

## DESAIN SISTEM PENGATUR LAMPU LALU LINTAS DENGAN IDENTIFIKASI KEPADATAN KENDARAAN MENGUNAKAN METODE *SUBTRACTION*

**Geminiesty Lathifasari Djavendra\*, Siti Aisyah\*, Eko Rudiawan Jamzuri**

Program Studi Teknik Mekatronika, Jurusan Teknik Elektro, Politeknik Negeri Batam

\*Corresponding author, e-mail: [geminiesty@gmail.com](mailto:geminiesty@gmail.com), [siti\\_aisyah@polibatam.ac.id](mailto:siti_aisyah@polibatam.ac.id)

**Abstrak**— Bertambahnya jumlah kendaraan menyebabkan meningkatnya kepadatan lalu lintas yang menjadi salah satu faktor utama penyebab kemacetan. Kepadatan lalu lintas biasanya teralokasi di beberapa titik-titik tertentu di ruas jalan, salah satunya di persimpangan. Saat ini lalu lintas di persimpangan jalan diatur oleh lampu lalu lintas menggunakan sistem prediksi kepadatan lalu lintas. Sistem prediksi ini nantinya akan menentukan lama aktifnya lampu hijau dan lampu merah di setiap persimpangan. Salah satu sistem prediksi yang banyak digunakan adalah metode estimasi statistik kepadatan kendaraan. Metode lain pengontrolan kepadatan lalu lintas seperti sistem pemantauan secara visual memungkinkan untuk diterapkan guna menambah performansi sistem. Untuk itu penelitian ini mengusulkan pembuatan sebuah sistem pengontrolan lampu lalu lintas secara otomatis dengan prediksi kepadatan kendaraan menggunakan teknik pengolahan citra. Sistem yang dibangun menggunakan kamera untuk memantau kondisi kendaraan di jalan raya. Data gambar yang didapat dari kamera kemudian diolah menggunakan teknik pengolahan citra dan teknik pengurangan citra. Teknik ini membandingkan citra objek dengan citra referensi sehingga dapat diketahui jumlah piksel putih pada citra hasil pengurangan citra. Berdasarkan jumlah piksel putih yang telah diperoleh tersebut dapat diketahui persentase panjang antrian kendaraan dan kepadatan kendaraan. Data persentase yang diperoleh kemudian dikirim ke mikrokontroler untuk mengontrol durasi nyala lampu hijau. Pengontrolan lampu lalu lintas dengan perhitungan kepadatan kendaraan memiliki akurasi hingga 77.03% sedangkan dengan perhitungan panjang antrian kendaraan mencapai 91.18%.

*Kata Kunci* : Pengolahan citra, pengurangan citra, sistem kontrol lampu, kepadatan kendaraan, dilation, erosion

**Abstract**— The increasing number of vehicle causes the increasing of traffic density in which one of the main factors of congestion. Traffic density is usually allocated at certain points of roads, one at the intersection. In the novel technology, traffic in the crossroads had been controlled by traffic light using a traffic density prediction system. This prediction system would determine the duration of active green lights and red lights at each intersection. One of the most common prediction systems is a statistical estimation of vehicle density. Other method at controlling traffic density such as visually monitoring system might be implemented to increase system performance. Therefore, this research proposes an automatically traffic control system by predicting traffic density using image processing techniques. The proposed system is using a camera to visually monitor traffic condition. The image data obtained from the camera would be processed using an image processing and background subtraction techniques. This technique compared the captured-image with a reference image to result a subtracted-image depicted the traffic density which is represented by the number of white pixels. Based on the number of white pixels that have been obtained, the percentage of vehicle queue length and vehicle density can be determine. The percentage then sent to the microcontroller in order to control the duration of the active green light. The traffic light control system using traffic density calculation has an accuracy of up to 77.03% while using the calculation of vehicle queue length reached 91.18%.

*Keywords* : Image processing, image subtraction, light control system, traffic density, dilation, erosion

### 1. Pendahuluan

Di era modern saat ini, lalu lintas menjadi hal yang sangat penting agar mobilitas manusia tidak mengalami gangguan. Seiring dengan meningkatnya keperluan dan kesibukan

masyarakat agar dapat menjangkau suatu tempat dengan cepat menyebabkan kendaraan menjadi kebutuhan primer bagi sebagian besar masyarakat khususnya yang tinggal di kota - kota besar. Hal tersebut berdampak pada peningkatan

Received date 2018-4-12, Revised date 2018-5-6, Accepted date 2018-5-12

<https://doi.org/10.25077/jnte.v7n2.541.2018>

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

jumlah kendaraan setiap tahunnya. Peningkatan tersebut secara otomatis dapat mengakibatkan kemacetan di jalan raya. Selain itu, pengontrolan lampu lalu lintas yang kurang efisien dapat menjadi salah satu penyebab kemacetan. Durasi pengontrolan lampu lalu lintas yang konstan dapat menyebabkan timbulnya kemacetan di jalur yang padat apabila jalur lain dalam kondisi sepi atau kosong.

