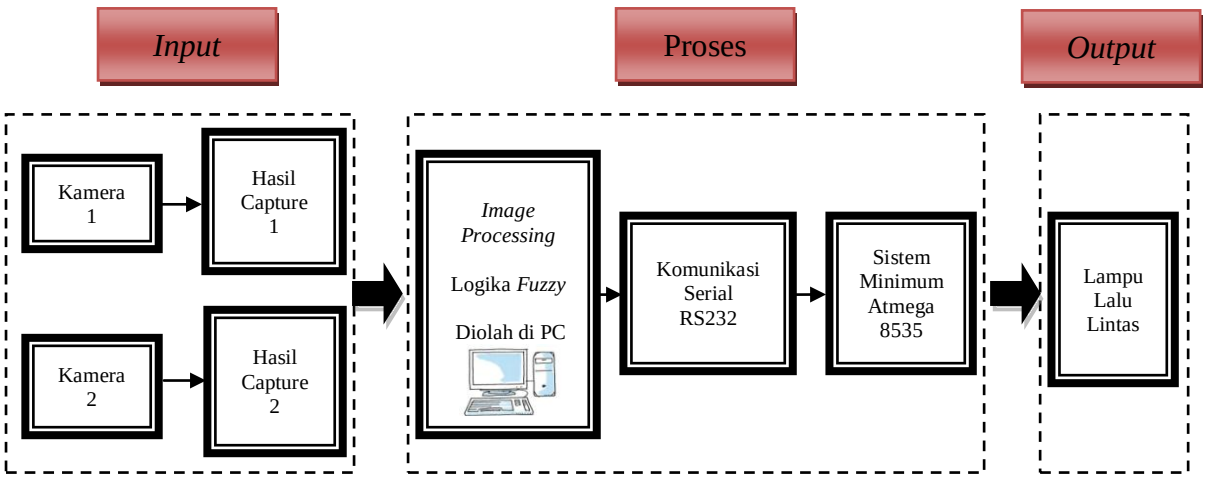
BAB III

PERANCANGAN ALAT DAN PEMBUATAN SIMULASI

Perancangan dan realisasi sistem merupakan bagian yang terpenting dalam seluruh pembuatan tugas akhir ini. Pada bab ini akan di jelaskan tentang pembuatan simulasi di MATLAB dan pembuatan *hardware* berupa miniature lampu lalu lintas.

3.1 Perancangan Sistem

Secara umum, perancangan dan pembuatan simulasi ini terdiri dari tiga bagian utama, yaitu masukan (*input*), proses dan keluaran (*output*) seperti yang terlihat pada **Gambar 3.1** dibawah ini. Setiap bagian dari sistem mempunyai fungsi masing-masing yang akan saling berkaitan dalam pengontrolan lampu lalu lintas ini.



Gambar 3.1 Blok Diagram Sistem

Pada **Gambar 3.1** memperlihatkan bagian-bagian dari perancangan sistem pengembangan algoritma pengolahan citra untuk mengontrol kepadatan lalu lintas pada persimpangan dua arah berbasis logika *Fuzzy*, yang meliputi beberapa bagian penting, yaitu:

a. bagian masukan (*input*)

Masukan (*input*) terdiri dari 2 (dua) yaitu hasil *capture* dari jalur 1 dan hasil *capture* dari jalur 2. Kedua masukan (*input*) ini disebut juga sebagai masukan awal untuk pengembangan algoritma pengolahan citra dalam pengontrolan kepadatan lalu lintas pada persimpangan dua arah berbasis logika *Fuzzy* ini. Dan kedua masukan ini nantinya akan diproses dengan software MATLAB untuk mendeteksi jumlah kendaraan dari hasil capure dari masing-masing jalur.

