



**TUGAS AKHIR - TE 141599**

**ESTIMASI PANJANG ANTREAN KENDARAAN PADA  
PERSIMPANGAN JALAN RAYA DENGAN SENSOR KAMERA  
MENGUNAKAN METODE *QUEUE LENGTH ESTIMATION***

Dwi Kuncoro  
NRP 2209100198

Dosen Pembimbing  
Arief Kurniawan, S.T., M.T.  
Ahmad Zaini, S.T., M.Sc.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018



**FINAL PROJECT - TE 141599**

**QUEUE LENGTH ESTIMATION OF VEHICLE  
IN ROAD JUNCTION WITH CAMERA SENSOR  
USING QUEUE LENGTH ESTIMATION METHOD**

Dwi Kuncoro  
NRP 2209100198

Advisors  
Arief Kurniawan, S.T., M.T.  
Ahmad Zaini, S.T., M.Sc.

DEPARTMENT OF ELECTRICAL ENGINEERING  
Faculty of Electrical Technology  
Sepuluh Nopember Institute of Technoogy  
Surabaya 2018

## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul “**Estimasi Panjang Antrean Kendaraan pada Persimpangan Jalan Raya dengan Sensor Kamera Menggunakan Metode *Queue Length Estimation***” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa bahan yang tidak diijinkan dan bukan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada Daftar Pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2018

Dwi Kuncoro  
2209100198

**ESTIMASI PANJANG ANTREAN KENDARAAN  
PADA PERSIMPANGAN JALAN RAYA  
DENGAN SENSOR KAMERA  
MENGUNAKAN METODE *QUEUE LENGTH ESTIMATION***

**TUGAS AKHIR**

Diajukan untuk Memenuhi Sebagian Persyaratan  
untuk Memperoleh Gelar Sarjana Teknik  
pada  
Bidang Studi Teknik Komputer dan Telematika  
Departemen Teknik Elektro  
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II

  
Arief Kurniawan, S.T., M.T.  
NIP. 197409072002121001

  
Ahmad Zaini, S.T., M.Sc.  
NIP. 197504192002121003



## ABSTRAK

Nama : Dwi Kuncoro  
Judul : Estimasi Panjang Antrean Kendaraan pada  
Persimpangan Jalan Raya dengan Sensor  
Kamera Menggunakan Metode *Queue  
Length Estimation*  
Pembimbing : 1. Arief Kurniawan, S.T., M.T.  
2. Ahmad Zaini, S.T., M.Sc.

Kemacetan lalu-lintas merupakan salah satu masalah yang banyak terjadi di kota-kota besar, khususnya di persimpangan yang dekat dengan pusat keramaian seperti kawasan industri, perkantoran, sekolah, dan sebagainya. Adanya pusat-pusat keramaian tersebut menyebabkan sebagian ruas jalan memiliki kepadatan yang tinggi, sedangkan ruas yang lain memiliki kepadatan yang rendah. Sementara itu, sistem lampu lalu-lintas yang ada masih menggunakan sistem pewaktuan yang tidak adaptif sehingga terjadi ketidakseimbangan antrean kendaraan pada persimpangan tersebut. Masalah ini dapat dikurangi, misalnya dengan menerapkan sistem *Smart Traffic Light* menggunakan *traffic IP camera*. Pendekatan yang digunakan adalah menghitung panjang antrean kendaraan di tiap ruas jalan pada saat lampu merah menyala, menggunakan metode *queue length estimation*. Kemudian hasil perhitungannya dapat digunakan untuk menentukan durasi lampu hijau untuk tiap ruas jalan pada siklus berikutnya. Hasil dari penelitian ini menunjukkan bahwa metode *queue length estimation* dapat diterapkan untuk deteksi panjang antrean kendaraan, tetapi akurasi belum cukup baik, sehingga diperlukan tambahan teknik lain agar hasil estimasinya lebih akurat.

Kata kunci : *smart traffic light, traffic IP camera, adaptive traffic light, queue length estimation*

*Halaman ini sengaja dikosongkan*

## ABSTRACT

*Name* : Dwi Kuncoro  
*Title* : *Queue Length Estimation of Vehicle in Road Junction with Camera Sensor Using Queue Length Estimation Method*  
*Advisors* : 1. Arief Kurniawan, S.T., M.T.  
2. Ahmad Zaini, S.T., M.Sc.