Pengontrolan lampu lalu lintas yang efisien menjadi salah satu solusi untuk meminimalisir kemacetan yang ada di jalan raya. Pengawasan kemacetan dapat dilakukan dengan memanfaatkan kamera sebagai detektor kemacetan di jalan raya yang dipasang di setiap simpang [1]. Data yang didapat dari kamera akan diolah dan nantinya akan digunakan sebagai data untuk mengontrol nyala lampu lalu lintas secara otomatis. Sistem pengaturan lampu lalu lintas ini bekerja berdasarkan masukan berupa gambar hasil *capture* yang diolah menggunakan metode pengolahan citra, dimana gambar diproses dari format RGB menjadi *grayscale*, *thresholding* dan *histogram* [1].

Penelitian lain memproses gambar menggunakan teknik *edge detection* untuk mendeteksi kendaraan dan *image matching* untuk mendapatkan persentase kondisi jalan sehingga dapat mengontrol durasi nyala lampu lalu lintas dengan akurasi 93,47% [2]. Metode *green colour subtraction* juga telah digunakan untuk dapat mengetahui kondisi kepadatan kendaraan [3].

Penggunaan *background subtraction* untuk memperoleh data volume kendaraan juga telah diterapkan pada Raspberry Pi dengan tingkat akurasi metode mencapai 87,6% [4]. Hanya saja, penelitian tersebut menerapkan metode *subtraction* dan juga metode *contour feature* untuk deteksi kendaraan. Metode deteksi objek, termasuk objek kendaraan, juga dapat dilakukan menggunakan metode kecerdasan buatan. Penerapan salah satu metode kecerdasan buatan yaitu algoritma *backpropagation* dikombinasi dengan metode pengolahan citra untuk deteksi objek terbukti cukup akurat dengan tingkat keberhasilan identifikasi dan deteksi hingga mencapai 99,02% [5].

Metode yang diusulkan pada penelitian ini adalah metode pengolahan citra (*smoothing*, *erosion* dan *dilation*) dikombinasi dengan *background subtraction* untuk mendeteksi objek kendaraan dengan menggunakan satu kamera untuk semua simpang. Secara *hardware*, sistem

ini terdiri atas beberapa bagian, antara lain kamera yang berfungsi untuk memantau kondisi jalan, lalu hasil pengamatan diolah di PC menggunakan metode pengolahan citra sehingga didapatkan jumlah piksel putih untuk mengetahui persentase panjang antrian dan kepadatan kendaraan di setiap jalur jalan. Kemudian data persentase dikirim ke mikrokontroler ATmega328 yang berfungsi sebagai piranti untuk mengontrol setiap perubahan lampu lalu lintas di setiap simpang.

## 2. Tinjauan Pustaka

Pengolahan citra merupakan suatu metode yang digunakan untuk mengolah gambar sehingga menghasilkan gambar yang sesuai dengan yang diinginkan. Pengambilan gambar dalam pengolahan citra ini dapat dilakukan dengan menggunakan gambar dari berkas yang sudah ada, ataupun dengan menggunakan kamera.

Citra merupakan kumpulan piksel yang disusun dalam larik dua dimensi. Indeks baris dan kolom (x,y) dari sebuah piksel dinyatakan dalam bilangan bulat. Piksel (0,0) terletak pada sudut kiri atas pada citra, indeks x bergerak ke kanan dan indeks y bergerak kebawah.

### 2.1. Grayscale

Dalam pengolahan citra, *grayscale* merupakan suatu citra yang hanya memiliki satu nilai warna pada setiap piksel-nya, dimana warna dari *grayscale* adalah warna keabuan yang berada diantara hitam dan putih, dengan kata lain nilai *red = green = blue*. Sistem warna *grayscale* digunakan karena pada proses *threshold*, citra yang dapat diproses adalah citra dengan derajat keabuan atau disebut dengan *grayscale* [6]. Persamaan 1 merupakan rumus untuk mendapatkan nilai *grayscale* dari citra RGB.

$$I_{Gray}(x,y) = 0.299I_R(x,y) + 0.587I_G(x,y) + 0.114I_B(x,y) \quad (1)$$

dengan  $I_{Gray}(x,y)$  merupakan hasil luaran *grayscale*, sedangkan  $I_R(x,y)$  merupakan piksel *red* pada citra,  $I_G(x,y)$  merupakan piksel *green* pada citra dan  $I_B(x,y)$  merupakan piksel *blue* pada citra.

