

## **SIMULASI PENGATURAN LAMPU LALU LINTAS BERDASARKAN DATA IMAGE PROCESSING KEPADATAN KENDARAAN BERBASIS MIKROKONTROLER ATMEGA16**

**Sunu Jatmika, Indra Andiko**  
STMIK Asia Malang

### **ABSTRAK**

Di beberapa kota di Indonesia, kemacetan lalu lintas merupakan salah satu masalah yang harus segera diatasi. Salah satu solusi yang dapat diterapkan untuk mengurangi kemacetan lalu lintas adalah dengan membuat sistem pengaturan lampu lalu lintas yang mampu mengatur kelacaran arus lalu lintas di persimpangan.

Seiring dengan kemajuan teknologi, ditemukan berbagai metode pendukung pengaturan lampu lalu lintas. Salah satu metode yang sedang berkembang saat ini adalah penggunaan kamera webcam. Dengan mengimplementasikan kamera webcam sebagai detektor kepadatan kendaraan di jalan raya diharapkan nyala lampu lalu lintas dapat menyesuaikan dengan tingkat kepadatan kendaraan dan menghitung pewaktuan lampu lalu lintas secara otomatis dan real time.

Sistem pengaturan lampu lalu lintas dengan kamera webcam ini bekerja berdasarkan inputan berupa gambar hasil capture. Kemudian gambar diproses dari format RGB menjadi grayscale, thresholding dan histogram. Proses ini dilakukan menggunakan personal computer yang selanjutnya data hasil kepadatan ini dikirim ke mikrokontroler ATmega16 melalui port serial.

Dari hasil pengujian didapatkan bahwa rata-rata presentase kepadatan kendaraan dipagi dan siang hari adalah 30% dan 28% artinya kondisi kepadatan kendaraan adalah sepi. Sedangkan pengujian yang dilakukan pada sore hari dan malam hari didapatkan bahwa rata-rata presentase kepadatan kendaraan adalah 53% dan 58% artinya kondisi kepadatan kendaraan pada sore hari dan malam hari adalah normal.

**Kata Kunci :** Lalu Lintas, kepadatan kendaraan, Mikrokontroler

### **ABSTRACT**

In several cities in Indonesia, traffic congestion is one of the issues that should be addressed. One of the solutions that can be applied to reduce the traffic congestion is making traffic light management system that is able to regulate the flow of traffic at the kelacaran intersection. Along with the advances in technology, found a variety of traffic-light setting support method. One of the methods that are being developed at this time is the use of a webcam camera. By implementing the webcam camera as a detector the density of vehicles on highways is expected to flame the traffic light can adjust the level of the vehicle density and calculating the timing of traffic lights automatically and in real time. Traffic lights system settings with this webcam camera work based on the input in the form of the result image capture. Then the images are processed from RGB to grayscale format, thresholding and histogram. This process committed using personal computer next to this data density results dikirim mikrokontroler atmega16 through serial port. Of average percentage haril testing got that congestion dipagi and afternoon is 30 % and 28 % means congestion are deserted condition. And testing done in the afternoon and night got that congestion average percentage is 53 % and 58 % mean condition of congestion in the afternoon and night is normal.

**Keywords :** Traffic, congestion, Microclontroler

---

### **PENDAHULUAN**

Lalu lintas dapat menjadi barometer kemajuan dari suatu daerah atau kota yang volume lalu lintas tinggi. Lalu lintas lancar dan teratur dapat menunjukkan bahwa disiplin

berlalu lintas dari penduduknya juga tinggi yang berarti pembangunan pada daerah tersebut berkembang secara baik. Semakin meningkatnya perekonomian penduduk sehingga mampu untuk memiliki kendaraan pribadi mengakibatkan ramainya lalu lintas pada

daerah yang memiliki kepadatan penduduk tinggi. Dengan demikian kemacetan lalu lintas juga meningkat.

Lalu lintas memegang peranan yang cukup penting dalam stabilitas sosial, pembangunan, dan peningkatan kehidupan masyarakat. Dengan semakin meningkatnya volume lalu lintas pada suatu persimpangan jalan serta pengaturan nyala lampu lalu lintas yang sudah tidak sesuai lagi dengan tingkat kepadatan kendaraan. Maka menimbulkan berbagai permasalahan yang dapat mengganggu kelancaran lalu lintas dan aktifitas masyarakat. Salah satu faktor yang menyebabkan terhambatnya lalu lintas adalah kemacetan. Di beberapa kota di Indonesia, kemacetan lalu lintas merupakan salah satu masalah yang harus segera diatasi. Salah satu solusi yang dapat diterapkan untuk mengurangi kemacetan lalu lintas adalah dengan membuat sistem pengaturan lampu lalu lintas yang mampu mengatur kelacaran arus lalu lintas di persimpangan namun tetap memperhitungkan biaya, keselamatan dan dampak terhadap lingkungan.

Saat ini sudah cukup banyak kamera yang dipasang di persimpangan-persimpangan jalan. Akan tetapi kamera tersebut hanya sebatas memantau keadaan (*monitoring*) arus lalu lintas dan tidak memberikan kontribusi secara langsung terhadap sistem pengaturan lampu lalu lintas di persimpangan-persimpangan jalan tersebut.

Pemakaian kamera *webcam* sebagai monitoring kondisi arus lalu lintas jalan raya juga bisa diaplikasikan sebagai *detector* kepadatan kendaraan di jalan raya. Oleh karena itu penelitian tugas akhir ini bertujuan mengimplementasikan kamera *webcam* sebagai sensor kepadatan untuk mengatur nyala lampu lalu lintas.

Dari uraian latar belakang di atas, maka dapat diambil sebuah rumusan masalah yaitu "Bagaimana merancang simulasi pengaturan lampu lalu lintas berdasarkan data *image processing* kepadatan kendaraan berbasis mikrokontroler ATMEGA16?".

Pada penelitian ini penulis hanya membahas masalah sebagai berikut :

- a. Sistem diterapkan pada persimpangan sebidang 4 jalur jalan.
- b. Lampu pengatur lalu lintas yang digunakan adalah lampu lalu lintas dengan tiga warna (merah, kuning, hijau).
- c. Kamera akan mendeteksi tingkat kepadatan kendaraan yang berada di dalam area yang telah ditentukan sedangkan kendaraan yang berada di luar area diabaikan.

- d. Kamera hanya mendeteksi kendaraan yang datang ke arah persimpangan lalu lintas, sedangkan kendaraan yang meninggalkan persimpangan lalu lintas diabaikan.
- e. Arah tujuan dari kendaraan diasumsikan bebas (belok kanan, belok kiri, ataupun lurus).
- f. Tingkat kepadatan kendaraan adalah sepi, normal dan padat.
- g. Tinggi kamera yang dipasang pada papan simulasi adalah 16 cm dari permukaan papan.

Tujuan penulis mengangkat judul tersebut adalah sebagai berikut :

- a. Mengatur waktu pergantian nyala lampu lalu lintas pada masing-masing ruas sesuai dengan kondisi kepadatan kendaraan secara real time.
- b. Mengimplementasikan *webcam* sebagai sensor kepadatan dalam sistem pengaturan lampu lalu lintas.

## KAJIAN TEORI

### 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 (Diktat Kuliah. *Rekayasa Lalu Lintas*. Hal 1). Pengaturan arus lalu lintas pada persimpangan pada dasarnya dimaksudkan untuk bagaimana pergerakan kendaraan pada masing-masing kelompok pergerakan kendaraan dapat bergerak secara pergantian sehingga tidak saling mengganggu antar arus yang ada. Ada berbagai jenis kendali dengan menggunakan lampu lalu lintas di mana pertimbangan ini sangat tergantung pada situasi dan kondisi persimpangan yang ada seperti volume, geometrik simpang dan sebagainya.

Berdasarkan cara pengoperasiannya, jenis kendali lampu lalu lintas pada persimpangan dibedakan antar lain: Fixed time traffic signals

Simpang dengan pengaturan waktu lampu lalu lintas tetap (*Fixed Time Operation*) dalam pengoperasiannya menggunakan waktu siklus dan panjang fase yang diatur terlebih dahulu dan dipertahankan untuk suatu periode tertentu. Panjang siklus dan fase adalah tetap selama interval tertentu, sehingga tipe ini merupakan bentuk pengendalian lampu lalu lintas yang paling murah dan sederhana.

Pada keadaan tertentu, tipe ini tidak efisien dibandingkan tipe aktual karena tidak memperhatikan perubahan-perubahan yang terjadi pada volume arus lalu lintas. Sehingga untuk kebutuhan pengendalian dimana lebih baik jika dipakai lebih dari satu pengaturan (*multi-setting*) untuk situasi yang berbeda dalam satu hari. Pada umumnya periode waktu

berhubungan dengan waktu sibuk dalam satu hari yaitu pagi, siang hari dan sore hari.