*Traffic jam is one of the problems commonly happen in a big city, especially in a junction which is located near a crowded place, such as industrial area, office affair, school, etc. Those crowded places cause some roads have a high density, while the others have a low density. Meanwhile, the current traffic light system still use a non-adaptive timing system so that there is unbalanced vehicle queue among roads in the junction. It can be decreased by using Smart Traffic Light with traffic IP camera. The approach used in this case is calculating vehicle queue length in each road when the red light is on, using queue length estimation method. Then the result is used to calculate a time duration for green light on next cycle. The result shows that queue length estimation method can be used to detect vehicle queue length, but the accuracy is not good enough, therefore it requires additional techniques in order to get more accurate result.*

*Keywords* : *smart traffic light, traffic IP camera, adaptive traffic light, queue length estimation*

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Puji syukur penulis ucapkan atas karunia Allah Subhanahu Wa Ta'ala berupa rahmat dan hidayah-Nya, yang dengannya penulis dapat menyelesaikan penelitian Tugas Akhir dengan judul “**Estimasi Panjang Antrean Kendaraan pada Persimpangan Jalan Raya dengan Sensor Kamera Menggunakan Metode *Queue Length Estimation***”. Tanpa karunia-Nya, tidak mungkin penulis mampu mewujudkan penelitian ini.

Penelitian ini disusun dalam rangka memenuhi kebutuhan riset di Departemen Teknik Elektro ITS, Bidang Studi Teknik Komputer dan Telematika, serta sebagai persyaratan menyelesaikan pendidikan S1 bagi penulis.

Penelitian ini dapat terselesaikan atas bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Orang tua dan keluarga tercinta yang telah memberikan bantuan material maupun dorongan spiritual dalam penyelesaian buku penelitian ini
2. Bapak Dr. Ardyono Priyadi, S.T., M.Eng. selaku Ketua Departemen Teknik Elektro, Fakultas Teknologi Elektro, ITS Surabaya
3. Bapak Arief Kurniawan, S.T., M.T. dan Bapak Ahmad Zaini, S.T., M.Sc. atas bimbingan selama mengerjakan penelitian ini
4. Segenap dosen dan karyawan di Departemen Teknik Elektro dan Departemen Teknik Komputer
5. Rekan-rekan B201 Laboratorium Bidang Studi Teknik Komputer dan Telematika

Tiada gading yang tak retak. Tidak ada karya manusia yang sempurna. Oleh karena itu, penulis terbuka untuk menerima kritik dan saran yang membangun dari para pembaca sekalian. Semoga penelitian ini dapat memberikan manfaat bagi penulis secara khusus, dan dunia pendidikan serta ilmu pengetahuan secara umum. Aamiin.

Surabaya, Januari 2018

Penulis

*Halaman ini sengaja dikosongkan*

## DAFTAR ISI

ABSTRAK .....	i
ABSTRACT .....	iii
KATA PENGANTAR .....	v
DAFTAR ISI .....	vii
DAFTAR GAMBAR .....	ix
DAFTAR TABEL .....	xi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Permasalahan .....	1
1.3 Tujuan .....	2
1.4 Batasan Masalah .....	2
1.5 Sistematika Penulisan .....	2
1.6 Relevansi .....	3
BAB II TINJAUAN PUSTAKA .....	5
2.1 <i>Smart Traffic Light</i> .....	5
2.2 <i>Traffic IP Camera</i> .....	5
2.3 <i>Background Reconstruction</i> .....	6
2.4 <i>Background Subtraction</i> .....	7
2.5 <i>Binary Tresholding</i> .....	9
2.6 Dilasi dan Erosi .....	10
2.7 <i>Traffic Management System</i> .....	10
2.8 <i>Queue Length Estimation</i> .....	11
2.9 <i>Progressive Block Processing</i> .....	12
2.10 Penelitian Terkait .....	12
Menghitung Kendaraan Roda Empat .....	12
Real Time Vehicle Counter System .....	13
Deteksi Kecepatan Kendaraan Bermotor .....	13
BAB III DESAIN DAN IMPLEMENTASI SISTEM .....	15
3.1 Desain Sistem .....	15
3.2 Alur Kerja .....	17
3.3 <i>Background Reconstruction</i> .....	17
3.4 <i>Background Subtraction</i> .....	18
3.5 <i>Thresholding</i> .....	18
3.6 Erosi .....	18