b. bagian proses

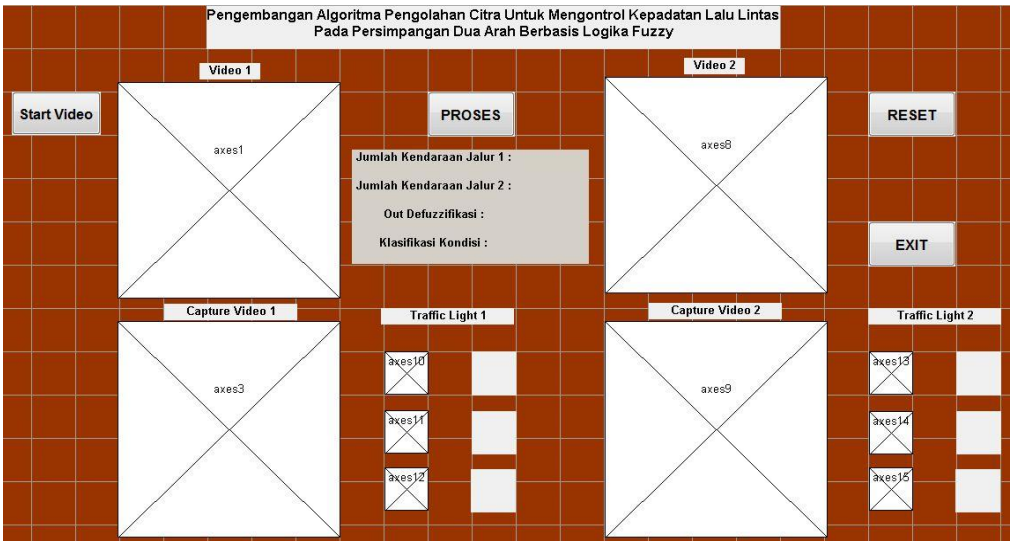
MATLAB merupakan *software* utama untuk mengolah hasil *capture* dari masing-masing jalur untuk mendeteksi jumlah kendaraan. Pendeteksian jumlah kendaraan dilakukan dengan cara membandingkan background dengan hasil *capture* dan dilanjutkan ke proses segmentasi. Setelah jumlah kendaraan dari masing-masing jalur didapat, maka dilanjutkan dengan proses logika *Fuzzy* yang mana prosesnya terdiri dari *Fuzzifikasi, Fuzzy Logic Inference*, dan *Defuzzifikasi*. *Output* dari hasil proses *logika Fuzzy* ini akan digunakan untuk membuat pengkondisian berupa kondisi-kondisi untuk penyalaan lampu lalu lintas. Kondisi-kondisi tersebut akan dikirim melalui komunikasi serial RS232 ke sistem

minimum mikrokontroller ATmega8535 untuk penyalaan lampu lalu lintas.

c. bagian keluaran (*output*)

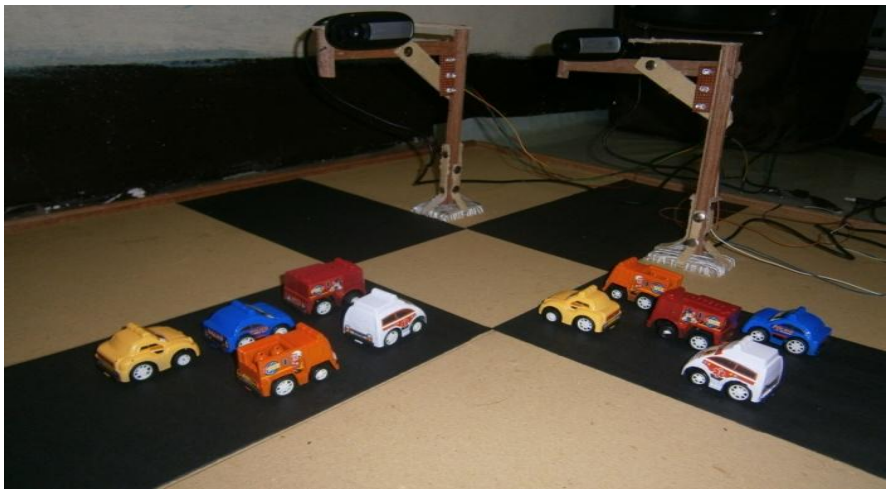
Bagian keluaran (*output*) ini merupakan hasil dari beberapa masukan (*input*) yang telah diproses pada bagian proses yang sudah dijelaskan sebelumnya. *Output* yang dihasilkan berupa kondisi-kondisi penyalaan lampu lalu lintas. Dalam sistem ini, lampu lalu lintas berada dalam dua tempat, yaitu yang pertama berada di dalam *software* MATLAB itu sendiri sebagai simulasi atau sebagai indikator untuk penyalaan lampu lalu lintas dan yang kedua berada di luar *software* atau berupa *hardware* LED yang dihubungkan dengan komunikasi serial RS232 ke sistem minimum ATmega8535.

Untuk membuat Tugas Akhir ini dibutuhkan suatu simulator dan miniature dua persimpangan untuk pengaturan lampu lalu lintas. Simulator yang digunakan adalah simulator MATLAB R2009b seperti yang terlihat di **Gambar 3.2** dibawah ini.



Gambar 3.2 Tampilan Simulasi di MATLAB

Sistem miniature lampu lalu lintas atau *prototype*, dimana sistem miniature ini hampir menyerupai sistem lampu lalu lintas aslinya. Dalam sistem miniature ini terdapat lampu lalu lintas di masing-masing jalur sebagai *output* dan kamera USB di masing-masing jalur sebagai masukan untuk pengambilan gambar (*capture*).



Gambar 3.3 Tampilan Miniature

Hasil capture akan diproses untuk mendeteksi dan mengetahui jumlah kendaraan yang ada di masing-masing jalur dan terdapat juga *output* berupa *led* sebagai lampu lalu lintas sebagai indikator untuk penyalaan lampu yang dihubungkan dengan menggunakan komunikasi serial RS232 ke sistem minimum ATmega8535.