### 2.2. Smoothing

*Smoothing* atau sering disebut *blurring* adalah salah satu teknik dalam pengolahan citra yang digunakan untuk

mengurangi *noise* pada suatu citra objek. Pada sistem pengolahan citra, seringkali gambar yang ditangkap oleh perangkat kamera tidak fokus. Hal ini terjadi salah satunya dikarenakan perubahan objek gambar yang menyebabkan munculnya *noise*. Oleh karena itu, dengan menggunakan teknik *blurring* ini akan menghapus detail - detail kecil dari gambar. Salah satu teknik *blurring* yang banyak digunakan yaitu *average* yang dinyatakan pada persamaan 2 [7].

$$O(x, y) = \frac{1}{M} \sum_{i=1}^M u_i(x, y) \quad (2)$$

dengan  $O$  menyatakan citra luaran,  $M$  menyatakan banyak matrik, sedangkan  $u$  menyatakan citra ke-  $i$ .

### 2.3. Subtraction

*Subtraction* merupakan suatu proses pengurangan dua buah citra untuk mengidentifikasi beda kedua citra. Sebagian menggunakan metode ini untuk mengenali objek bergerak. Konsep dari metode ini adalah mengidentifikasi sebuah citra yang disebut *background image* (citra tanpa objek) dan membandingkannya dengan *foreground image* (citra dengan objek) untuk mendeteksi objek seperti ditunjukkan pada persamaan 3. Hasil pengurangan piksel pada *foreground image* dengan *background image* secara absolut akan selalu menghasilkan nilai positif [8].

$$R(x, y) = I(x, y) - B(x, y) \quad (3)$$

dengan  $R(x, y)$  merupakan hasil dari pengolahan *subtraction*,  $I(x, y)$  merupakan piksel citra objek atau *foreground image*, sedangkan  $B(x, y)$  merupakan piksel citra *background image*.

### 2.4. Threshold

*Threshold* merupakan nilai batas ambang untuk membagi sebuah citra menjadi dua kondisi yaitu *foreground* dan *background*. Di dalam teknik pengolahan citra, *threshold* dapat digunakan sebagai nilai penentu dalam mengubah citra *gray* menjadi citra *binary*. Didalam proses perubahan citra *grayscale* ke citra *binary*, piksel yang memiliki nilai lebih atau sama dengan nilai ambangnya akan bernilai 255 (berwarna putih), sedangkan jika nilai piksel lebih kecil dari nilai ambang maka nilai citra piksel adalah 0 (berwarna hitam) seperti ditunjukkan pada persamaan 4 [9].

$$I_{Threshold}(x, y) = \begin{cases} 0 & I_{Gray}(x, y) < T \\ 255 & I_{Gray}(x, y) \geq T \end{cases} \quad (4)$$

dengan  $I_{Threshold}(x, y)$  merupakan hasil luaran citra setelah di *threshold*, sedang  $T$  merupakan nilai ambang yang akan digunakan.

### 2.5. Erosion

*Erosion* merupakan suatu proses yang berfungsi untuk memindahkan atau mengurangi piksel pada batas dari suatu objek citra, sehingga objek akan tampak mengecil. Proses *erosion* dilakukan dengan membandingkan setiap piksel citra dengan nilai pusat *SE (Structuring Elements)* dengan cara melapiskan *SE* dengan citra sehingga pusat *SE* tepat dengan posisi piksel citra yang diproses. Dalam citra *binary*, jika semua piksel pada *SE* tepat sama dengan semua nilai objek citra maka piksel diatur nilainya sama dengan nilai piksel objek, bila tidak maka piksel diberi nilai sesuai dengan piksel *background* seperti ditunjukkan pada persamaan 5 [9].

$$D(A, B) = A \ominus B = \{x: B_x \subset A\} \quad (5)$$

$A \ominus B$  adalah operasi erosi A dengan B.