3.7 Dilasi .....	19
3.8 Deteksi Objek .....	19
3.9 Penghitungan <i>Queue Length Estimation</i> .....	20
<b>BAB IV PENGUJIAN DAN ANALISIS</b> .....	<b>23</b>
4.1 <i>Background Reconstruction</i> .....	23
4.2 <i>Background Subtraction dan Image Segmentation</i> .....	24
4.3 Penentuan ROI .....	26
4.4 Nilai Intensitas pada ROI .....	27
4.5 <i>Queue Length Estimation</i> .....	28
<b>BAB V PENUTUP</b> .....	<b>31</b>
5.1 Kesimpulan .....	31
5.2 Saran .....	31
<b>DAFTAR PUSTAKA</b> .....	<b>33</b>
<b>BIOGRAFI PENULIS</b> .....	<b>35</b>

## DAFTAR GAMBAR

2.1 Konsep <i>Smart Traffic Light</i> menggunakan kamera .....	5
2.2 <i>Background reconstruction</i> .....	6
2.3 <i>Background subtraction</i> .....	8
2.4 Dilasi .....	10
2.5 Erosi .....	10
3.1 Diagram blok sistem keseluruhan .....	16
3.2 Proses deteksi antrean kendaraan .....	20
3.3 Penghitungan panjang antrean kendaraan .....	21
4.1 <i>Background reconstruction</i> menggunakan 100 frame .....	23
4.2 <i>Background reconstruction</i> menggunakan 200 frame .....	23
4.3 <i>Background reconstruction</i> menggunakan 300 frame .....	24
4.4 Segmentasi citra menggunakan 100 frame .....	24
4.5 Segmentasi citra menggunakan 200 frame .....	25
4.6 Segmentasi citra menggunakan 300 frame .....	25
4.7 ROI yang terdiri dari dua garis .....	26
4.8 Citra hasil segmentasi .....	27
4.9 Nilai intensitas di sepanjang garis A .....	27
4.10 <i>Frame</i> untuk pengujian <i>queue length estimation</i> .....	28

*Halaman ini sengaja dikosongkan*

## DAFTAR TABEL

4.1 Hasil penghitungan <i>queue length estimation</i> .....	29
---	----

*Halaman ini sengaja dikosongkan*

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Kemacetan lalu-lintas merupakan salah satu masalah yang banyak terjadi di kota-kota besar, khususnya di persimpangan yang dekat dengan pusat-pusat keramaian seperti kawasan industri, perkantoran, sekolah, dan sebagainya. Adanya pusat-pusat keramaian tersebut menyebabkan sebagian ruas jalan terisi oleh kendaraan dalam jumlah yang sangat banyak, sedangkan ruas jalan yang lain hanya terisi oleh kendaraan dalam jumlah yang sedikit. Kepadatan yang berbeda tersebut memerlukan penanganan khusus agar lalu-lintas di semua ruas dapat berjalan dengan lancar dan seimbang. Sementara itu, sistem lampu lalu-lintas yang ada di Indonesia masih menggunakan sistem pewaktuan yang tidak adaptif terhadap tingkat kepadatan lalu-lintas di tiap ruas jalan. Akibatnya, terjadi ketidakseimbangan antrean kendaraan pada persimpangan tersebut. Lalu-lintas di sebagian ruas berjalan lancar, sedangkan di ruas lainnya terjadi penumpukan dan antrean yang panjang karena lamanya lampu hijau menyala tidak sebanding dengan banyaknya kendaraan pada ruas tersebut.

Permasalahan tersebut dapat dikurangi, misalnya dengan menerapkan sistem *Smart Traffic Light*. *Smart Traffic Light* adalah sistem lampu lalu-lintas cerdas yang dapat mengatur pewaktuan pada lampu lalu-lintas secara adaptif berdasarkan tingkat kepadatan lalu-lintas pada masing-masing ruas jalan. Dengan sistem ini, ruas yang memiliki tingkat kepadatan lebih tinggi akan diberi durasi yang lebih lama untuk menyalakan lampu hijau, dibandingkan ruas lain yang tingkat kepadatannya lebih rendah. Dengan demikian, akan terjadi keseimbangan dan kelancaran lalu-lintas pada semua ruas jalan di persimpangan tersebut.