3.2 Perancangan *Software*

Hasil *capture* merupakan masukan awal untuk melanjutkan ke proses selanjutnya. Hasil *capture* ini disebut sebagai citra asli yang nantinya akan diproses untuk mendeteksi dan mengetahui berapa jumlah kendaraan dari hasil

capture tersebut. Untuk mengetahui berapa jumlah kendaraan dari hasil *capture* dilakukan beberapa pemrosesan antara lain, konversi citra asli ke *grayscale*, konversi citra *grayscale* ke biner, deteksi tepi menggunakan *edge detection Canny*, proses segmentasi, dan penentuan jumlah kendaraan.

3.2.1 Konversi Citra Asli ke *Grayscale*

Pada tahap ini citra asli hasil capture akan dikonversikan ke bentuk mode warna yang lain yaitu mode warna *graycale*, hal ini bertujuan untuk menyederhanakan informasi-informasi warna yang terdapat pada ketiga layer matriks warna yakni *Red*, *Green*, and *Blue* (RGB) menjadi satu layer yang bernilai dari perkalian ketiga layer matriks warna tersebut dengan konstanta atau dengan cara melakukan rata-rata dari ketiga matriks.

Gambar 3.4a Citra Asli

Gambar 3.4b Citra *Grayscale*

Perubahan format citra asli (**gambar 3.4a**) menjadi format *grayscale* (**gambar 3.4b)** mengalami perubahan nilai. Dimana saat format gambar masih citra asli nilainya terdiri dari nilai *red* (R), *green* (G), dan *blue* (B), tetapi saaat format gambar sudah menjadi *grayscale* nilainya antara 0 dan 1, hal ini dikarenakan semua gambar berubah menjadi warna *gray* yang mempunyai nilai 0 sampai 1.

Gambar 3.5 Citra *Grayscale* Bernilai Antara 0 Sampai 1

3.2.2 Konversi ke Biner

Pada tahap ini citra yang telah melalui tahap konversi ke bentuk *grayscale* akan dilakukan konversi kembali kebentuk biner, tahap konversi ini sangatlah penting mengingat dalam proses pengolahan selanjutnya akan dilakukan proses pengolahan citra yang bernilai 1 dan 0.

Gambar 3.6a Citra *Grayscale*

Gambar 3.6b Citra Biner

Perubahan format *grayscale* (**Gambar3.6a**) menjadi format biner (**Gambar 3.6b)** mengalami perubahan nilai. Dimana saat format gambar masih *grayscale* nilainya antara 0 – 1, tetapi saat format gambar sudah menjadi biner nilainya menjadi 0 dan 1, dimana gambar yang berwarna putih mempunyai nilai 1 dan gambar yang berwarna hitam mempunyai nilai 0.

Gambar 3.7 Citra Biner bernilai 1 dan bernilai 0

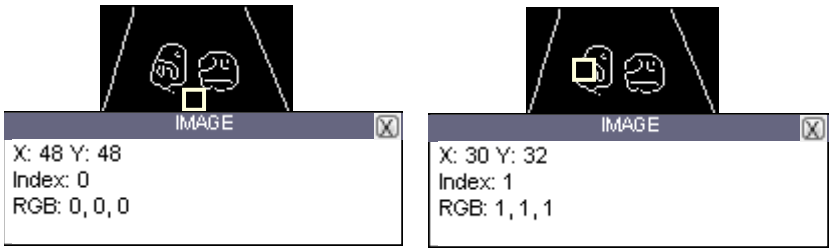
3.2.3 Deteksi Tepi dengan Metode *Canny Edge Detection*

Pada tahap ini citra yang telah melalui tahap konversi ke bentuk biner akan dilakukan pendetesian tepi dengan menggunakan metode *canny edge detection* untuk mengurangi *noise* sebelum melakukan perhitungan deteksi tepi sehingga tepi-tepi yang dihasilkan lebih banyak.

Gambar 3.8a Citra Biner

Gambar 3.8b *Canny Edge Detection*

Pendeteksian tepi akan mempermudah untuk mendeteksi jumlah kendaraan. Hal ini dikarenakan nilai-nilai setiap piksel dari gambar yang sudah dideteksi tepi tegas, sehingga sulit untuk mendeteksi nilai bernilai 1 dan bernilai 0.



