFTAR PUSTAKA

- [1] Ch. D. Sarante, "Rancang Bangun Sistem Lokasi Parkir Kamera Berbasis *Fuzzy Logic*", *Skripsi Program S1 Teknik Elektro Universitas Sam Ratulangi*. Manado, 2014.
- [2] E. Prasetyo, *Pengolahan Citra Digital dan Aplikasinya Menggunakan Matlab*, ANDI, Yogyakarta, 2011.
- [3] F. A. Irawan, M.Kom, Buku Pintar Pemrograman MATLAB, ANDI, Yogyakarta.
- [4] J. Ramesh, *Machine Vision*, 1995.
- [5] Kusumadewi, *Artificial intelligence (teknik dan aplikasinya)*, 2003.
- [6] R. A. Makalalag, "Perancangan Sistem Pemantau Dan Pentuan Tempat Parkir Berdasarkan *Digital Image Processing*", *Skripsi Program S1 Teknik Elektro Universitas Sam Ratulangi*. Manado, 2012.
- [7] R. Munir, *Fuzzy Logic dengan Menggunakan MATLAB. Bahan Ajar Teknik Informatika – STEI ITB*, Bandung.
- [8] T. Sutoyo *et al.* *Pencitraan Digital*, ANDI, Yogyakarta, 2009.
- [9] T. Sutoyo, S.Si, M.Kom., E. Mulyanto, S.Si, M.Kom., Suhartono, Dr., Nurhayati, Oky Dwi, M.T., Wijanarto, M.Kom. *Teori Pengolahan Citra Digital*, ANDI, Yogyakarta, 2009.