

## BAB 3

### METODOLOGI

#### 3.1 Alat dan Bahan

Pada penelitian ini diperlukan perangkat keras berupa laptop dan perangkat lunak untuk menjalankan simulasi, dengan spesifikasi sebagai berikut:

1. Intel® Core i5 5200U (3 MB Cache, 2.7 GHz)
2. RAM 8 GB (DDR4 2400 MHz)
3. Harddisk 500 GB
4. Nvidia GeForce GT 930M (2 GB, GDDR3)
5. Windows 10 Pro
6. MATLAB 2013b
7. Internet Network

#### 3.2 Alur Penelitian

Alur dari penelitian ini dimulai dari melakukan studi literatur untuk mengumpulkan referensi yang berhubungan dengan penelitian, kemudian melakukan percobaan pada program. Percobaan dilakukan untuk membandingkan citra referensi dengan citra yang diambil kamera secara *real-time*, kemudian data yang diperoleh berupa nilai persentase *match* diolah algoritme *fuzzy* dan akan dibedakan menjadi tiga dengan *fuzzy rule*, karena percobaan yang dilakukan dibagi menjadi tiga yaitu sepi, normal, dan padat. Dan yang terakhir algoritme *fuzzy* akan melakukan perhitungan untuk alokasi *green light time* yang efektif sesuai dengan kepadatan kendaraan di jalan. Setelah melakukan percobaan maka akan sampai pada tahap hasil, dalam tahap ini melakukan penelitian apakah berhasil atau tidak proses penelitian yang dilakukan.

#### 3.3 Studi Literatur

Tahap awal dalam proses mengerjakan penelitian ini adalah dilakukannya studi literatur, yang berguna untuk pengumpulan data penelitian sebelumnya, mencari referensi untuk menjadi acuan dalam memilih metode apa yang cocok untuk dilakukan dalam skripsi ini, dan mencari informasi mengenai parameter apa saja, baik yang akan digunakan atau hanya untuk menambah wawasan penulis. Studi literatur yang dilakukan dapat berasal dari jurnal lokal maupun

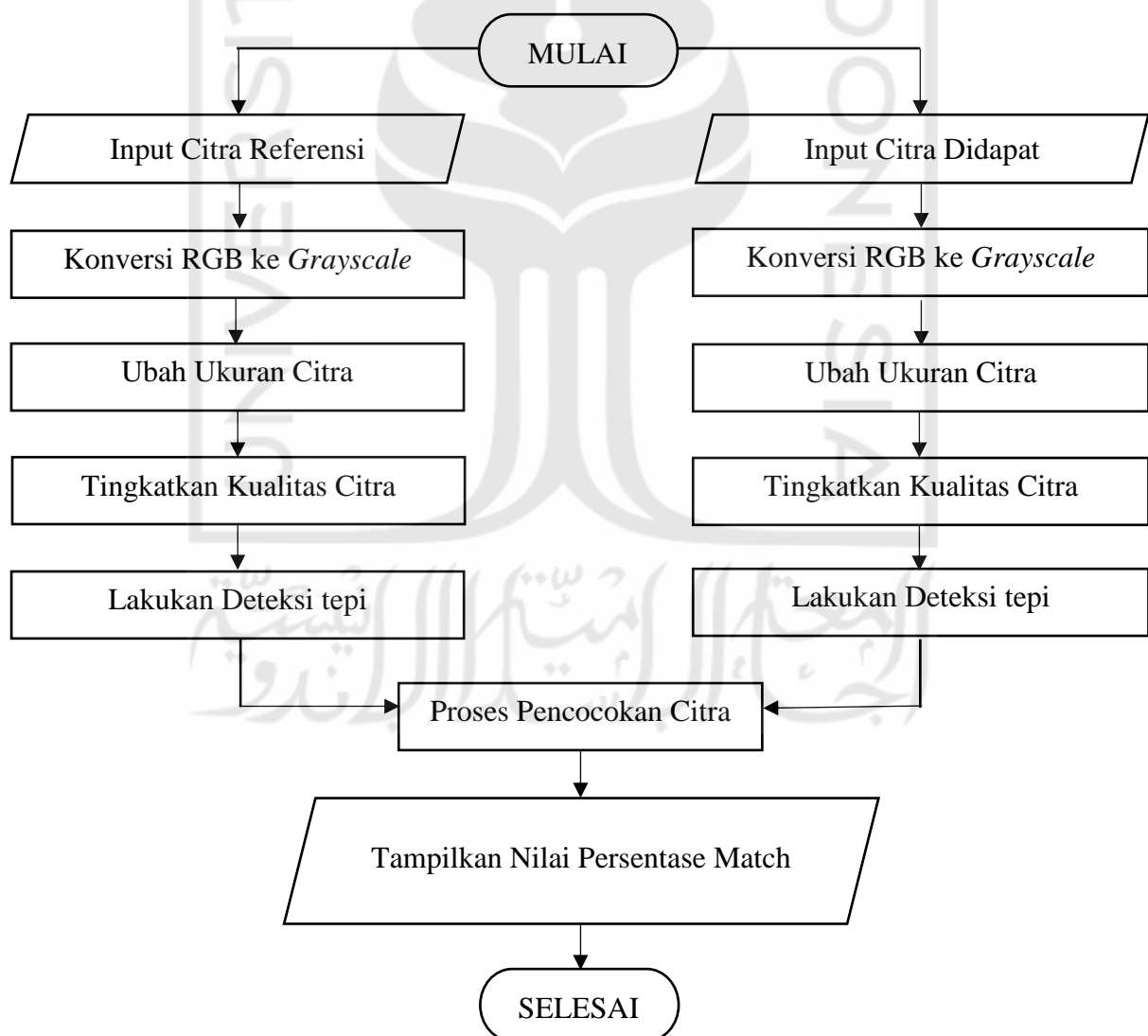
internasional, paper, website terpercaya, dll yang dapat dijadikan sebagai rujukan dalam melakukan penelitian.