Gambar 3.9 Pendeteksian tepi bernilai 1 dan bernilai 0

3.2.4 Proses Segmentasi

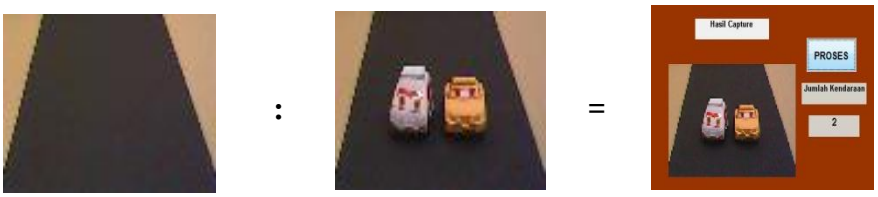
Segmentasi merupakan sebuah proses untuk memisahkan sebuah objek dari *background* atau latar, sehingga objek tersebut dapat diproses untuk

mendeteksi jumlah objek yang ada bersama *background* atau latar. Proses segmentasi dalam tugas akhir ini adalah hanya sebagai *window* atau *segmen* untuk melihat kendaraan, dimana 1 *window* atau *segmen* hanya untuk melihat 1 kendaraan. Berarti dalam 1 gambar terdapat 18 *window* atau 18 *segment*, sehingga maksimum jumlah kendaraan yang terhitung ada sebanyak 18 kendaraan.

Gambar 3.10 Proses Segmentasi

3.2.5 Pendeteksian Jumlah Kendaraan

Penentuan jumlah kendaraan ditentukan berdasarkan perbandingan *background* dengan hasil *capture* yang sudah ada objeknya dengan proses *image processing*. Tahapan ini sangatlah penting untuk melanjutkan ke tahapan berikutnya, karena jumlah kendaraan yang sudah didapat akan menjadi masukan (*input*) untuk proses logika *Fuzzy*.



Gambar 3.11 *Background*, Citra Asli, dan Hasil Deteksi

Jumlah objek dari hasil capture ada sebanyak 2. Setelah dilakukan proses segmentasi dan perbandingan background dengan hasil capture maka jumlah yang terdeteksi ada sebanyak 2.

3.3 Logika *Fuzzy*

Perancangan sistem kontrol berbasis logika *Fuzzy*, memerlukan beberapa tahapan yaitu pembentukan himpunan *Fuzzy*, penentuan fungsi keanggotaan, dan penentuan basis aturan atau kaidah *Fuzzy*. Metode yang digunakan dalam tugas akhir ini adalah metode FIS sugeno.

3.3.1 Pembentukan Himpunan *Fuzzy*

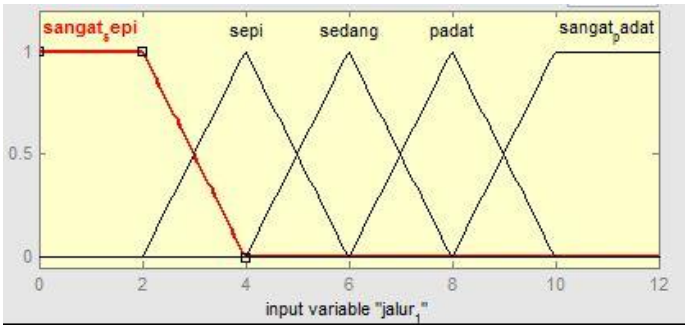
Jumlah kendaraan yang sudah didapat dari masing-masing jalur dibagi menurut kepadatan. Kondisi-kondisi kepadatan untuk jalur 1 sama dengan kondisi-kondisi kepadatan untuk jalur 2, yakni sangat sepi, sepi, sedang, padat, dan sangat padat, sedangkan untuk *output* adalah sangat sedikit, sedikit, normal, banyak, dan sangat banyak.

Tabel 3.1 Himpunan *Fuzzy*

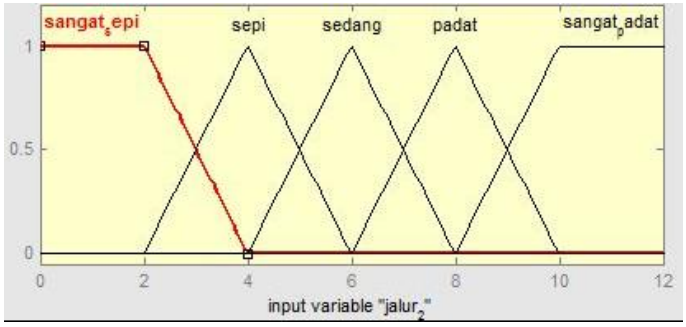
Fungsi	Variabel	Himpunan *Fuzzy*
Input	Jalur 1	Sangat Sepi (SS) Sepi (S) Sedang (SD) Padat (P) Sangat Padat (SP)
	Jalur 2	Sangat Sepi (SS) Sepi (S) Sedang (SD) Padat (P) Sangat Padat (SP
Output		Sangat Sedikit (SSDT) Sedikit (SDT) Normal (N) Banyak (B) Sangat Banyak (SB)