**Actuated traffic signals**

Sistem ini mengatur waktu siklus dan panjang fase secara berkelanjutan disesuaikan dengan kedatangan arus lalu lintas setiap saat. Kemudian ditentukan nilai waktu hijau maksimum dan minimum. Alat detektor dipasang disetiap cabang simpang untuk mendeteksi kendaraan yang lewat, kemudian data disimpan dalam memori lalu diolah untuk mendapatkan nilai tambah waktu diatas nilai waktu hijau minimum untuk suatu cabang simpang. Oleh karena itu sistem pengaturan ini sangat peka terhadap situasi dan sangat efektif jika diterapkan meminimumkan tundaan pada simpang tersebut.

Terdapat dua jenis *traffic actuated operation*, yaitu *semi actuated operation* dan *fully actuated operation*. Operasional waktu sinyal separuh nyata (*semi actuated operation*) ditetapkan pada simpang dimana arus lalu lintas pada jalan utama jauh lebih besar daripada jalan yang lebih kecil. Sebuah alat deteksi dipasang di jalan minor untuk mengetahui kedatangan kendaraan dari jalan tersebut, dan diatur sedemikian rupa sehingga jalan mayor selalu mendapat sinyal lampu hijau lebih lama.

Operasional waktu sinyal yang nyata *fully actuated operation* ditempatkan pada simpang dimana arus lalu lintas relatif sama disetiap cabang simpang tetapi distribusinya bervariasi dan berfluktuasi. Detektor ditempatkan disetiap cabang simpang. Pada simpang *fully actuated operation* ini untuk tiap-tiap cabang simpang ditentukan waktu hijau maksimum dan minimumnya

Arus lalu lintas yang memasuki suatu simpang akan bervariasi dari waktu ke waktu selama satu hari, sehingga akan dibutuhkan waktu siklus yang bervariasi. Kondisi ini tidak menjadi masalah bagi sistem pengaturan *traffic actuated operation*, sedangkan untuk pengaturan lampu lalu lintas waktu tetap perlu ditentukan waktu siklus yang dapat menghindari terjadinya tundaan yang berlebihan pada suatu arus lalu lintas tinggi.

**Image Processing**

*Image Processing* adalah suatu metode yang digunakan untuk memproses atau memanipulasi gambar dalam bentuk 2 dimensi (Ahmad, Usman : 4). *Image processing* dilakukan untuk memperbaiki kesalahan data sinyal gambar yang terjadi akibat transmisi dan selama akuisisi sinyal, serta untuk meningkatkan kualitas penampakan gambar agar lebih mudah diinterpretasi oleh sistem penglihatan manusia baik dengan melakukan manipulasi dan juga

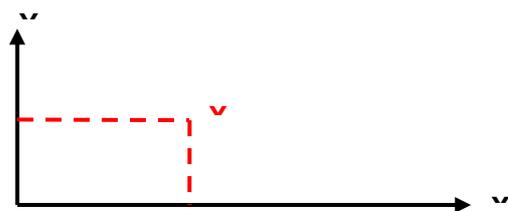
penganalisisan terhadap gambar. *Image processing* dapat juga dikatakan sebagai segala operasi untuk memperbaiki, menganalisa, atau mengubah suatu gambar.

**Citra Digital**

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. (Sutoyo: 9). Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal – sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan.

Citra dapat dikelompokkan menjadi dua bagian yaitu citra diam yaitu citra tunggal yang tidak bergerak dan citra bergerak yaitu rangkaian citra diam yang ditampilkan secara beruntun, sehingga memberi kesan pada mata sebagai gambar yang bergerak. Setiap citra didalam rangkaian itu disebut frame. Gambar-gambar yang tampak pada film layar lebar atau televisi yaitu terdiri dari ribuan sampai ratusan-ribu frame.

Citra digital dapat didefinisikan sebagai fungsi dua variabel,  $f(x,y)$ , dimana  $x$  dan  $y$  adalah koordinat spasial dan nilai  $f(x,y)$  adalah intensitas citra pada koordinat tersebut, hal tersebut diilustrasikan pada gambar 2.1. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red, Green, Blue - RGB*).



**Gambar 2.1** Koordinat (x,y) pada citra digital (Teori Pengolahan Citra Digital, 2005:10)

Berdasarkan Gambar 2.1, secara matematis citra digital dapat dituliskan sebagai fungsi intensitas  $f(x,y)$ , di mana harga  $x$  (baris) dan  $y$  (kolom) merupakan koordinat posisi dan  $f(x,y)$  adalah nilai fungsi pada setiap titik  $(x,y)$  yang menyatakan besar intensitas citra atau warna dari piksel di titik tersebut.

**Citra Grayscale**

Proses awal yang banyak dilakukan dalam *image processing* adalah mengubah citra berwarna menjadi citra grayscale. Hal ini digunakan untuk menyerderhanakan model

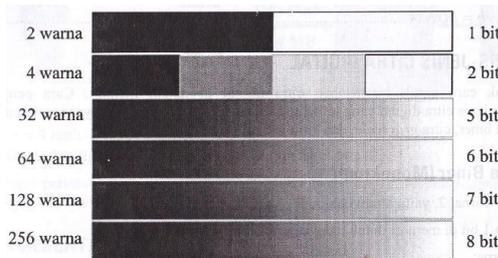
citra. Citra *grayscale* adalah citra yang hanya menggunakan warna pada tingkatan warna abu-abu. (Sutoyo: 12). Warna abu-abu adalah satu-satunya warna pada ruang RGB dengan komponen merah, hijau dan biru mempunyai intensitas yang sama. Pada citra beraras keabuan hanya perlu menyatakan nilai intensitas untuk tiap piksel sebagai nilai tunggal, sedangkan pada citra berwarna perlu tiga nilai intensitas untuk tiap pikselnya. Intensitas dari citra grayscale disimpan dalam 8 bit integer yang memberikan 256 kemungkinan yang mana dimulai dari level 0 sampai dengan 255 (0 untuk hitam dan 255 untuk putih dan diantaranya adalah derajat keabuan).



**Gambar 2.2** Grayscale level

(Teori Pengolahan Citra Digital, 2005: 9).

Semakin besar warna yang disediakan di memori, semakin halus gradasi warna yang terbentuk. Gambar 2.4 menunjukkan perbandingan warna untuk bit tertentu.



**Gambar 2.3** Perbandingan gradasi warna

(Teori Pengolahan Citra Digital, 2005: 9).

Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = *Red, Green, Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak  $2^8 \cdot 2^8 \cdot 2^8 = 2^{24} = 16$  juta warna lebih. Itulah sebabnya format ini dinamakan *true color* karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna di alam.

Penyimpanan *true color* di dalam memori berbeda dengan citra *grayscale*. Setiap piksel dari citra *grayscale* 256 gradasi warna diwakili oleh 1 *byte*. Sedangkan 1 piksel citra *true color* diwakili oleh 3 *byte*, dimana masing-masing *byte* merepresentasikan warna merah (*red*), hijau (*green*), dan biru (*blue*).

Citra warna bisa diubah menjadi citra *grayscale* dengan cara menghitung rata-rata

elemen warna *red, green, blue*. Secara matematis penghitungannya adalah sebagai berikut :

$$f_o(x, y) = \frac{f_i^R(x, y) + f_i^G(x, y) + f_i^B(x, y)}{3} \dots \dots \dots (2.2)$$

Keterangan :

$f_o(x, y)$  = derajat keabuan o di titik (x,y)

$f_i^R(x, y)$  = Nilai komponen warna merah pada derajat keabuan i di titik (x,y)

$f_i^G(x, y)$  = Nilai komponen warna hijau pada derajat keabuan i di titik (x,y)

$f_i^B(x, y)$  = Nilai komponen warna biru pada derajat keabuan i di titik (x,y)

Contoh perhitungan digital:

R = 50	R = 40	R = 90
G = 65	G = 40	G = 90
B = 50	B = 55	B = 90
R = 40	R = 50	R = 40
G = 80	G = 80	G = 90
B = 30	B = 50	B = 80
R = 80	R = 70	R = 80
G = 60	G = 70	G = 90
B = 40	B = 70	B = 70
R = 50	R = 40	R = 70
G = 90	G = 60	G = 70
B = 70	B = 50	B = 70

Citra warna

$$f_o = \frac{(50 + 65 + 50)}{3}$$

55	45	90
50	60	70
60	70	80
70	50	70

Citra grayscale

Diketahui citra warna 8 bit dengan ukuran 4 x 2 piksel akan diubah menjadi *grayscale*. Perhitungan fungsi negasi dilakukan sebagai berikut. Setiap titik yang terletak di posisi (x,y), nilai-nilai komponen red, green dan blue ditambahkan, kemudian hasilnya dibagi 3.