### 3.4 Percobaan

Ditahap ini penulis melakukan pembuatan program MATLAB yang akan digunakan, dengan metode yang didapat setelah melakukan studi literatur sebelumnya. Setelah dibuatnya program akan dilakukan juga percobaan eksekusi pada program yang telah dirancang, untuk melihat apakah program yang dibuat hasilnya sudah sesuai dengan keinginan pada penelitian ini.

#### 3.4.1 Program *Image Processing*

Bagian ini menjelaskan proses yang terjadi di dalam program *Image Processing* yang dijalankan dengan software MATLAB 2013b, seperti ditunjukkan oleh diagram alir pada gambar 3.1.



Gambar 3.1 Diagram alir *image matching*.

### 1. Input citra referensi dan citra kamera

Tahap pertama yang dilakukan adalah memasukan citra yang telah diperoleh citra referensi dan citra yang akan dibandingkan dari kamera yang dipasang di persimpangan ke dalam program *image matching*. Citra ini disimpan di dalam satu folder dengan format JPEG karena yang umum digunakan dalam *image processing*.

### 2. Konversi citra RGB ke *Grayscale*

Pada tahap ini program melakukan proses mengkonversi atau mengubah citra yang berwarna menjadi citra degradasi abu-abu. Masing-masing jenis citra (*background* dan objek) akan diubah menjadi citra gradasi abu-abu. Tujuan dari proses ini adalah untuk memudahkan proses segmentasi yang ada pada tahap setelahnya, agar mudah untuk proses modifikasi [18].

### 3. Ubah ukuran citra

Tahap ini adalah tahap dimana program akan mengubah ukuran piksel citra yang dimasukkan menjadi ukuran yang dikehendaki yaitu 300x300 piksel. Proses ini dilakukan dengan metode *image resizing*.

### 4. Tingkatkan kualitas citra

Setelah diubah sesuai ukuran yang dikehendaki, selanjutnya meningkatkan kualitas citra dengan menggunakan metode *Image Enhancement* yang mana proses ini menyesuaikan citra digital sehingga hasilnya lebih bagus dan sesuai untuk tampilan atau analisis lebih lanjut. Misalnya, untuk menghilangkan *noise*, yang akan membuatnya lebih mudah untuk diidentifikasi.

### 5. Deteksi tepi

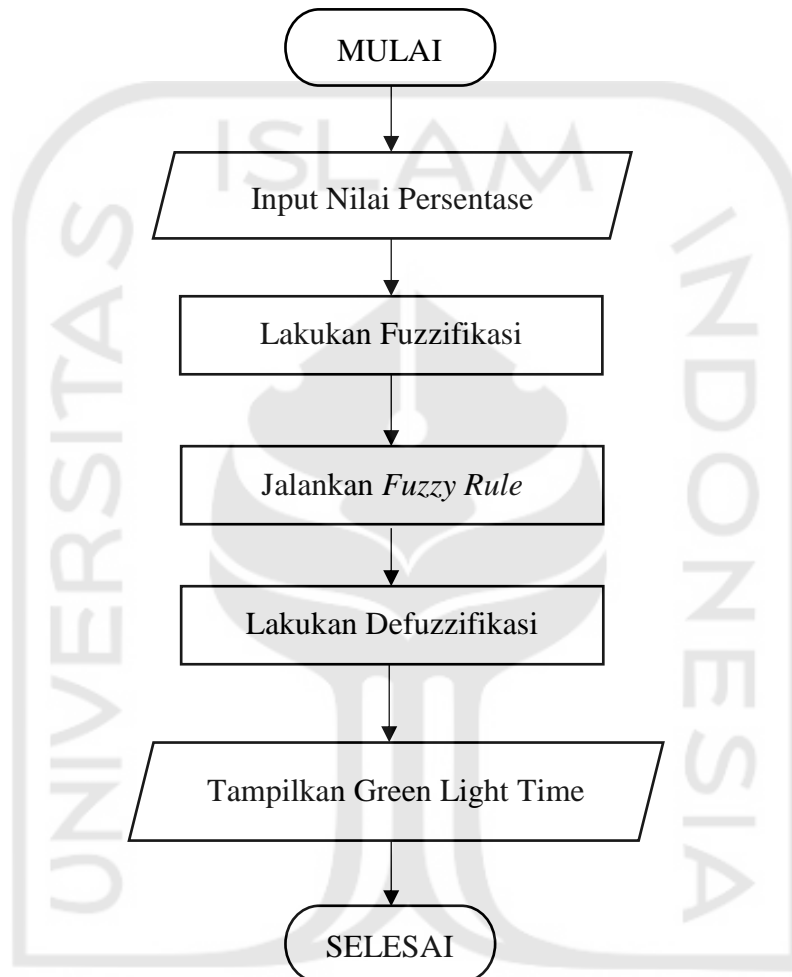
Setelah citra ditingkatkan kualitasnya maka citra masuk ke dalam proses maka akan dilakukan metode deteksi tepi Sobel, yang bertujuan mengidentifikasi titik-titik di citra digital di mana kecerahan citra berubah tajam atau, lebih teknis atau, memiliki diskontinuitas (*noise*). Titik-titik di mana kecerahan citra berubah dengan tajam biasanya disusun menjadi satu set segmen garis melengkung yang disebut tepi (*edge*) [8].