3.3.2 Fungsi Keanggotaan

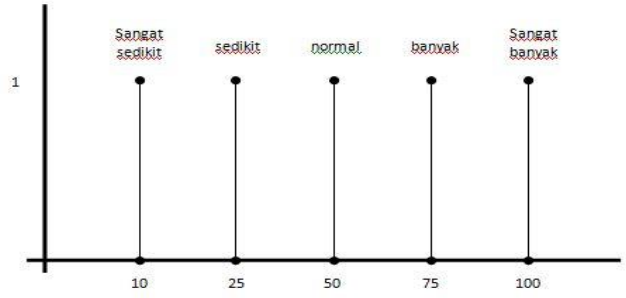
Fungsi keanggotaan berguna untuk proses perubahan nilai input umum ke dalam bentuk input *Fuzzy*, dengan derajat keanggotaan μ(x) yang bernilai antara 0 sampai 1 pada himpunan *Fuzzy*. Digunakan fungsi kurva segitiga dan kurva trapesium untuk mempresentasikannya.



Gambar 3.12 *Membership Function* Untuk Jalur 1



Gambar 3.13 *Membership Function* Untuk Jalur 2



Gambar 3.14 *Membership Function Output*

3.3.3 Basis Kaidah *Fuzzy* (*Fuzzy Rule Base*)

Basis Kaidah *Fuzzy* (*Fuzzy Rule Base*) merupakan himpunan implikasi-implikasi yang berlaku sebagai aturan dalam sistem. Aturan *Fuzzy ditetapkan* terlebih dahulu agar sistem kontrol *Fuzzy* sesuai dengan keadaan. Kaidah atau *rule* yang digunakan adalah :

IF (anteseden 1) AND (anteseden 2) THEN (konsekuen)

Tabel 3.2 Aturan *Fuzzy* Untuk Kedua Jalur

RULE	*INPUT*		*OUTPUT*
	JALUR 1	**JALUR 2**	
1	SS	SS	N
2	SS	S	SDT
3	SS	SD	SDT
4	SS	P	SSDT
5	SS	SP	SSDT
6	S	SS	B
7	S	S	N
8	S	SD	SDT
9	S	P	SSDT
10	S	SP	SSDT
11	SD	SS	SDT
12	SD	S	SDT
13	SD	SD	N
14	SD	P	SDT
15	SD	SP	SSDT
16	P	SS	SSDT
17	P	S	SB
18	P	SD	B
19	P	P	N
20	P	SP	N
21	SP	SS	SB
22	SP	S	SB
23	SP	SD	B
24	SP	P	N
25	SP	SP	N

Dari **Tabel 3.2** di bawah ini, dapat dibuat 25 *rule* atau aturan *Fuzzy* yang nanti digunakan dalam proses inferensi, sebagai berikut:

Rule 1 : IF jumlah kendaraan jalur 1 Sangat Sepi AND jumlah kendaraan jalur 2 Sangat Sepi THEN N

Rule 2 : IF jumlah kendaraan jalur 1 Sangat Sepi AND jumlah kendaraan jalur 2 Sepi THEN output SDT

Rule 3 : IF jumlah kendaraan jalur 1 Sangat Sepi AND jumlah kendaraan jalur 2 Sedang THEN output SDT

Rule 4 : IF jumlah kendaraan jalur 1 Sangat Sepi AND jumlah kendaraan jalur 2 Padat THEN output SSDT

Rule 5 : IF jumlah kendaraan jalur 1 Sangat Sepi AND jumlah kendaraan jalur 2 Sangat Padat THEN output SSDT

dan seterusnya

Keterangan :

- SS = sangat sepi, S = sepi, SD = sedang, P = padat, SP = sangat padat
- SSDT = sangat sedikit, SDT = sedikit, N = normal, B = banyak, SB = sangat banyak
- sangat sedikit terjadi apabila,
    1. jalur 1 ‘SS’ dan jalur 2 ‘P’
    2. jalur 1 ‘SS’ dan jalur 2 ‘SP’
    3. jalur 1 ‘S’ dan jalur 2 ‘P’
    4. jalur 1 ‘S’ dan jalur 2 ‘SP’
    5. jalur 1 ‘SD’ dan jalur 2 ‘SP’
- sedikit terjadi apabila,

1. jalur 1 ‘SS’ dan jalur 2 ‘S’
2. jalur 1 ‘SS’ dan jalur 2 ‘SD’
3. jalur 1 ‘S’ dan jalur 2 ‘SD’
4. jalur 1 ‘SD’ dan jalur 2 ‘SS’
5. jalur 1 ‘SD’ dan jalur 2 ‘S’
6. jalur 1 ‘SD’ dan jalur 2 ‘P