### **1.2 Permasalahan**

Permasalahan yang dibahas dalam tugas akhir ini adalah sebagai berikut:

1. Pewaktuan yang tidak adaptif pada lampu lalu-lintas di suatu persimpangan menyebabkan terjadinya ketidakseimbangan panjang antrean antara ruas jalan yang satu dengan ruas jalan yang lain.
2. Pendekatan penghitungan jumlah kendaraan yang melintas[1] tidak dapat diterapkan dalam keadaan kendaraan berhenti, sehingga diperlukan pendekatan lain yang dapat mengatasi masalah tersebut.

### **1.3 Tujuan**

Dari latar belakang dan permasalahan yang ada, maka dirumuskan tujuan dari tugas akhir ini, yaitu:

1. Membuat sistem pewaktuan lampu lalu-lintas yang adaptif berdasarkan tingkat kepadatan setiap ruas jalan di suatu persimpangan, sehingga terjadi keseimbangan dan kelancaran lalu-lintas pada semua ruas jalan tersebut.
2. Mengimplementasikan metode *queue length estimation* pada sistem *Smart Traffic Light*, sehingga identifikasi kepadatan lalu-lintas dapat menggambarkan keadaan yang sebenarnya.

### **1.4 Batasan Masalah**

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Masukan yang digunakan adalah video, yang berasal dari *Traffic IP Camera*.
2. Pendekatan yang digunakan untuk identifikasi kepadatan adalah menghitung *queue length estimation*.
3. Bagian persimpangan yang dianalisis hanya satu ruas jalan.
4. Keluaran yang dihasilkan adalah panjang antrean, dalam satuan meter.

### **1.5 Sistematika Penulisan**

Laporan penelitian tugas akhir ini disusun dengan sistematika sebagai berikut:

1. Bab I Pendahuluan  
Bab ini berisi uraian tentang latar belakang, permasalahan, tujuan penelitian, batasan masalah, dan sistematika penulisan laporan.
2. Bab II Tinjauan Pustaka  
Bagian ini menguraikan teori-teori yang berkaitan dengan permasalahan yang diteliti, antara lain tentang sistem *smart traffic*

*light, traffic IP camera*, visi komputer, pengolahan citra, dan sebagainya.

3. Bab III Desain dan Implementasi Sistem  
Bab ini berisi blok diagram dari sistem yang dirancang, beserta langkah-langkah yang dilakukan dalam penelitian ini.
4. Bab IV Pengujian dan Analisis  
Bagian ini memberikan penjelasan tentang pengujian terhadap sistem yang telah dirancang pada bab sebelumnya, serta analisis terhadap sistem tersebut.
5. Bab V Penutup  
Bab ini berisi kesimpulan akhir dari penelitian yang telah dilakukan, dan saran untuk penelitian selanjutnya yang berkaitan dengan penelitian ini.

### **1.6 Relevansi**

Penelitian tentang *adaptive traffic light* ini dapat berguna untuk menguraikan kemacetan di kota-kota besar, terutama di persimpangan yang kepadatannya tidak sama antara ruas jalan yang satu dengan ruas lainnya, sehingga penelitian ini relevan untuk diteliti lebih lanjut kemudian diimplementasikan secara nyata.

*Halaman ini sengaja dikosongkan*

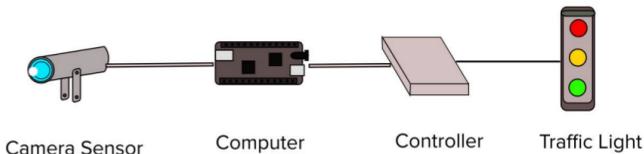
## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 *Smart Traffic Light*

*Smart Traffic Light*, yang juga dikenal dengan *Intelligent Traffic Light* atau *Advanced Traffic Light* adalah sistem lampu lalu lintas terintegrasi yang berfungsi untuk mengatur alur lalu lintas berdasarkan informasi yang diterima sehingga dapat beradaptasi terhadap kondisi jalan pada saat itu. *Smart Traffic Light* ditujukan untuk mengatasi kemacetan pada perempatan jalan karena waktu berhenti yang tidak efisien [3].

#### Smart Traffic Light



**Gambar 2.1** Konsep *Smart Traffic Light* menggunakan kamera

Secara umum, sistem ini terdiri dari kamera, komputer, controller, dan lampu lalu-lintas. Sensor kamera menangkap video kendaraan yang melintas lalu mengirimkan data tersebut ke sebuah komputer yang berfungsi untuk melakukan pengolahan citra hingga menghasilkan perhitungan yang dibutuhkan. Hasil akhir dari proses tersebut akan dikirimkan ke *controller* lampu lalu-lintas sehingga lampu akan menyala sesuai dengan keadaan lalu-lintas di persimpangan tersebut.