> 128, diubah menjadi 255  
< 128, diubah menjadi 0

**Thresholding**

Pada operasi ini nilai piksel yang memenuhi syarat ambang batas dipetakan ke suatu nilai yang dikehendaki. Dalam hal ini syarat ambang batas dan nilai yang dikehendaki disesuaikan dengan kebutuhan. Operasi *thresholding* yang mempunyai ketentuan berikut: nilai intensitas output  $f_o(x, y) = 0$ , bila nilai intensitas input  $f_i(x, y) \leq 0$ , nilai  $f_o(x, y) = T_1$  bila  $T_1 < f_i(x, y) \leq T_2$ , nilai  $f_o(x, y) = T_2$  bila  $T_2 < f_i(x, y) \leq T_3$ , ..., nilai  $f_o(x, y) = T_{n-1}$  bila  $T_{n-1} < f_i(x, y) < T_n$  secara matematis dapat ditulis sebagai berikut:

$$f_o(x, y) = \left\{ \begin{array}{l} T_1, f_i(x, y) \leq T_1 \\ T_2, T_1 < f_i(x, y) \leq T_2 \\ T_3, T_2 < f_i(x, y) \leq T_3 \\ \vdots \\ T_n, T_{n-1} < f_i(x, y) \leq T_n \end{array} \right\} \dots \dots \dots (2.3)$$

Keterangan :  
 $f_0(x, y)$  = piksel citra baru (hasil/output)  
 $f_i(x, y)$  = citra asli (input)  
 $T_1, T_2, T_3, \dots, T_n$  = nilai-nilai threshold yang dikehendaki.

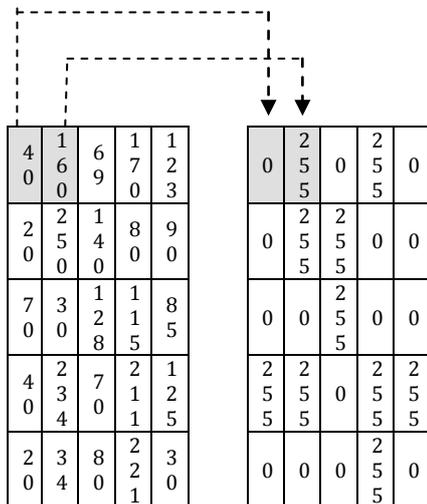
Contoh perhitungan digital:  
 Misalkan diketahui citra grayscale 256 warna dengan ukuran 5 x 5 piksel,

4	16	69	17	12
0	0	0	0	3
2	25	14	80	90
0	0	0	0	0
7	30	12	11	85
0	30	8	5	0
4	23	70	21	12
0	4	0	1	5
2	34	80	22	30
0	0	0	1	0

Akan dilakukan operasi ambang tunggal dengan fungsi berikut

$$f_0(x, y) = \begin{cases} 0, & f_i(x, y) < 128 \\ 255, & f_i(x, y) \geq 128 \end{cases} \dots \dots \dots (2.4)$$

Maka perhitungan digital yang dilakukan adalah sebagai berikut : untuk setiap nilai intensitas citra asli yang nilainya < 128, diubah menjadi 0, sedangkan setiap nilai intensitas citra asli yang nilainya  $\geq 128$  diubah menjadi 255.



**Histogram**

Histogram adalah grafik yang menunjukkan frekuensi kemunculan setiap nilai gradasi warna (Sutoyo : 29). Histogram juga dapat menunjukkan banyak kecerahan (*brightness*) dan kontras (*contrast*) dari sebuah gambar.

Misalkan citra digital memiliki  $L$  derajat keabuan, yaitu dari nilai 0 sampai  $L - 1$  (misalnya pada citra dengan kuantisasi derajat

keabuan 8-bit, nilai derajat keabuan dari 0 sampai 255).

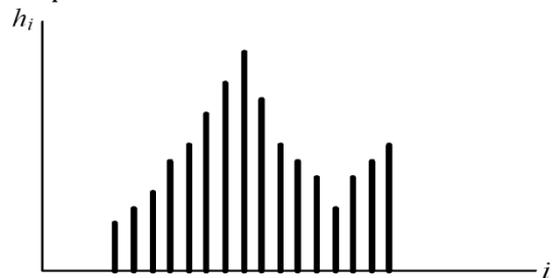
Secara matematis histogram citra dihitung dengan rumus

$$h_i = \frac{n_i}{n}, i = 0, 1, \dots, L - 1 \dots \dots \dots (2.5)$$

Keterangan :

$h_i$  = nilai histogram  
 $n_i$  = jumlah piksel yang memiliki derajat keabuan  $i$   
 $n$  = jumlah seluruh piksel di dalam citra

Bila digambarkan pada koordinat kartesian maka sumbu X (absis) menunjukkan tingkat warna sumbu Y (ordinat) menunjukkan frekuensi kemunculan. Gambar 2.4 adalah contoh sebuah histogram citra. Secara grafis histogram ditampilkan dengan diagram batang. Perhatikan dari persamaan 2.4 bahwa nilai  $n_i$  telah dinormalkan dengan membaginya dengan  $n$ . Nilai  $h_i$  berada di dalam selang 0 sampai 1.



**Gambar 2.4** Histogram citra (Teori Pengolahan Citra Digital, 2005: 29).

Puncak histogram menunjukkan intensitas *pixel* yang menonjol. Lebar dari puncak menunjukkan rentang kontras dari gambar. Citra yang mempunyai kontras yang terlalu terang (*overexposed*) atau terlalu gelap (*underexposed*) memiliki histogram yang sempit. Histogramnya terlihat hanya menggunakan setengah dari daerah derajat keabuan. Citra yang baik akan memiliki histogram yang mengisi daerah derajat keabuan secara penuh dengan distribusi yang merata pada setiap nilai intensitas *pixel*.

Kita ambil contoh citra greyscale ukuran 5x5 piksel dengan peta berikut.

1	3	0	2	1
4	4	5	7	6
1	6	6	7	0
7	4	1	2	1
2	0	0	7	4

Kita bisa lihat pada gambar di atas, nilai 1 muncul 4 kali, nilai 2 muncul 3 kali, nilai 3 muncul 3 kali dan seterusnya. Tabulasi perhitungan histogramnya ditunjukkan pada di bawah ini, bahwa semakin besar nilai  $n$  maka semakin besar pula nilai  $h_i$ .

**Tabel 2.1** Perhitungan histogram citra

Intensitas warna (x)	0	1	2	3	4	5	6	7
Jumlah (y)	4	5	3	1	4	1	3	4
$y/n \rightarrow N = 25$	0,6	0,2	0,2	0,4	0,6	0,4	0,2	0,6

Keterangan :

X = intensitas warna

Y = jumlah kemunculan intensitas warna pada citra digital

N = total kemunculan intensitas warna pada citra digital

**Mikrokontroler ATmega 16**

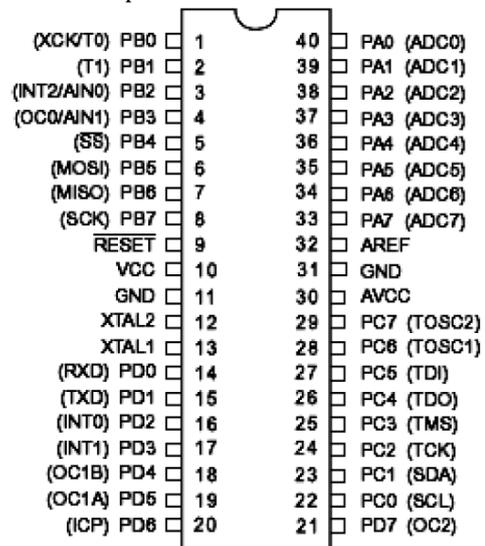
Mikrokontroler AVR adalah mikrokontroler RISC 8 bit berdasarkan arsitektur *Harvard*, yang dibuat oleh Atmel pada tahun 1996 (Andrianto, Heri : 32). AVR mempunyai kepanjangan *Advanced Versatile RISC* atau *Alf and Begard'd Risc Processor*. AVR memiliki keunggulan dibandingkan dengan mikrokontroler lain, keunggulan AVR yaitu AVR memiliki kecepatan eksekusi program yang lebih cepat, karena sebagian besar instruksi dieksekusi dalam 1 siklus *clock*, lebih cepat dibandingkan MCS51 yang membutuhkan 12 siklus *clock* untuk mengeksekusi 1 instruksi. Mikrokontroler ATmega16 memiliki fitur yang lengkap (ADC internal, EEPROM internal, *Timer/Counter*, *Watchdog Timer*, PWM, Port I/O, komunikasi serial, Komparator, I2C,dll).