### 2.6. Dilation

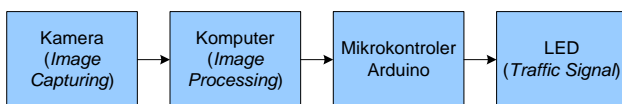
*Dilation* adalah suatu proses yang berfungsi untuk menambahkan piksel pada batas dari suatu objek citra, sehingga objek akan terlihat semakin besar. Proses *dilation* dilakukan dengan membandingkan setiap piksel citra dengan nilai pusat *SE (Structuring Elements)* dengan cara melapiskan *SE* dengan citra sehingga pusat *SE* tepat dengan posisi piksel citra yang diproses. Dalam citra *binary*, jika paling sedikit terdapat 1 piksel pada *SE* sama dengan nilai objek citra maka piksel diatur nilainya sesuai dengan nilai piksel objek dan bila semua piksel yang berhubungan adalah *background* maka piksel diberi nilai sesuai dengan piksel *background* seperti ditunjukkan pada persamaan 6 [9].

$$D(A, B) = A \oplus B = \{x: B_x \cap A \neq \emptyset\} \quad (6)$$

$\emptyset$  menyatakan himpunan kosong. Suatu objek (masukan citra) dinyatakan dengan  $A$  dan *SE* dinyatakan dengan  $B$ .  $B_x$  menyatakan translasi  $B$  sedemikian sehingga pusat  $B$  terletak pada  $x$ .  $A \oplus B$  adalah operasi dilasi A dengan B.

### 3. Metoda

Pada penelitian ini, digunakan beberapa perangkat keras untuk mendukung proses kerja sistem agar dapat berjalan dengan baik dan perangkat lunak yang berperan penting untuk mengontrol dan memonitoring sistem. Perangkat keras yang digunakan yaitu sensor kamera yang berfungsi untuk menangkap gambar kondisi kendaraan di jalan raya. Dimana kamera yang digunakan adalah kamera *Logitech HD Webcam C270*. Gambar yang telah didapat kemudian diterima dan diolah oleh PC menggunakan metode pengolahan citra. Metode ini berfungsi untuk mendapatkan nilai persentase kepadatan dan panjang antrian kendaraan. PC terhubung langsung ke mikrokontroler *Arduino Uno* melalui *serial port*. Mikrokontroler *Arduino Uno* berfungsi untuk memberikan sinyal perubahan pada lampu lalu lintas. Sistem yang dibangun diperlihatkan pada gambar 1.



Gambar 1. Diagram blok sistem pengaturan lampu lalu lintas dengan metode pengolahan citra

Metode pengolahan citra dilakukan melalui 2 tahap yaitu:

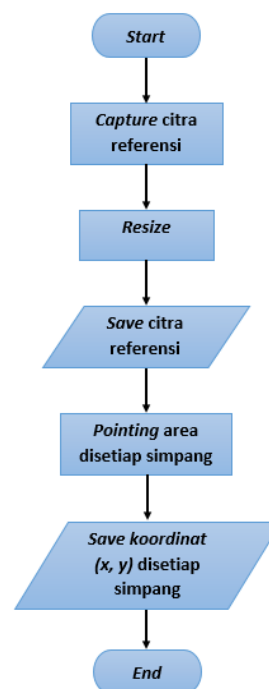
a. Pengolahan citra referensi

Citra referensi yaitu citra yang didapat saat kondisi jalan kosong atau tidak ada kendaraan. Proses pengambilan citra referensi dilakukan ditahap awal dan ditetapkan sebagai *background image*. Setelah itu *resize* citra referensi sesuai dengan ukuran yang ditentukan, lalu menyimpan citra tersebut ke dalam format *bitmap* (bmp). Kemudian dilanjutkan dengan proses *pointing area* disetiap simpang sebagai batas area untuk mengetahui kondisi jalan pada citra referensi. Setelah itu, data koordinat area yang telah ditentukan disimpan ke file .ini yang telah disediakan. Proses pengolahan citra referensi dapat dilihat dalam diagram alir gambar 2.

b. Pengolahan citra objek

Proses awal pada tahap kedua yaitu mengambil data referensi yang telah disimpan sebelumnya berupa citra referensi dalam format *bitmap* (bmp) dan data koordinat piksel disemua simpang. Citra referensi yang di-load dikonversi menjadi citra *grayscale*. Selanjutnya