#### 2.2 *Traffic IP Camera*

*Traffic IP Camera* merupakan sebuah *IP Camera* yang dilengkapi kemampuan *video analytic*. Kamera ini biasanya digunakan untuk pengawasan pada jalan-jalan yang membutuhkan pengawasan yang

khusus, seperti jalan bebas hambatan, persimpangan jalan raya, parkir dan lain-lain. *Video analytic* pada *Traffic IP Camera* ini ditujukan untuk melakukan proses analisis terhadap kondisi jalan. Beberapa contoh kemampuan *Traffic IP Camera*, misalnya dalam masalah *Line Crossing*, *Path Tracking*, atau deteksi objek asing [1].

### 2.3 Background Reconstruction

*Background reconstruction* merupakan suatu teknik untuk mendapatkan citra latar (*background image*) dari sejumlah *frame* dalam suatu video. Citra latar yang didapatkan dari proses ini biasanya digunakan dalam proses lain yaitu *background subtraction*.

Proses *background reconstruction* dilakukan dengan mengekstrak sejumlah *frame* dari suatu video, kemudian mengkonversi *frame* atau citra tersebut menjadi citra *grayscale*. Setelah itu, dilakukan rata-rata berjalan (*running average*) pada citra tersebut sehingga dihasilkan citra latar yang dikehendaki. Semakin banyak jumlah *frame* yang digunakan maka citra latar yang dihasilkan akan semakin jelas, tetapi juga membutuhkan waktu yang lebih lama.



**Gambar 2.2** *Background reconstruction*: (a) Salah satu *frame* dari sebuah video, (b) Citra latar hasil rekonstruksi

*Background subtraction* merupakan teknik yang umum digunakan oleh para peneliti dalam bidang *computer vision*. *Background reconstruction* merupakan salah satu proses utama dalam *background subtraction*. *Background reconstruction* yang bagus merupakan hal yang esensial untuk menghasilkan *background subtraction* yang efektif.

Ada banyak penelitian yang berfokus pada *adaptive background reconstruction*. N.A. Zainuddin dkk [8] menyebutkan beberapa di antaranya, yaitu *temporal smoothing*, *pixel intensity classification*,

*running Gaussian average, Gaussian mixture model, hidden Markov model, dan kernel density estimation.*

Gaussian *mixture model* (GMM) merupakan sebuah metode yang umum digunakan oleh para peneliti dalam hal deteksi *foreground*. Gagasan utama dari Gaussian *mixture model* adalah dengan mendapatkan fungsi distribusi Gaussian sebanyak  $N$ , untuk merekonstruksi *background pixel*. Implementasi GMM telah dikemukakan sebagai teknik *adaptive background reconstruction*, khususnya pada sistem pengawasan (*surveillance system*) sejak akhir 1990.

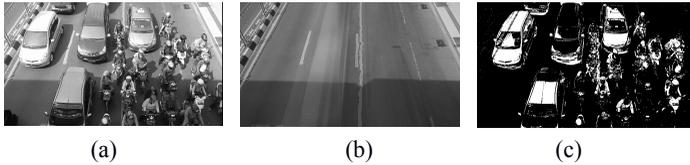
Elgammal dkk [9] mengklaim bahwa GMM hanya ideal untuk citra dalam ruangan (*indoor*). Oleh karena itu, dikenalkan penggunaan *kernel density estimation* (KDE) untuk *background reconstruction*. Pada metode KDE, piksel *background* dan piksel *foreground* merupakan model dari *probability density function* (pdf), dan pdf tersebut diestimasi oleh sebuah *kernel function* atau dikenal juga dengan *window function*. Akan tetapi, masalah utama pada metode ini adalah beban komputasi.

Metode lain yang umum digunakan oleh para peneliti dalam merekonstruksi citra latar adalah *temporal smoothing*. Implementasi metode ini pada citra lalu-lintas telah dilakukan sejak 1993. Gagasan utama pada temporal smoothing adalah mengkombinasikan citra yang tersimpan (*stored image*) dengan citra baru dengan berbasis pada piksel. Piksel-piksel pada smoothed image akan ditimpa oleh sebuah bagian dari nilai sebelumnya, dikombinasikan dengan sebuah nilai baru dari citra baru pada posisi yang sama.