Fitur-fitur yang dimiliki ATmega16 sebagai berikut:

1. Arsitektur RISC dengan *throughput* mencapai 16 MIPS pada frekuensi 16 Mhz.
2. Memiliki kapasitas *flash* memori 16Kbyte, EEPROM 512 Byte, dan SRAM 1Kbyte
3. Saluran Port I/O sebanyak 32 buah, yaitu Port A, Port B, Port C, dan Port D.
4. CPU yang terdiri atas 32 buah register
5. User interupsi internal dan eksternal
6. Port USART sebagai komunikasi serial
7. Konsumsi daya rendah (DC 5V)
8. Fitur *peripheral*, yang terdiri dari

- a. Tiga buah *Timer/Counter* dengan perbandingan: 2 (dua) buah *Timer/Counter* 8 bit dengan *Prescaler* terpisah dan *Mode Compare*, 1 (satu) buah *Timer/Counter* 16 bit dengan *Prescaler* terpisah, *Mode Compare*, dan *Mode Capture*
- b. *Real Time Counter* dengan osilator tersendiri
- c. 4 channel PWM
- d. 8 channel, 10-bit ADC, 8 *Single-ended Channel*, *Differential Channel* hanya pada kemasan TQFP, *Differential Channel* dengan *Programmable Gain* 1x, 10x, atau 200x
- e. *Byte-oriented Two-wire Serial Interface*
- f. Antarmuka SPI
- g. *Watchdog Timer* dengan osilator internal
- h. *On-chip Analog Comparator*

Susunan pin mikrokontroler ATmega16 diperlihatkan pada Gambar di bawah ini



**Gambar 2.5** Konfigurasi pin ATmega16 (Pemrograman mikrokontroler AVR ATmega16 Menggunakan Bahasa C (Code Vision AVR), 2008: 8).

Konfigurasi *pin* ATMEGA16 dengan kemasan 40 *pin Dual In-line Package* (DIP) dapat dilihat pada Gambar 2.7. Dari gambar diatas dapat dijelaskan fungsi dari masing-masing *pin* ATMEGA16 sebagai berikut.

- 1) VCC merupakan *pin* yang berfungsi sebagai masukan catu daya
- 2) GND merupakan *pin* Ground
- 3) Port A (PA0 – PA7) merupakan *pin input/output* dua arah (*full duplex*) dan selain itu merupakan *pin* masukan ADC
- 4) Port B (PB0 – PB7) merupakan *pin input/output* dua arah (*full duplex*) dan selain itu merupakan *pin* khusus, seperti dapat dilihat pada tabel dibawah ini.

**Tabel 2.2** Fungsi Khusus Port B

Pin	Fungsi Khusus
-----	---------------

PB0	XCK (USART External Clock Input/Output)to (Timer/Counter External Counter Input)
PB1	T1 (Timer/Counter1 External Counter Input)
PB2	INT2 (External Interupt 2 Input) AIN0 (Analaog Comparator Negative Input)
PB3	OC0 (Timer/Counter0 Output Compare Macth Output) AIN1 (Analaog Comparator Negative Input)
PB4	(SPI Slave Select Input)
PB5	MOSI (SPI Bus Master Output /Slave Input)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB7	SCK (SPI Bus Serial Clock)

- 5) Port A (PC0 – PC7) merupakan pin input/output dua arah (full duplex) dan selain itu merupakan pin khusus, seperti dapat dilihat pada tabel dibawah ini.

**Tabel 2.3** Fungsi Khusus Port C

Pin	Fungsi Khusus
PC0	SCL (Two-wire Serial Bus Clock Line)
PC1	SDA (Two-wire Serial BusData Input/Output Line)
PC2	TCK (Joint Test Action Group Test Clock)
PC3	TMS (JTAG Test Mode Select)
PC4	TDO (JTAG Data Out)
PC5	TDI (JTAG Test Data In)
PC6	TOSC1 (Timer Oscillator pin 1)
PC7	TOSC2 (Timer Oscillator pin 2)

- 6) Port D (PD0 – PD7) merupakan pin input/output dua arah (full duplex) dan selain itu merupakan pin khusus, seperti dapat dilihat pada tabel dibawah ini.

**Tabel 2.4** Fungsi Khusus Port D

Pin	Fungsi Khusus
PD0	RXD (USART Input Pin)
PD1	TXD (USART Output Pin)
PD2	INT0 (External Interupt 0 Input)
PD3	INT1 (External Interupt 1 Input)
PD4	OC1B (Timer/Counter1 Output Compare B Macth Output)
PD5	OC1A (Timer/Counter1 Output

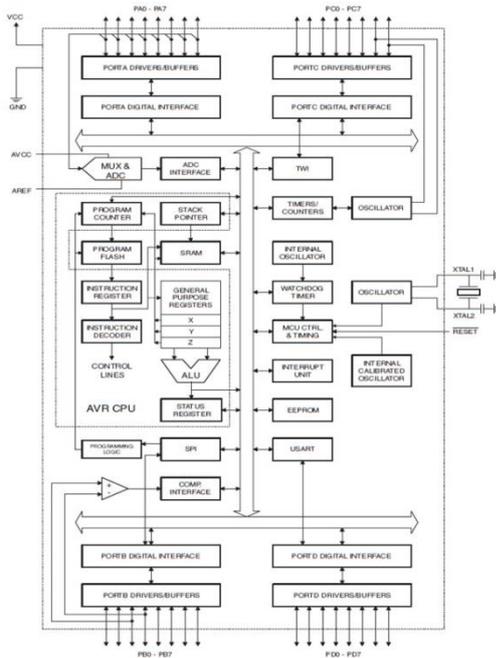
Pin	Fungsi Khusus
	Compare A Macth Output)
PD6	ICP (Timer/Counter1 Input Capture Pin)
PD7	OC2 (Timer/Counter2 Output Compare Macth Output)

- 7) RESET merupakan pin yang digunakan untuk me-reset mikrokontroler  
 8) XTAL1 dan XTAL2, merupakan pin masukan external clock  
 9) AVCC merupakan pin masukan tegangan untuk ADC  
 10) AREF merupakan pin masukan tegangan referensi untuk ADC.

**Arsitektur Mikrokontroler ATmega16**

Mikrokontroler ATmega 16 dibangun berdasarkan arsitektur seperti ditunjukkan gambar 2.6 dibawah ini. Seluruh bagian yang digambar pada gambar tersebut saling berhubungan melalui internal bus 8 bit menelusuri bagian serpih. Bus tersebut kemudian dihubungkan ke luar melalui input output port apabila memori atau ekspansi diperlukan.

Unit pengolah pusat (CPU) terdiri atas dua bagian, yaitu unit pengendali control unit (CU), serta unit aritmatika dan logika (ALU). Fungsi utama unit pengendali ini adalah mengambil, mengkode, dan melaksanakan urutan intruksi sebuah program yang tersimpan dalam memori, unit pengendali juga berfungsi untuk mengatur urutan operasi seluruh sistem. Unit pengendali atau CPU juga menghasilkan dan mengatur sinyal pengendali yang diperlukan untuk menyerempakkan operasi, juga aliran intruksi program. Aliran informasi pada bus-bus data dan bus alamat juga diatur oleh unit ini.



**Gambar 2.6** Arsitektur Mikrokontroler ATmega16

(Pemrograman mikrokontroler AVR ATmega16 Menggunakan Bahasa C (Code Vision AVR), 2008: 11).

software, dan pembuatan rangkaian elektronika. Ketiga aspek tersebut harus saling berkaitan satu sama lain terhadap kinerja simulasi pengaturan lampu lalu lintas.

**Perancangan dan Pembuatan Alat Perancangan Perangkat Keras**

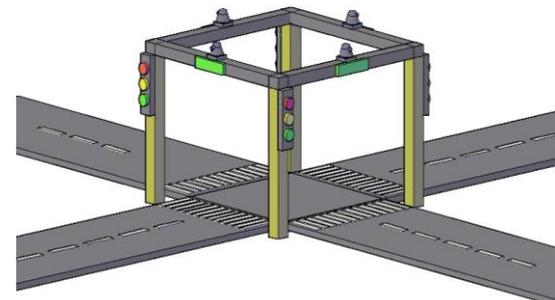
Desain perangkat keras pada tugas akhir ini merupakan salah satu aspek yang sangat penting karena tanpa perangkat keras sistem tidak dapat berjalan dengan baik. Perangkat keras dapat didefinisikan sebagai perangkat yang terdiri dari kerangka (papan simulasi) atau komponen elektronika yang bersifat dapat dilihat dan diraba secara langsung dan berbentuk nyata dalam bentuk fisik untuk mendukung proses suatu sistem yang akan dijalankan. Adapun bagian-bagian perangkat keras pada simulasi pengaturan lampu lalu lintas berdasarkan data image kepadatan kendaraan meliputi perancangan papan simulasi dan perancangan elektronika.