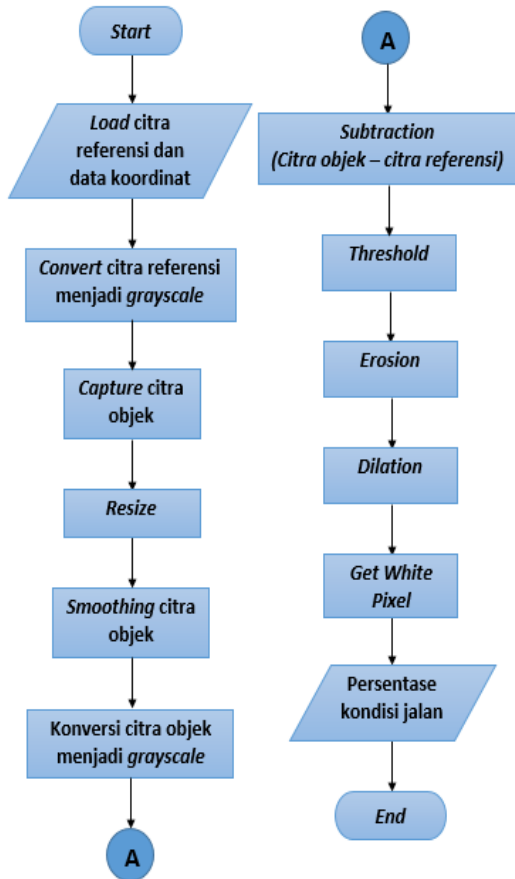
mengaktifkan kamera untuk mengambil citra objek atau *foreground image* secara *realtime*. Citra objek yang telah didapat kemudian diubah ukurannya. Citra objek yang telah didapat kemudian di-*smoothing* menggunakan teknik *average*. Teknik ini berfungsi untuk mengurangi *noise* akibat pengambilan gambar yang tidak fokus menggunakan kamera. Selanjutnya merubah citra objek RGB menjadi citra *grayscale* yang berfungsi agar citra dapat diimplementasikan pada proses *threshold*.



Gambar 2. Urutan pengolahan citra referensi

Berikutnya melakukan proses *subtraction* yang berfungsi untuk mendapatkan citra objek, dimana citra objek didapat dari hasil pengurangan piksel dari *foreground image* (citra dengan kondisi jalan terdapat kendaraan) dengan *background image* (citra dengan kondisi jalan kosong). Setelah dilakukan proses *subtraction*, citra akan dikonversi menjadi citra *binary* dengan teknik *threshold*. Setelah dilakukan proses *binary* selanjutnya dilakukan proses *erosion* dan *dilation*. *Erosion* berfungsi untuk mengikis piksel citra yang tidak menyatu pada piksel citra objek, akan tetapi pada proses ini piksel pada citra objek akan ikut terkikis sehingga perlu dilakukan proses *dilation* untuk memulihkan piksel pada citra objek yang terkikis. Dimana *dilation* berfungsi untuk memulihkan bagian-bagian tepi yang kosong pada citra objek. Selanjutnya

dilakukan proses *get white pixels* disetiap disimpang, sehingga dapat diketahui nilai persentase panjang antrian dan kepadatan kendaraan. Data persentase tersebut akan dijadikan masukan untuk mengontrol durasi nyala lampu hijau pada lampu lalu lintas. Diagram alir lengkap proses pengolahan citra objek diperlihatkan pada gambar 3.



Gambar 3. Urutan pengolahan citra objek

Pengontrolan nyala lampu hijau pada mikrokontroler berdasarkan data persentase yang telah didapat dari hasil pengolahan citra ditentukan dengan durasi nyala lampu hijau minimum dan maksimum masing – masing 3 dan 20 detik. Adapun algoritma lengkap dijelaskan pada pseudocode berikut:

```

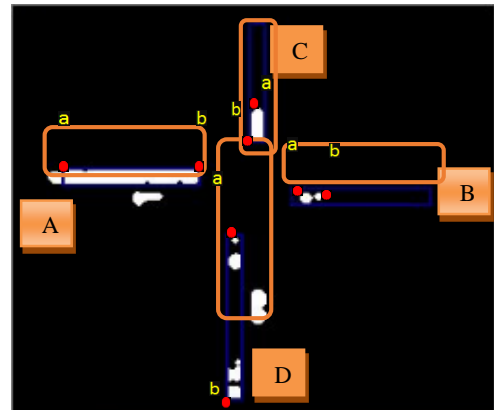
Deklarasi:
Persentase : integer
Pembagi = 6 : integer
Delay      : integer
Led red, led yellow, led green: boolean

Algoritma:
If Led red = true
write(Led red)
delay ← 1 detik
    
```

```

If Led red, led yellow = true
write(Led red, led yellow)
delay ← 3 detik
read(Persentase, Pembagi)
If led green = true, Led red = false, led
yellow = false
write(Led red, led yellow, led green)
delay ← 3 + (Persentase piksel putih
/Pembagi)
If led green = false, led yellow = true
write(led yellow)
delay ← 3 detik
If Led red = true
write(Led red)
If Led red, led yellow = true
Loop simpang berikutnya ← write(Led red, led
yellow)
delay ← 3 detik
    
```

Berdasarkan algoritma pengontrolan durasi nyala lampu hijau, dapat diketahui jika kondisi jalan kosong durasi lama nyala lampu hijau adalah 3 detik, sedangkan jika terdapat kendaraan maka durasi nyala lampu hijau yaitu 3 ditambah dengan (persentase piksel putih / 6), sehingga saat jalan dalam kondisi padat durasi nyala lampu hijau maksimum 20 detik.