- normal terjadi apabila,
    1. jalur 1 ‘SS’ dan jalur 2 ‘SS’
    2. jalur 1 ‘S’ dan jalur 2 ‘S’
    3. jalur 1 ‘SD’ dan jalur 2 ‘SD’
    4. jalur 1 ‘P’ dan jalur 2 ‘P’
    5. jalur 1 ‘P’ dan jalur 2 ‘SP’
    6. jalur 1 ‘SP’ dan jalur 2 ‘P’
    7. jalur 1 ‘SP’ dan jalur 2 ‘SP’
- banyak terjadi apabila,
    1. jalur 1 ‘S’ dan jalur 2 ‘SS’
    2. jalur 1 ‘P’ dan jalur 2 ‘SD’
    3. jalur 1 ‘SP’ dan jalur 2 ‘SD’
- sangat banyak terjadi apabila,
    1. jalur 1 ‘P’ dan jalur 2 ‘SS’
    2. jalur 1 ‘P’ dan jalur 2 ‘S’
    3. jalur 1 ‘SP’ dan jalur 2 ‘SS’
    4. jalur 1 ‘SP’ dan jalur 2 ‘S’

3.3.4 Kondisi Penyalaan Lampu lalu lintas

Kondisi penyalaan lampu lalu lintas ditetapkan berdasarkan *output* dari proses *defuzzifikasi* dalam proses *Fuzzy*. Penentuan kondisi penyalaan lampu lalu lintas ada sebanyak 5 (lima) kondisi, antara lain kondisi 1, kondisi 2, kondisi 3, kondisi 4, dan kondisi 5. Berikut kondisi-kondisi penyalaan lampu lalu lintas.

Tabel 3.3 Kondisi Penyalaan Lampu Lalu Lintas

Output Defuzzifikasi	Status Kondisi
5 ≤ *output defuzzifikasi* < 21	Kondisi 1
21 ≤ *output defuzzifikasi* < 41	Kondisi 2
41 ≤ *output defuzzifikasi* < 61	Kondisi 3
61 ≤ *output defuzzifikasi* < 81	Kondisi 4
81 ≤ *output defuzzifikasi* < 101	Kondisi 5

Adapun lama waktu nyala lampu lalu lintas untuk masing-masing kondisi adalah sebagai berikut.

1. Nyala lampu lalu lintas untuk kondisi 1:

Tabel 3.4 Lama Waktu Kondisi 1

Jalur 1	Merah			Hijau	Kuning	Merah
Jalur 2	Hijau	kuning	Merah			
Waktu	10 detik	2 detik	1 detik	70 detik	2 detik	1 detik

2. Nyala lampu lalu lintas untuk kondisi 2:

Tabel 3.5 Lama Waktu Kondisi 2

Jalur 1	Merah			Hijau	Kuning	Merah
Jalur 2	Hijau	kuning	Merah			
Waktu	20 detik	2 detik	1 detik	50 detik	2 detik	1 detik

3. Nyala lampu lalu lintas untuk kondisi 3:

Tabel 3.6 Lama Waktu Kondisi 3

Jalur 1	Merah			Hijau	Kuning	Merah
Jalur 2	Hijau	kuning	Merah			
Waktu	40 detik	2 detik	1 detik	40 detik	2 detik	1 detik

4. Nyala lampu lalu lintas untuk kondisi 4:

Tabel 3.7 Lama Waktu Kondisi 4

Jalur 1	Merah			Hijau	Kuning	Merah
Jalur 2	Hijau	kuning	Merah			
Waktu	50 detik	2 detik	1 detik	20 detik	2 detik	1 detik

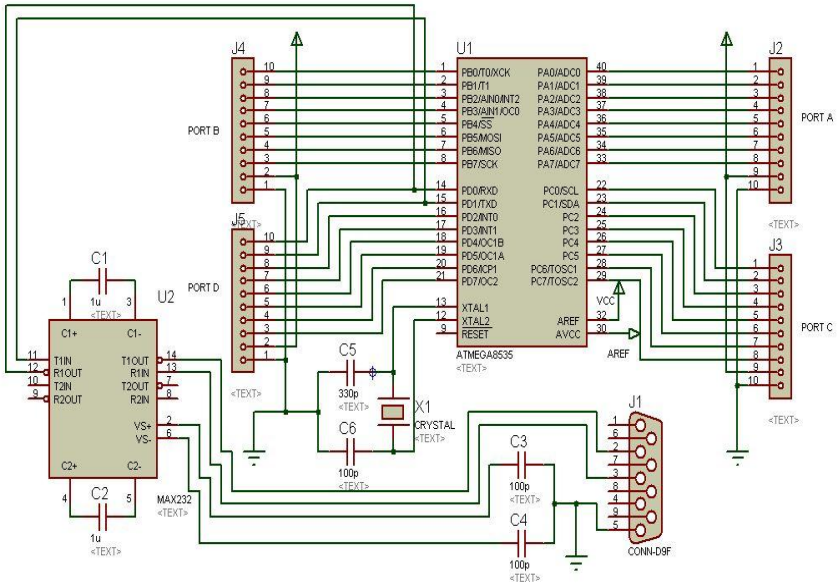
5. Nyala lampu lalu lintas untuk kondisi 5:

Tabel 3.8 Lama Waktu Kondisi 5

Jalur 1	Merah			Hijau	Kuning	Merah
Jalur 2	Hijau	kuning	Merah			
Waktu	70 detik	2 detik	1 detik	10 detik	2 detik	1 detik

3.4 Rangkaian Sistem Minimum Mikrokontroler AVR ATmega8535

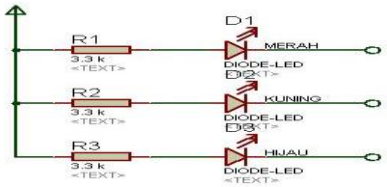
Sistem minimum Atmega8535 adalah rangkaian untuk mengoperasikan IC Atmega8535. Mikrokontroler hanya digunakan sebagai pengendalian lampu lalu lintas, dimana data yang diterima dari komunikasi serial RS232 akan diolah dan diteruskan ke *led* sebagai lampu lalu lintas. Pemilihan jenis mikrokontroler Atmega8535 ini karena program untuk mentransfer data komunikasi serial RS232 ke led lebih mudah dibandingkan mikrokontroler ATmega yang lain. Berikut adalah skematik dari rangkaian sistem minimum mikrokontroler Atmega8535:



Gambar 3.15 Skematik Sistem Minimum ATmega8535

3.5 Rangkaian Lampu Lalu Lintas

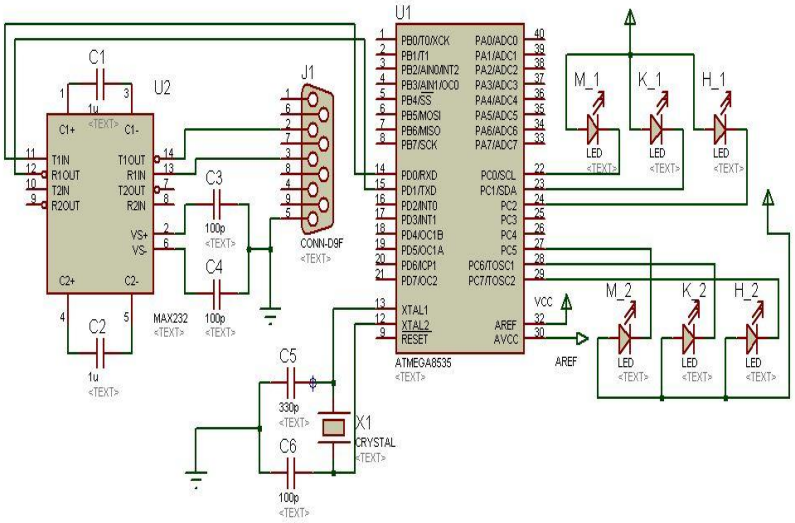
Pada tugas akhir ini, *led* digunakan sebagai lampu lalu lintas yang terdiri dari merah, kuning, dan hijau. Posisi lampu lalu lintas ini berada di persimpangan jalur 1 dan persimpangan jalur 2. Apabila lampu lalu lintas di jalur 1 merah maka lampu lalu lintas di jalur 2 hijau, dan sebaliknya.



Gambar 3.16 Skematik Lampu Lalu Lintas

Masukan untuk lampu lalu lintas ini berupa data yang diterima sistem minimum Atmega8535 dari komunikasi serial RS232. Kondisi penyalaan lampu lalu lintas akan sesuai dengan kondisi penyalaan lampu lalu lintas yang ada di pemrograman MATLAB dan di pemrograman sistem minimum ATmega8535.