**Perancangan Papan Simulasi**

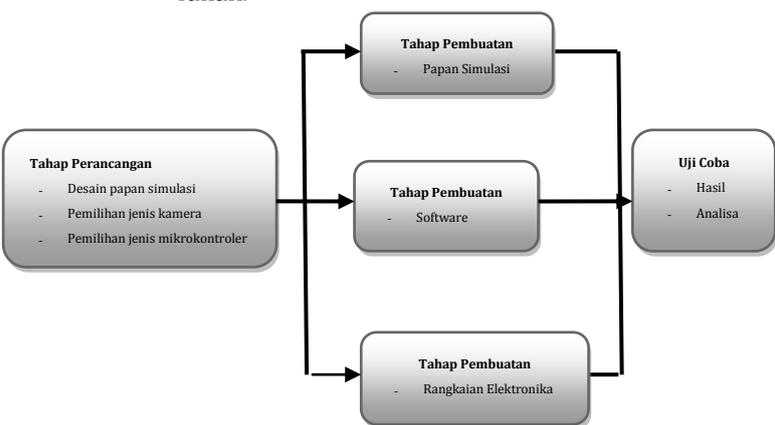
Pada proyek tugas akhir ini, papan simulasi pengaturan lampu lalu lintas dirancang menggunakan papan kayu berukuran 160 cm<sup>2</sup> dan untuk tiang-tiang lampu lalu lintas digunakan aluminium dengan ketinggian 15 cm.

**PEMBAHASAN**

Pada bab ini akan dijelaskan mengenai desain sistem lalu lintas yang dikembangkan pada tugas akhir yang meliputi uraian singkat cara kerja sistem, desain perangkat keras yang terdiri dari desain papan simulasi, desain rangkaian elektronika dan desain perangkat lunak.

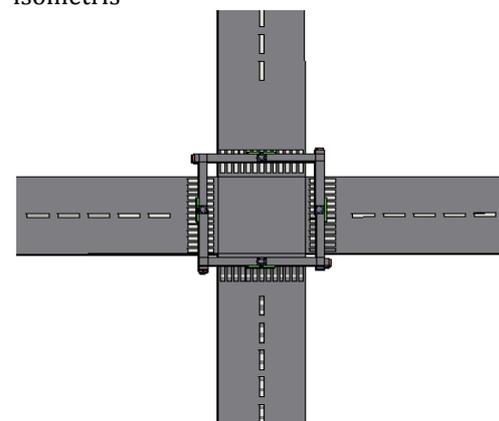


**Gambar 3.3** Kontruksi papan simulai tampak isometris



**Gambar 3.1** TahapPembuatan Simulasi Pengaturan Lampu Lalu Lintas

Berdasarkan gambar di atas dapat dijelaskan bahwa ada 3 tahapan untuk membuat simulasi pengaturan lampu lalu lintas, yaitu : pembuatan papan simulasi, pembuatan

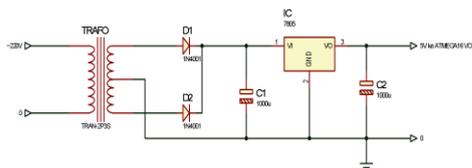


**Gambar 3.4** Kontruksi papan simulai tampak atas

**Perancangan Elektronika Rangkaian Power Supply**

Pada dasarnya prinsip kerja power supply adalah mengubah tegangan bolak-balik (AC) menjadi tegangan searah (DC). Sesuai dengan rangkaian catu daya yang akan dibuat, ketika memasukkan tegangan bolak-balik pada input transformator, maka transformator tersebut akan memindahkan tegangan pada bagian sekunder trafo dan membaginya ke dalam bentuk tegangan yang bernilai kecil.

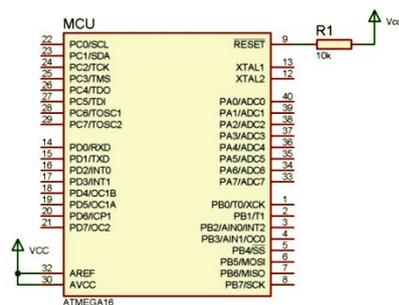
Nilai tersebut sesuai dengan jumlah lilitan yang ada di dalam lilitan sekunder, misalnya masukan tegangan AC 220 volt kemudian keluar pada bagian sekunder trafo 5 V AC. Tegangan bolak-balik pada bagian sekunder trafo kemudian disearahkan menggunakan komponen diode 1N4001. Penyearahan hasil dari diode ini masih belum sempurna, oleh karena itu maka perlu disaring tegangan tersebut dengan menggunakan kapasitor 1000µF. Agar trasformator stabil maka diperlukan tambahan regulator IC LM 7805 yang digunakan untuk menstabilkan tegangan 5 volt tersebut.



Gambar 3.5 Rangkaian Power Supply

**Rangkaian Sistem Minimum**

Sistem minimum mikrokontroler adalah rangkaian elektronik minimum yang diperlukan untuk beroperasinya IC mikrokontroler. Sistem ini kemudian bisa dihubungkan dengan rangkaian lain untuk menjalankan fungsi tertentu:



Gambar 3.6 Minumum Sistem ATmega16

Adapun untuk konfigurasi pin I/O bisa dilihat pada tabel berikut :

Tabel 3.1 Konfigurasi Pin I/O yang digunakan pada ATmega16

Nama Port	Pin No	Pin I/O	Fungsi
PORT A	40	0	Driver katode 7 segment
	39	1	Driver katode 7

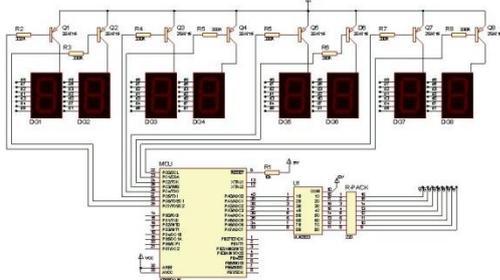
Nama Port	Pin No	Pin I/O	Fungsi	
			segment	
		38	2	Driver katode 7 segment
		37	3	Driver katode 7 segment
		36	4	Driver katode 7 segment
		35	5	Driver katode 7 segment
		34	6	Driver katode 7 segment
		33	7	Driver katode 7 segment
PORT B	1	0	Driver LED merah lajur 4	
	2	1	Driver LED hijau lajur 3	
	3	2	Driver LED merah lajur 3	
	4	3	Driver LED kuning lajur 3	
	5	4	Driver LED hijau lajur 2	
	6	5	Driver LED kuning lajur 4	
	7	6	Driver LED hijau lajur 4	
	8	7	Driver LED kuning lajur 1	
PORT C	22	0	Driver anode 7 segment lajur 1	
	23	1	Driver anode 7 segment lajur 2	
	24	2	Driver anode 7 segment lajur 2	
	25	3	Driver anode 7 segment lajur 3	
	26	4	Driver anode 7 segment lajur 3	
	27	5	Driver anode 7 segment lajur 4	
	28	6	Driver anode 7 segment lajur 4	
PORT D	29	7	Driver anode 7 segment lajur 1	
	14	0	Penerima komunikasi data serial	
	15	1	Pengirim komunikasi data serial	
	16	2	Driver LED kuning lajur 2	
	17	3	Driver LED merah lajur 2	

Nama Port	Pin No	Pin I/O	Fungsi
	18	4	Driver LED hijau jalur 1
	19	5	Driver LED kuning jalur 1
	20	6	Driver LED merah jalur 1
	21	7	Driver anode 7 segment lajur 1

**Rangkaian Seven Segment**

Seven segment merupakan penampil yang setingkat dengan led, hanya saja seven segment dapat menampilkan data ke bentuk karakter. Seven segment yang nyala tergantung pada output dari Port A.0 – Port A.7 dan Port C.0 – Port C.7. Pada gambar seven segment common adonda dikendalikan dengan menggunakan transistor PNP melalui port C.0 – port C.7. Apabila ada logika low “0” pada basis transistor, maka seven segment akan menyala dan sebaliknya akan padam.

Supaya arus keluaran mikrokontroler mampu menyalakan 8 seven segment bersamaan perlu menggunakan penguat arus. Penguat arus yang digunakan dalam sistem ini adalah berupa IC ULN seri 2803 untuk driver seven segment. IC ini dipilih karena bisa digunakan secara langsung dengan mikrokontroler. IC ini menggunakan rangkaian darlington sebagai penguat arus yang penguatannya mampu mencapai 500 mA

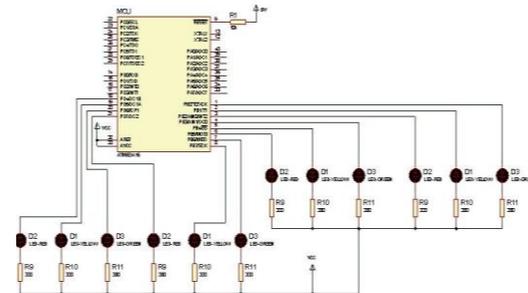


Gambar 3.7 Rangkaian Seven Segment

**Rangkaian 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 tugas akhir ini mengikuti arah seperti ditunjukkan pada Lampu lalu lintas bekerja secara bergantian pada tiap jalur sesuai dengan tingkat kepadatan di setiap lajur jalan dengan urutan menyala lampu hijau, lampu kuning dan lampu merah.