### 6. Bandingkan citra

Terakhir yaitu membandingkan piksel citra referensi dengan piksel citra yang diambil oleh kamera secara piksel per piksel.

### 3.4.2 Program Algoritme *Fuzzy*

Pada bagian ini dijelaskan proses pembuatan program dengan metode *fuzzy logic* untuk mengolah data yang berasal dari program sebelumnya. Tujuannya untuk melakukan perhitungan *green light time* yang sesuai dengan kepadatan kendaraan di persimpangan lalu lintas. Dengan metode ini diharapkan mampu mencapai hasil yang efektif untuk alokasi *green light time*.



Gambar 3.2 Diagram alir algoritme *fuzzy*.

#### 1. Input persentase

Langkah pertama adalah memasukan nilai persentase yang diperoleh proses sebelumnya ke dalam program *fuzzy*, nilai yang diinputkan yaitu hasil dari metode perbandingan citra yang sebelumnya sudah dilakukan. Nilai ini akan di masukkan ke dalam *fuzzy*.

## 2. Fuzzifikasi

Kedua nilai tadi akan diolah *fuzzy* dengan melakukan proses fuzzifikasi yaitu suatu metode yang akan mengubah inputan dari bentuk himpunan biasa (*crisp*) menjadi bentuk *fuzzy* (variabel linguistik) yang biasanya outputannya dalam bentuk himpunan-himpunan *fuzzy* dengan fungsi keanggotaannya masing-masing.

Tabel 3.1 Fuzzifikasi

Variabel Angka	Variabel Fuzzy	
	Persenstase Match	Keadaan Jalan
0-15 %	Kecil (S)	Sepi (L)
15-85 %	Normal (N)	Normal (N)
85-100 %	Besar (B)	Padat (H)

## 3. Dasar *Fuzzy Rule*

Langkah selanjutnya dalam dasar aturan *fuzzy* berisi pernyataan-pernyataan logika *fuzzy* (*fuzzy statement*), yang berbentuk pernyataan *IF-THEN*. Setelah dilakukan fuzzifikasi untuk setiap inputan, berikutnya yaitu membuat aturan (*rule base*). *Rule base* ini terdiri dari kumpulan aturan yang berbasis logika *fuzzy* untuk menyatakan suatu kondisi, penyusunan *rule base* sangat berpengaruh pada presisi model. Pada tahap pengambilan keputusan ditentukan berdasarkan rancangan *rule base*, aturan *IF-THEN* yang dihubungkan dengan logika operasi *AND-OR*.

Tabel 3.2 Parameter untuk *fuzzy rule*

Parameter	Nilai
Jumlah Input	2
Jumlah Output	1
Jumlah <i>Rules</i>	9
Lebar Jalan	7 meter
Jumlah Persimpangan	1
Jumlah Jalur	4
<i>Green Time</i> Min	8 detik
<i>Green Time</i> Maks	81 detik

Berdasarkan gambar 3.2 setelah metode perbandingan dilakukan dan menghasilkan output berupa nilai persentase perbandingan maka algoritme *fuzzy* akan mengoptimalkan alokasi green time dengan *rules* menggunakan parameter seperti diatas. Program ditulis menggunakan MATLAB untuk mengimplementasikan algoritme di atas diberikan dalam lampiran namun keluaran dari setiap langkah dan hasil akhir program diberikan pada bagian selanjutnya.

#### 4. Defuzzifikasi

Terakhir yaitu defuzzifikasi yang mengubah output *fuzzy* menjadi *crisp value* (nilai pasti) berdasarkan fungsi keanggotaan yang telah ditentukan. Proses perubahan data-data *fuzzy* tersebut menjadi data-data numerik yang dapat ditampilkan di layar komputer.