Gambar 4. Hasil simulasi sistem dengan tampilan piksel putih di setiap simpang

Gambar 4 menunjukkan tampilan simulasi penelitian yang telah dilakukan, dimana banyaknya jumlah piksel didalam area *rectangle* (dibatasi garis biru) pada simpang:

- A (Kiri) : 12 x 91 Piksel
- B (Kanan) : 14 x 91 Piksel
- C (Atas) : 79 x 12 Piksel
- D (Bawah) : 111 x 12 Piksel

a) Metode Kepadatan Kendaraan

Metode kepadatan kendaraan menggunakan rumus 7.

$$\text{Persentase kepadatan} = \frac{\text{total piksel putih} \times 100}{\text{luas area rectangle}} \quad (7)$$

Pada piksel citra *binary* dengan ukuran 7x6 seperti ditunjukkan pada gambar 5, dilakukan proses *scan line* di area *rectangle* sehingga diperoleh banyaknya piksel putih (nilai 255) yaitu 24 piksel, dengan perhitungan:

$$\text{Persentase kepadatan} = \frac{24 \times 100}{42} = 57.14\%$$

	(0,0)	(1,0)				(n-1,0)
	0	0	0	0	0	0
(0,1)	0	0	255	255	255	255
	0	255	255	255	255	255
	255	255	255	255	255	0
	255	255	255	255	255	255
	0	255	255	255	255	0
(0, n-1)	0	0	0	0	0	0

Gambar 5. Piksel citra biner dengan ukuran 7 x 6 untuk perhitungan kepadatan kendaraan

b) Metode Panjang Antrian Kendaraan

Untuk mendapatkan persentase panjang antrian kendaraan digunakan rumus 8.

$$\text{Persentase panjang antrian} = \frac{\text{panjang piksel putih} \times 100}{\text{panjang area rectangle}} \quad (8)$$

Diambil contoh piksel citra *binary* dengan resolusi 9 x 5, kemudian dilakukan proses *scan line* di area *rectangle* sehingga diperoleh panjang piksel putih sebanyak 9 piksel. Dimana titik a berada pada piksel (0, 1) dan titik b pada piksel (0, 6), maka panjang piksel putih adalah 6 - 1 = 5. Dengan demikian dapat dihitung persentase panjang antrian sebagai berikut:

$$\text{Persentase panjang antrian} = \frac{5 \times 100}{9} = 55.5\%$$

	(0,0)	(1,0)				(n-1,0)
	0	0	0	0	0	0
(0,1)	0	0	0	255	255	a
	0	0	0	0	0	
	0	255	255	255	255	
	255	255	255	255	255	
	255	255	0	0	255	
	0	255	0	255	255	b
	0	0	0	0	0	
(0, n-1)	0	0	0	0	0	

Gambar 6. Piksel citra biner dengan ukuran 9 x 5 untuk perhitungan panjang antrian kendaraan

4. Hasil dan Pembahasan

Pengujian sistem pada penelitian ini menggunakan citra Gambar 4, dengan *rectangle* biru berfungsi sebagai batas area untuk mengetahui persentase kepadatan ataupun panjang antrian berdasarkan jumlah piksel putih yang berada didalam area tersebut.

4.1. Pengujian Sistem Berdasarkan Perhitungan Kepadatan Kendaraan

Pengujian sistem dengan perhitungan kepadatan kendaraan dilakukan dengan proses *scan line* pada area *rectangle* dan menghitung jumlah piksel putih pada area tersebut. Dari pengujian diperoleh jumlah piksel putih untuk area:

- kiri : 843 piksel
- kanan : 72 piksel
- atas : 177 piksel
- bawah : 236 piksel

4.2. Pengujian Sistem Berdasarkan Perhitungan Panjang Antrian Kendaraan

Untuk mendapatkan nilai persentase panjang antrian kendaraan, dilakukan proses *scan line* hingga didapatkan piksel putih (*get white pixels*) didalam area *rectangle*. Pada simpang atas dan bawah, proses *get white pixels* dilakukan berdasarkan sumbu y sedangkan pada simpang kiri dan kanan proses *get white pixels* dilakukan berdasarkan sumbu x. Pada simpang atas dan bawah proses *scan line* dilakukan dari piksel y-minimum (0,0) yang bergerak terhadap sumbu x didalam area *rectangle*. Apabila saat proses *scan line*, piksel putih pertama terdeteksi maka titik a akan menyimpan panjang piksel y, kemudian proses pergerakan *scan line* berpindah ke piksel y-maksimum (0, n-1) yang bergerak menuju piksel y-minimum terhadap sumbu x pada area *rectangle*. Apabila saat proses *scan line*, piksel putih terdeteksi maka titik b akan menyimpan panjang piksel y. Sehingga dapat diketahui panjang piksel putih dengan mengurangi piksel pada titik b dengan piksel pada titik a (titik b – titik a).