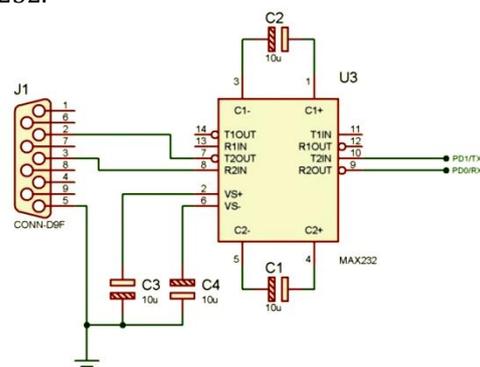
Untuk lampu lalu lintas digunakan LED untuk menampilkan simulasi nyala lampu lalu lintas, dimana warna LED yang digunakan adalah merah, kuning dan hijau, sebanyak 12 LED. Port mikrokontroler yang digunakan adalah Port D.4 – Port D.7 dan Port B.0 – Port B.7 dan digunakan resistor 330 ohm sebagai pembatas arusnya. Tegangan yang digunakan sebesar 5 V.



Gambar 3.8 Rangkaian Lampu Lalu Lintas

**Rangkaian Interface**

Hubungan hardware dengan PC adalah melalui serial port yang mengadopsi standar RS-232 yang umumnya tersedia pada setiap PC. Namun standar RS-232 mempunyai level tegangan yang berbeda dengan format data digital TTL. Oleh sebab itu untuk menghubungkan alat dengan PC dibutuhkan rangkaian konverter yang bisa menyesuaikan data TTL ke RS 232 dan sebaliknya. Rangkaian ini digunakan untuk menyelaraskan beda level tegangan RS 232 dengan level tegangan TTL. Berikut adalah gambar rangkaian interface RS 232.

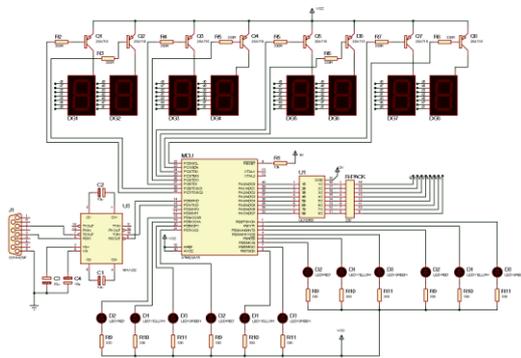


Gambar 3.9 Rangkaian Interface

**Rangkaian Keseluruhan**

Rangkaian keseluruhan yang dimaksud adalah satu kesatuan dari bagian-bagian hardware sistem pengaturan lampu lalu lintas dalam satu rangkaian. Adapun bagian hardware yang dijadikan menjadi satu kesatuan rangkaian adalah rangkaian minimum sistem ATmega16, rangkaian seven segment, rangkaian interface,

dan rangkaian lampu lalu lintas. Adapun gambar rangkainnya sebagai berikut :



Gambar 3.10 Rangkaian Keseluruhan

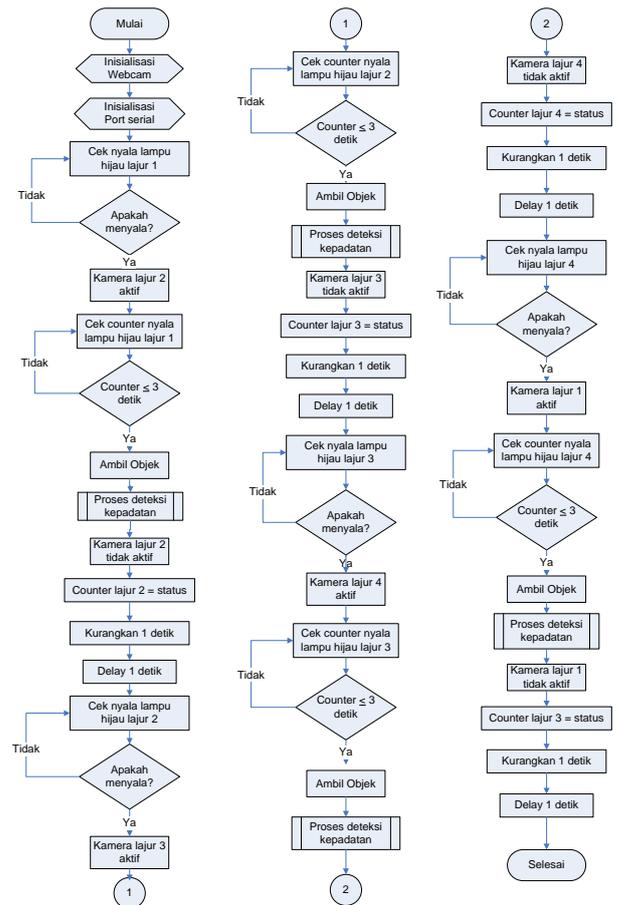
**Perancangan Perangkat Lunak**

Untuk membuat program pada simulasi pengaturan lampu lalu lintas ada langkah-langkah yang harus ditempuh. Salah satu langkah tersebut adalah dengan membuat flowchart atau alur program yang akan diterapkan pada sistem.

Konsepnya adalah bagaimana mengatur nyala lampu lalu lintas berdasarkan kepadatan kendaraan pada tiap lajur jalan. Langkah pertama ialah mengalirkan arus listrik pada modul perangkat keras dari simulasi ini meliputi kamera, *personal computer*, dan pengendali lampu lalu lintas. Selanjutnya ialah meletakkan miniatur kendaraan disetiap lajur sehingga didapatkan kondisi kepadatan awal.

Selanjutnya webcam yang telah dipasang disetiap lajur jalan akan melakukan proses pengambilan gambar objek secara bergantian. Objek gambar yang diproses selanjutnya ialah objek terakhir yang diambil pada saat lampu merah menyala kurang dari atau sama dengan 5 detik pada tiap webcam yang dipasang pada tiap lajur jalan. Gambar ini akan diproses selanjutnya sehingga didapatkan presentase kepadatan kendaraan. Presentase kepadatan kendaraan meliputi normal, padat dan macet.

Berikut ini adalah flowchat pengambilan objek oleh kamera webcam pada tiap lajur jalan.



Gambar 3.11 Flowchat Pengambilan Objek

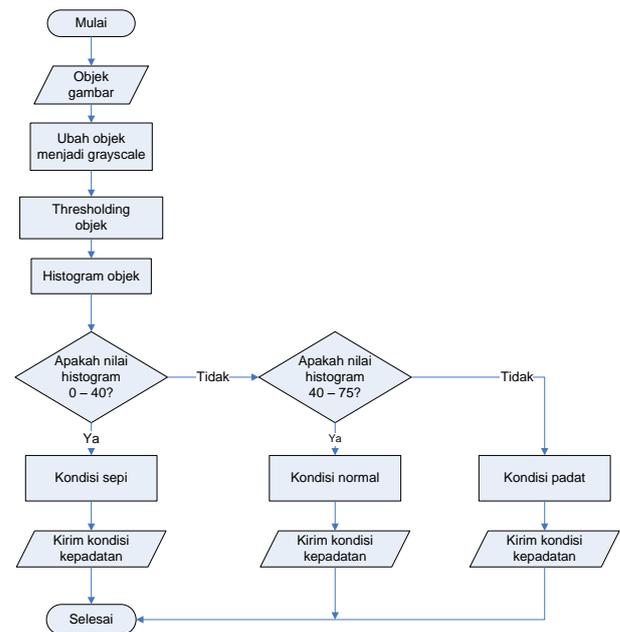
Keterangan :

1. Ketika sistem mulai dioperasikan langkah pertama ialah inisialisasi atau pengenalan 4 buah kamera webcam yang dipasang di setiap lajur jalan
2. Kemudian inisialisasi port serial sebagai media komunikasi antar personal computer dengan mikrokontroler ATmega16.
3. Sistem melakukan pengecekan nyala lampu hijau pada lajur 1.
4. Apabila pada lajur 1 lampu hijau menyala maka sistem mengaktifkan kamera webcam pada lajur 2.
5. Sistem melakukan pengecekan counter nyala lampu hijau pada lajur 1.
6. Webcam pada lajur 2 akan mengambil gambar kendaraan apabila counter nyala lampu hijau pada lajur 1 kurang dari atau sama dengan 3 detik.
7. Sistem melakukan proses deteksi kepadatan kendaraan pada lajur 2 dari gambar terakhir yang dicuplik oleh kamera webcam.
8. Sistem menonaktifkan kamera webcam pada lajur 2.
9. Counter nyala lampu hijau akan berjalan mundur sampai menunjukkan angka 0.