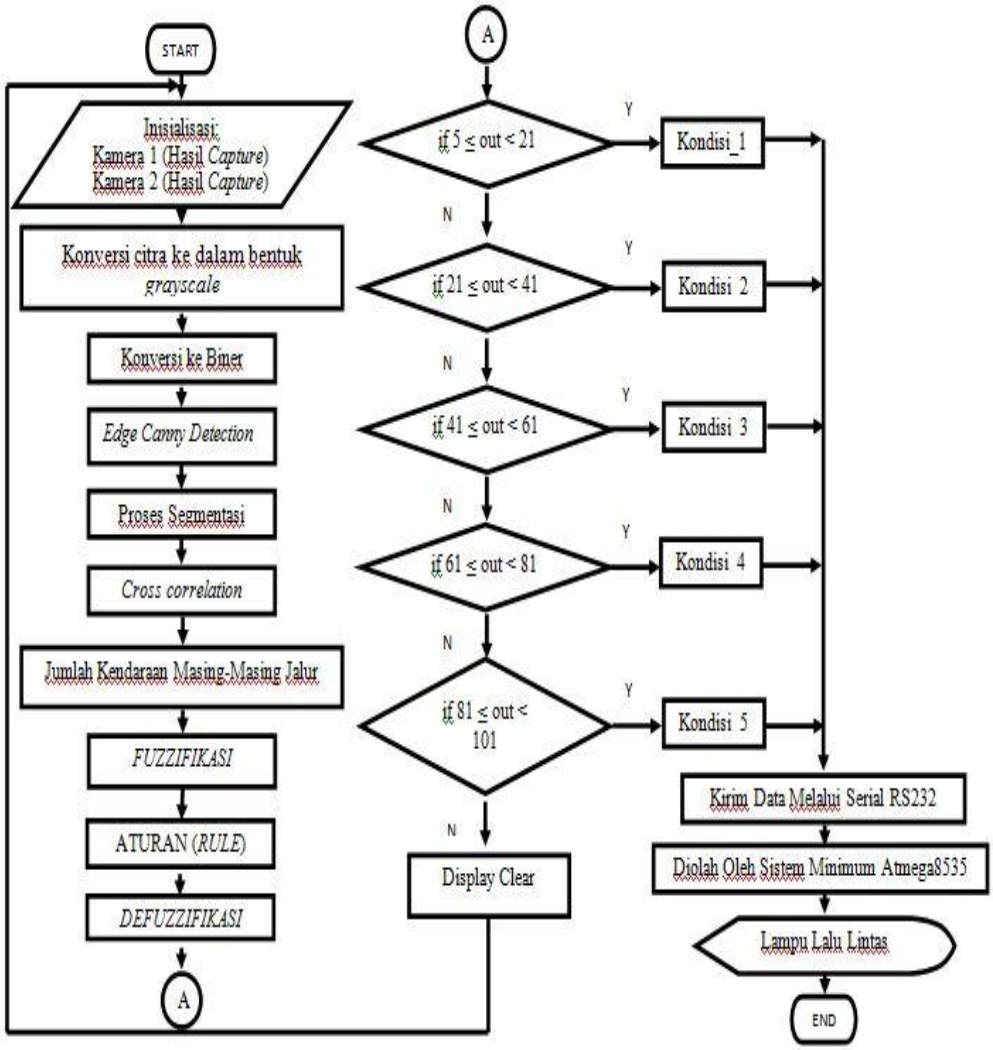
Berikut sekematik keseluruhan sistem minimum ATmega8535 dengan keluaran berupa *led* sebagai lampu lalu lintas.



Gambar 3.17 Skematik Gabungan Sistem Minimum ATmega8535 dengan Lampu Lalu Lintas

3.6 Perancangan Perangkat Lunak Sistem

Tahapan untuk sistem pengembangan algoritma pengolahan citra untuk mengontrol kepadatan lalu lintas pada persimpangan dua arah berbasis logika *Fuzzy* ini menggunakan MATLAB 2009b yang mana terhubung dengan komunikasi serial RS232 ke sistem minimum untuk menyalakan *output led* sebagai lampu lalu lintas. Flowchart dari sistem ini dapat dilihat pada **Gambar 3.17** di bawah ini.



Gambar 3.18 *Flowchart* Sistem Pengembangan Algoritma Pengolahan Citra Untuk Mengontrol Kepadatan Lalu Lintas pada Persimpangan Dua Arah Berbasis Logika *Fuzzy*

Pada saat kamera di masing-masing jalur *on*, akan dilakukan pengambilan gambar (*capture image*), hasil *capture* akan diubah ke dalam format *grayscale*. Hal ini dilakukan agar mempermudah mengubah format gambar ke dalam bentuk biner. Dengan format yang sudah dalam bentuk biner, dilakukan *edge Canny detection* untuk mendeteksi bentuk kendaraan. *Edge Canny detection* mencari objek kendaraan berdasarkan kontur tepinya. Sebelum dilakukan proses *cross*

correlation, terlebih dahulu hasil *capture* disegmentasi. Hal ini bertujuan untuk mempermudah dalam membandingkan hasil *capture* dengan *background* untuk mendeteksi jumlah kendaraan.

Setelah didapat jumlah kendaraan, maka dilanjutkan ke sistem logika *Fuzzy* dengan proses *fuzzifikasi, fuzzy logic inference*, dan *defuzzifikasi*. *Output* dari hasil proses *logika Fuzzy* ini akan digunakan untuk membuat pengkondisian berupa kondisi-kondisi untuk penyalaan lampu lalu lintas. Kondisi-kondisi tersebut akan dikirim melalui komunikasi serial RS232 ke sistem minimum mikrokontroller ATmega8535 untuk penyalaan lampu lalu lintas.