Dari hasil pengujian yang dilakukan, banyaknya jumlah piksel putih untuk area:

Kiri : 90 Piksel      Kanan : 15 Piksel  
 Atas : 22 Piksel      Bawah : 107 Piksel

Dari percobaan yang telah dilakukan dapat dilihat hasil persentase kondisi kendaraan di jalan berdasarkan panjang antrian kendaraan dan kepadatan kendaraan pada tabel 1. Pada penelitian ini telah diuji bahwa jumlah mobil yang dapat masuk pada area *rectangle* yang telah ditentukan yaitu 4 mobil dengan ukuran sama, dengan asumsi ada jarak antar mobil. Sehingga persentase kepadatan kendaraan pada *area rectangle* akan mencapai 100% jika terdapat lebih dari 4 mobil.

Untuk mengetahui akurasi data yang telah didapat pada tabel 1, maka dibutuhkan data persentase kondisi kendaraan di jalan secara *real* pada perancangan alat yang telah dibuat. Data persentase kendaraan secara *real* dapat dilihat pada tabel 2.

Dari data tabel 1 dapat diketahui bahwa hasil simulasi menggunakan metode panjang antrian kendaraan lebih baik dibandingkan metode kepadatan kendaraan. Untuk kondisi nomor 1 pada tabel 1, apabila terdapat 1 mobil didalam area *rectangle* maka data persentase panjang antrian adalah 22%, sedangkan persentase kepadatan adalah 16%. Jika dibandingkan dengan data persentase secara *real* pada tabel 2, nilai persentase panjang antrian kendaraan dan kepadatan kendaraan saat kondisi jalan sepi masing - masing adalah 23.44% dan 20.51%, sehingga nilai rata - rata akurasi yang didapat dari 10 kali percobaan yang dilakukan yaitu masing - masing 91.18% dan 77.03%.

Tabel 1. Data persentase kepadatan lalu lintas berdasarkan metode panjang antrian dan kepadatan kendaraan dengan kontrol durasi nyala lampu hijau

No	Simpan g	Kondis i	Metode Panjang Antrian		Metode Kepadatan	
			Persentase Panjang Antrian (%)	Waktu Hijau (detik)	Persentase Kepadatan (%)	Waktu Hijau (detik)
1	Kiri	Sepi	22	6	16	5
	Kanan	Sepi	23	6	15	5
	Atas	Sepi	28	6	23	6
	Bawah	Sepi	0	3	0	3
2	Kiri	Sepi	22	6	18	6
	Kanan	Padat	58	12	41	10
	Atas	Norma l	52	11	46	10
	Bawah	Sepi	17	5	10	4
3	Kiri	Padat	71	14	52	11
	Kanan	Norma l	48	11	31	8

4	Atas	Sepi	0	3	0	3
	Bawah	Padat	64	13	37	9
	Kiri	Sepi	0	3	0	3
	Kanan	Sepi	0	3	0	3
	Atas	Padat	75	15	61	12
5	Bawah	Sangat Padat	85	17	60	11
	Kiri	Norma l	45	10	26	7
	Kanan	Sepi	0	3	0	3
	Atas	Sangat Padat	100	20	82	16
	Bawah	Sangat Padat	90	18	56	12

Tabel 2. Data real persentase kepadatan lalu lintas pada area *rectangle* dengan simulasi kondisi sepi (1 mobil), normal (2 mobil), padat (3 mobil) dan sangat padat (4 atau lebih mobil)

Kondisi	Persentase Panjang Antrian (%)	Persentase Kepadatan (%)
Sepi	23.44	20.51
Normal	50	43.75
Padat	76.56	66.99
Sangat Padat	100	100

Dari hasil tabel 1 dan 2, dapat dilihat bahwa metode panjang antrian kendaraan lebih presisi jika dibandingkan dengan metode kepadatan kendaraan dengan asumsi jarak antar kendaraan tidak cukup jauh. Hal ini dikarenakan metode panjang antrian akan memprediksi nilai persentase kepadatan kendaraan 100% jika terdapat kendaraan di posisi minimum dan maksimum area *rectangle* walaupun di posisi tengah tidak terdapat kendaraan. Kemudian persentase piksel putih pada metode kepadatan kendaraan juga kurang akurat dikarenakan adanya proses *filter* pengolahan citra berupa *dilation* dan *erosion* dan adanya jarak antar kendaraan yang cenderung mengurangi jumlah piksel putih.