10. Sistem melakukan pengecekan nyala lampu hijau pada lajur 2.
11. Apabila pada lajur 2 lampu hijau menyala maka sistem mengaktifkan kamera webcam pada lajur 3.
12. Sistem melakukan pengecekan counter nyala lampu hijau pada lajur 2.
13. Webcam pada lajur 3 akan mengambil gambar kendaraan apabila counter nyala lampu hijau pada lajur 1 kurang dari atau sama dengan 3 detik.
14. Sistem melakukan proses deteksi kepadatan kendaraan pada lajur 3 dari gambar terakhir yang dicuplik oleh kamera webcam.
15. Sistem menonaktifkan kamera webcam pada lajur 3.
16. Counter nyala lampu hijau akan berjalan mundur sampai menunjukkan angka 0.
17. Sistem melakukan pengecekan nyala lampu hijau pada lajur 3.
18. Apabila pada lajur 3 lampu hijau menyala maka sistem mengaktifkan kamera webcam pada lajur 4.
19. Sistem melakukan pengecekan counter nyala lampu hijau pada lajur 3.
20. Webcam pada lajur 4 akan mengambil gambar kendaraan apabila counter nyala lampu hijau pada lajur 1 kurang dari atau sama dengan 3 detik.
21. Sistem melakukan proses deteksi kepadatan kendaraan pada lajur 4 dari gambar terakhir yang dicuplik oleh kamera webcam.
22. Sistem menonaktifkan kamera webcam pada lajur 4.
23. Counter nyala lampu hijau akan berjalan mundur sampai menunjukkan angka 0.
24. Sistem melakukan pengecekan nyala lampu hijau pada lajur 4.
25. Apabila pada lajur 4 lampu hijau menyala maka sistem mengaktifkan kamera webcam pada lajur 14.
26. Sistem melakukan pengecekan counter nyala lampu hijau pada lajur 4.
27. Webcam pada lajur 1 akan mengambil gambar kendaraan apabila counter nyala lampu hijau pada lajur 1 kurang dari atau sama dengan 3 detik.
28. Sistem melakukan proses deteksi kepadatan kendaraan pada lajur 1 dari gambar terakhir yang dicuplik oleh kamera webcam.
29. Sistem menonaktifkan kamera webcam pada lajur 1.
30. Counter nyala lampu hijau akan berjalan mundur sampai menunjukkan angka 0.
31. Selesai

Gambar 3.12 ialah gambar flowchart untuk proses deteksi kepadatan. Proses ini

dilakukan pada semua objek gambar yang dihasilkan oleh kamera webcam



**Gambar 3.12** Flowchat Proses Deteksi Kepadatan

Keterangan :

1. Object gambar yang diambil oleh webcam masih berupa object berwarna.
2. Object gambar akan dirubah menjadi gambar abu-abu atau citra grayscale.
3. Object akan diberi nilai ambang batas atau threshold.
4. Pada object akan dilakukan histogram.
5. Nilai dari histogram akan dirubah menjadi presentase kepadatan kendaraan.
6. Kondisi kepadatan kendaraan akan dikirm ke mikrokontroler melalui port serial.

Sedangkan gambar 3.13 adalah gambar flowchart untuk pengiriman kondisi kepadatan. Kondisi kepadatan ini dikirimkan oleh personal computer ke mikrokontroler melalui port serial



**Gambar 3.13** Flowchat pengiriman kondisi kepadatan

Keterangan :

1. Ketika sistem mulai dioperasikan langkah pertama ialah inisialisasi atau pengenalan yaitu port serial.
2. Sistem akan menerima inputan berupa kondisi kepadatan kendaraan.
3. Kondisi kepadatan kendaraan ini meliputi normal, padat dan macet
4. Sistem akan melakukan pengaturan lampu lalu lintas berdasarkan inputan yang diterima.

Selanjutnya akan dibahas mengenai pengujian dan analisa berdasarkan perancangan dari sistem yang dibuat. Pengujian ini dilakukan untuk mengetahui kehandalan dari sistem apakah sudah sesuai dengan perancangan atau belum. Pengujian dilakukan secara terpisah dan selanjutnya dilakukan ke dalam sistem yang telah terintegrasi.

Pengujian akan dilakukan meliputi :

1. Pengujian pengambilan objek
2. Pengujian rangkaian seven segment
3. Pengujian komunikasi serial
4. Pengujian sistem secara keseluruhan

**Pengujian sistem secara keseluruhan**

Pengujian keseluruhan merupakan pengujian dengan menggabungkan seluruh bagian dari sistem baik hardware maupun software dan dilakukan berulang kali.

Pengujian dilakukan pada waktu yang berbeda yaitu pagi, siang, sore, dan malam hari di luar ruangan karena pencahayaan dari luar ruangan dengan waktu yang berbeda merupakan salah satu aspek yang mempengaruhi gambar yang dihasilkan oleh kamera webcam.

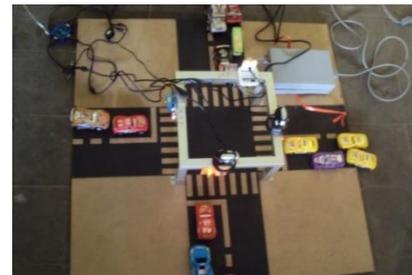
Pada pengujian sistem secara keseluruhan ini, diketahui bahwa pengambilan gambar objek kendaraan dilakukan tiap webcam yang terpasang pada tiang lampu lalu lintas pada saat lampu merah menyala 5 detik terakhir menuju nol. Hasil gambar ini akan diterima oleh PC yang selanjutnya dilakukan proses grayscale, thresholding dan histogram sehingga didapatkan kondisi kepadatan kendaraan, apakah sepi, normal atau padat.

Data kondisi kepadatan kendaraan akan dikirimkan ke mikrokontroler melalui port serial. Mikrokontroler akan mengatur nyala lampu lalu lintas disetiap tiang. Arah pergerakan lampu lalu lintas adalah searah jarum jam.

Berikut gambar tampilan program simulasi pengaturan lampu lalu lintas berdasarkan data image kepadatan kendaraan berbasis mikrokontroler ATmega16



**Gambar 4.22** Tampilan software simulasi pengaturan lampu lalu lintas



**Gambar 4.23** Gambar papan simulasi lalu lintas

**Tabel 4.2** Data Pengujian dilakukan dipagi hari

No	Lajur	Presentase Kepadatan	Kondisi	Waktu Hijau (detik)
1	I	28	sepi	14
	II	15	Sepi	10
	III	6	Sepi	10
	IV	23	sepi	12
2	I	26	sepi	13
	II	18	Sepi	10
	III	6	Sepi	10
	IV	23	sepi	12
3	I	52	Normal	26
	II	10	Sepi	10
	III	14	Sepi	10
	IV	72	Normal	36
4	I	60	Normal	30
	II	23	Sepi	12
	III	14	Sepi	10
	IV	72	Normal	36
5	I	73	Normal	36
	II	20	Sepi	19
	III	14	Sepi	10
	IV	69	Normal	34
6	I	37	Sepi	18
	II	6	Sepi	10
	III	67	Normal	33
	IV	52	Normal	26
7	I	44	Sepi	22
	II	23	Sepi	12
	III	48	Sepi	24
	IV	6	Sepi	10

No	Lajur	Presentase Kepadatan	Kondisi	Waktu Hijau (detik)
8	I	42	Sepi	21
	II	6	Sepi	10
	III	11	Sepi	10
	IV	6	Sepi	10
9	I	43	Sepi	21
	II	6	Sepi	10
	III	11	Sepi	10
	IV	6	Sepi	10
10	I	54	Normal	27
	II	7	Sepi	10
	III	11	Sepi	10
	IV	6	Sepi	10

Rata-rata pengujian dilakukan disiang hari adalah sebagai berikut :  

$$\frac{\text{jumlah presentase rata} - \text{rata kepadatan tiap pengujian}}{\text{jumlah pengujian}}$$

$$\frac{18 + 34 + 37 + 42 + 44 + 40 + 30 + 16 + 16 + 19}{10} = \frac{296}{10} = 29,6 = 30\%$$

Artinya kondisi kepadatan pada pengujian dipagi hari adalah sepi

**Tabel 4.3** Data Pengujian dilakukan disiang hari

No	Lajur	Presentase Kepadatan	Kondisi	Waktu Hijau (detik)
1	I	18	Sepi	10
	II	30	Sepi	15
	III	8	Sepi	10
	IV	42	Normal	20
2	I	18	Sepi	10
	II	38	Sepi	19
	III	12	Sepi	10
	IV	33	Sepi	16
3	I	25	Sepi	12
	II	43	Normal	41
	III	15	Sepi	10
	IV	37	sepi	18
4	I	23	Sepi	11
	II	48	Normal	24
	III	15	sepi	10
	IV	40	normal	20
5	I	15	Sepi	10
	II	46	Normal	23
	III	32	Sepi	16
	IV	37	sepi	18
6	I	15	Sepi	10
	II	11	Sepi	10
	III	27	Sepi	13

No	Lajur	Presentase Kepadatan	Kondisi	Waktu Hijau (detik)
	IV	20	sepi	10
7	I	18	Sepi	10
	II	20	Sepi	10
	III	42	Normal	21
	IV	28	Sepi	14
8	I	22	Sepi	11
	II	20	Sepi	10
	III	53	Normal	26
	IV	13	sepi	10
9	I	15	Sepi	10
	II	20	Sepi	10
	III	53	Normal	26
	IV	25	Sepi	12
10	I	15	Sepi	10
	II	32	Sepi	16
	III	44	Normal	22
	IV	27	Sepi	13

$$\frac{\text{jumlah presentase rata} - \text{rata kepadatan tiap pengujian}}{\text{jumlah pengujian}}$$

$$\frac{37 + 25 + 30 + 31 + 18 + 27 + 27 + 29 + 28 + 29}{10} = \frac{284}{10} = 28,4 = 28\%$$

Artinya kondisi kepadatan pada pengujian disiang hari adalah sepi

**Tabel 4.4** Data Pengujian dilakukan disore hari

No	Lajur	Presentase Kepadatan	Kondisi	Waktu Hijau (detik)
1	I	43	Normal	21
	II	40	Normal	20
	III	26	Sepi	13
	IV	85	Padat	42
2	I	27	Sepi	13
	II	29	Sepi	15
	III	26	Sepi	13
	IV	85	Padat	42
3	I	51	Normal	26
	II	26	Sepi	13
	III	36	Sepi	18
	IV	46	Normal	23
4	I	83	Padat	42
	II	59	Normal	30
	III	76	Padat	38
	IV	86	Padat	42
5	I	44	Normal	22
	II	39	Normal	20
	III	79	Padat	38
	IV	46	Normal	23

No	Lajur	Presentase Kepadatan	Kondisi	Waktu Hijau (detik)
6	I	45	Normal	22
	II	39	Normal	20
	III	62	Normal	31
	IV	75	Padat	38
7	I	45	Normal	23
	II	42	Normal	12
	III	29	Sepi	14
	IV	75	Padat	37
8	I	75	Padat	37
	II	76	Padat	38
	III	76	Padat	38
	IV	75	Padat	37
9	I	47	Normal	23
	II	49	Normal	24
	III	37	Sepi	19
	IV	81	Padat	41
10	I	45	Normal	22
	II	77	Padat	39
	III	45	Normal	23
	IV	81	Padat	41

$$\frac{\text{jumlah presentase rata - rata kepadatan tiap pengujian}}{\text{jumlah pengujian}}$$

$$\frac{48 + 41 + 39 + 76 + 52 + 52 + 56 + 76 + 53 + 62}{10}$$

52,5  
53%

Artinya kondisi kepadatan pada pengujian disore hari adalah Normal

**Tabel 4.5** Data Pengujian dilakukan didalam hari

No	Lajur	Presentase Kepadatan	Kondisi	Waktu Hijau (detik)
1	I	45	Normal	23
	II	77	Padat	39
	III	30	Normal	15
	IV	68	Normal	34
2	I	70	Normal	35
	II	55	Normal	28
	III	36	Sepi	18
	IV	75	Padat	38
3	I	80	Padat	40
	II	57	Normal	29
	III	40	Normal	20
	IV	76	Padat	38
4	I	77	Padat	39
	II	40	Normal	20
	III	77	Padat	39

No	Lajur	Presentase Kepadatan	Kondisi	Waktu Hijau (detik)
	IV	37	Sepi	18
5	I	47	Normal	24
	II	82	Padat	41
	III	70	Normal	35
	IV	80	Padat	40
6	I	37	Sepi	18
	II	52	Normal	26
	III	65	Normal	33
	IV	83	Padat	42
7	I	37	Sepi	18
	II	40	Normal	20
	III	55	Normal	28
	IV	76	Padat	38
8	I	43	Normal	22
	II	47	Normal	24
	III	90	Padat	45
	IV	82	Padat	41
9	I	58	Normal	29
	II	52	Normal	26
	III	83	Padat	42
	IV	35	Sepi	18
10	I	47	Normal	24
	II	37	Sepi	18
	III	98	Padat	49
	IV	43	Normal	22

$$\frac{\text{jumlah presentase rata - rata kepadatan tiap pengujian}}{\text{jumlah pengujian}}$$

$$\frac{56 + 59 + 63 + 58 + 70 + 52 + 60 + 52 + 66 + 46}{10}$$

58,2  
58%

Artinya kondisi kepadatan pada pengujian didalam hari adalah Normal

**PENUTUP**

Setelah melakukan penelitian terhadap simulasi pengaturan lampu lalu lintas berdasarkan data *image* kepadatan kendaraan dapat diambil kesimpulan dan saran sebagai berikut :

Berdasarkan hasil pengujian dan analisa pada bab sebelumnya, maka didapatkan kesimpulan antara lain :

1. Lama waktu nyala lampu hijau pada lampu lalu lintas bergantung dari hasil citra yang diambil oleh webcam. Semakin tinggi tingkat kepadatan

kendaraan semakin lama nyala lampu hijau di jalur tersebut juga sebaliknya.

2. Lama waktu nyala lampu merah pada lampu lalu lintas pada kondisi sepi, normal dan padat bergantung pada kondisi-kondisi kepadatan pada lajur lainnya.
3. Semakin tinggi letak kamera dari permukaan papan maka semakin luar area jangkauan kamera.
4. Proses deteksi kepadatan kendaraan hanya mengukur tingkat kepadatan yaitu berdasarkan seberapa besar citra jalan tertutup oleh citra kendaraan.

Saran yang dapat digunakan untuk pengembangan lebih lanjut terhadap penelitian ini adalah sebagai berikut :

1. Untuk perkembangan selanjutnya sebaiknya sistem sudah mampu menghitung jumlah kendaraan yang berada disetiap lajur jalan.
2. Setiap kamera webcam yang dipasang disetiap tiang lampu lalu lintas sebaiknya memiliki resolusi di atas 2 mega piksel sehingga hasil yang gambar yang dihasilkan akan lebih baik.
3. Sebaiknya menggunakan metode jaringan saraf tiruan atau fuzzy logic untuk menentukan parameter kepadatan kendaraan.
4. Sebaiknya teknik pengiriman data tidak lagi melalui media kabel melainkan menggunakan media wireless.
5. Untuk mengantisipasi pemadaman listrik oleh PLN, pengendali ini bisa menggunakan catu daya cadangan seperti sel surya

#### DAFTAR PUSTAKA

1. Achmad, Balza. 2011. Pemrogram Delphi Untuk Aplikasi Mesin Visi Menggunakan Webcam. Gava Media. Yogyakarta
2. Andrianto, Heri. 2008. Pemrograman mikrokontroler AVR ATmega16 Menggunakan Bahasa C (Code Vision AVR). Bandung. Informatika
3. Diktat Kuliah. 2008. Rekayasa Lalu Lintas. Universitas Widyagama. Malang.
4. Fadlisyah. 2007. Computer Vision dan Pengolahan Citra Digital. Andi Offset. Yogyakarta
5. Fadlisyah, dkk. 2010. Pemrograman Kamera PC Menggunakan Delphi. Graha Ilmu. Yogyakarta
6. Lingga, Wardana. 2006. Belajar Sendiri Mikrokontroler AVR Seri ATmega16, Simulasi Hardware dan Aplikasi. Andi. Yogyakarta
7. Supriadi, Muhammad. 2005. Pemrograman IC PPI 8255 Menggunakan Delphi. Andi. Yogyakarta
8. Sutoyo, dkk. 2009. Teori Pengolahan Citra Digital. Andi. Yogyakarta
9. Usman, Ahmad. 2005. Pengolahan Citra Digital & Teknik Pemrogramannya. Graha Ilmu. Jakarta

10. Widodo, Romy Budhi dan Joseph Dedi Irawan. 2007. Interfacing Paralel dan Serial Menggunakan Delphi. Graha Ilmu. Yogyakarta.