## 5. Kesimpulan

Setelah melakukan perencanaan pembuatan, pengujian dan analisa sistem, maka dapat diambil kesimpulan dari sistem kerja alat yaitu sebagai berikut:

1. Berdasarkan hasil luaran yang dihasilkan, terdapat 2 metode untuk mengetahui persentase kendaraan, yaitu:
    - a. Menghitung jumlah seluruh piksel putih yang berada didalam area *rectangle* atau berdasarkan kepadatan.
    - b. Mendapatkan titik paling atas dan titik paling bawah piksel putih yang berada di area *rectangle* atau berdasarkan panjang antrian.
  2. Tingkat akurasi sistem kendali kepadatan lalu lintas menggunakan perhitungan kepadatan kendaraan dan panjang antrian kendaraan masing-masing mencapai 77.03% dan 91.18%.
  3. Penggunaan metode panjang antrian lebih akurat untuk kondisi kepadatan lalu lintas yang rapat yaitu posisi kendaraan dekat satu dengan yang lainnya.
  4. Persentase piksel putih sebagai representasi kepadatan pada metode kepadatan kendaraan tidak cukup akurat dikarenakan adanya proses *dilation* dan *erosion* serta adanya jarak antar kendaraan yang cenderung mengurangi jumlah piksel putih.
- [5] Kasdianto dan Siti Aisyah, *Desain Sistem Pendeteksi untuk Citra Base Sub-Assembly dengan Algoritma Backpropagation*, Jurnal Rekayasa Elektrika, Vol. 13, No. 1, April 2017, hal. 1-7.
  - [6] Wan Yi, Xie Qisong, *A Novel Framework for Optimal RGB to Grayscale Image Conversion*, International Conference on Intelligent Human-Machine Systems and Cybernetics, 2016.
  - [7] Praba D. Surya daold & Kumar J. Satheesh, *Performance Analysis of Image Smoothing Methods for Low Level of Distortion*, IEEE International Conference on Advances in Computer Applications (ICACA), 2016.
  - [8] Wang Yuan-Kai, Chen Hung-Yu, *The Design of Background Subtraction on Reconfigurable Hardware*, Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IEEE (2012).
  - [9] Putra Darma, *Pengolahan Citra Digital*, Penerbit Andi, 2010.

#### Daftar Pustaka

- [1] Jatmika Sunu, dan Indra Andiko, *Simulasi Pengaturan Lampu Lalu Lintas Berdasarkan Data Image Processing Kepadatan Kendaraan Berbasis Mikrokontroler Atmega16*, Jurnal Ilmiah Teknologi dan Informasi ASIA, Vol.8, No.2, pp.81, Agustus 2014.
  - [2] M. Chandrasekhar, etc., *Traffic Control Using Digital Image Processing*, International Journal Of Advanced Electrical and Electronics Engineering, Vol.2, No.5, pp.96, 2013.
  - [3] Gaikwad Omkar Ramdas, etc., *Image Processing Based Traffic Light Control*, International Journal of Science, Engineering and Technology Research (IJSETR), Vol. 3, Issue 4, April 2014.
  - [4] Fitria Rahmadina dan Zaini, *Sistem Informasi Kepadatan Lalu Lintas Berbasis Raspberry pi PC Board*, Jurnal Nasional Teknik Elektro, Vol: 5, No. 1, Maret 2016.
- Biodata Penulis**
- Gemiesty Lathifasari Djavendra** lahir di Pekanbaru tanggal 26 Mei 1996. Menerima gelar Sarjana Sains Terapan dari Jurusan Teknik Elektro Politeknik Negeri Batam tahun 2017.
- Siti Aisyah** memperoleh gelar Sarjana Teknik dari Institut Teknologi Sepuluh Nopember (ITS) dan gelar Master di The Queens University of Belfast, Northern Ireland, United Kingdom. Bidang penelitiannya meliputi pengolahan citra, *wireless telemetry system*, *IoT*, *artificial intelligence* dan sistem kontrol.
- Eko Rudiawan Jamzuri** memperoleh gelar Sarjana Sains Terapan dari Institut Teknologi Bandung (ITB) pada tahun 2013. Bidang penelitiannya meliputi robotika, *computer vision*, *artificial intelligence* dan *factory automation*.