Hou dkk [10] telah memperkenalkan metode *pixel intensity classification* (PIC) sebagai teknik *background reconstruction*. Dalam pendekatan ini, pada setiap *frame*, perbedaan intensitas piksel dihitung. Kemudian klasifikasi dibuat berdasarkan perbedaan tersebut. Model *background* diasumsikan sebagai frekuensi tertinggi dalam nilai intensitas.

## **2.4 Background Subtraction**

*Background subtraction* merupakan proses pengurangan antara sebuah frame dari suatu video, dengan *background* yang telah didapatkan dari proses lain yaitu *background reconstruction*.



**Gambar 2.3** *Background subtraction*: (a) Frame yang akan dideteksi objeknya, (b) Citra latar, (c) Citra hasil background subtraction yang sudah dijadikan citra biner

*Background subtraction* dikenal juga dengan istilah *foreground detection* [11]. Gagasan utama dalam *background subtraction* adalah mendeteksi objek-objek yang bergerak, dari perbedaan antara *frame* sekarang (*present frame*) dan *frame* sebelumnya. Dalam kasus di mana citra latarnya bersifat tetap (*fixed*), objek-objek yang bergerak dapat dideteksi dari perbedaan antara *present frame* dan *fixed background*. Secara umum, *fixed background* tidak dapat diaplikasikan pada lingkungan *real time*.

Untuk melakukan *background subtraction*, sebuah *background model* harus dibuat terlebih dahulu. Kemudian objek dideteksi dengan mencari simpangan atau selisih untuk setiap *frame* baru dengan *background model* yang digunakan.

Ada beberapa macam teknik *background subtraction*, di antaranya adalah *frame difference*, *statistical difference*, dan Mixture of Gaussian (MoG). Metode paling dasar untuk mendeteksi objek bergerak dalam sebuah citra stasioner dikenal dengan istilah *frame differencing*. *Frame* sekarang (*current frame*) dikurangkan dari *past frame*, dan jika perbedaan nilai piksel untuk piksel yang diproses lebih besar daripada nilai *threshold T* maka piksel tersebut dianggap bagian dari *foreground*.

Teknik *statistical difference* memberikan hasil yang lebih baik pada citra yang mengandung *noise*, tetapi disertai dengan berkurangnya kecepatan komputasi. Untuk menghitung statistik, diperlukan sekumpulan data yang baik, misalnya sejumlah *frame* untuk inisiasi proses deteksi. Statistik dihitung di setiap titik pada  $N$  *past frames*. *Frame* tersebut kemudian dikurangkan dari *current frame*. Pada kasus data yang mengandung *noise*, digunakan sebuah metode standar yaitu menggunakan nilai tengah pada setiap titik. Kemudian, mirip dengan

metode *frame difference*, citra latar dikurangkan dari *current frame* dan dibandingkan dengan nilai *threshold* untuk menentukan piksel *foreground*.

Suatu Gaussian *mixture model* (GMM) berguna untuk pemodelan data jika data tersebut berasal dari salah satu kelompok yang berbeda-beda. Kelompok-kelompok tersebut bisa berbeda antara yang satu dengan yang lain, tetapi titik-titik data dalam suatu kelompok dapat dimodelkan dengan sebuah distribusi Gaussian.

Pada Mixture of Gaussian (MoG), model citra latar bersifat parametrik. Setiap lokasi piksel direpresentasikan oleh sejumlah fungsi Gaussian yang dijumlahkan bersama-sama untuk membentuk sebuah *probability distribution function* (pdf).

Nilai rata-rata pada setiap fungsi Gaussian dapat dianggap sebagai tebakan nilai piksel pada *frame* berikutnya. Dalam hal ini, diasumsikan bahwa piksel-piksel tersebut merupakan citra latar. Nilai bobot dan deviasi standar pada setiap distribusi Gaussian merupakan ukuran kepastian pada tebakan tersebut. Untuk menentukan apakah suatu piksel merupakan bagian dari citra latar, piksel tersebut dibandingkan dengan komponen-komponen Gaussian yang mengikutinya. Jika nilai pikselnya berada dalam rentang *threshold*, maka piksel tersebut dianggap sebagai bagian dari citra latar. Jika tidak, maka piksel tersebut merupakan *foreground*.

## **2.5 Binary Thresholding**

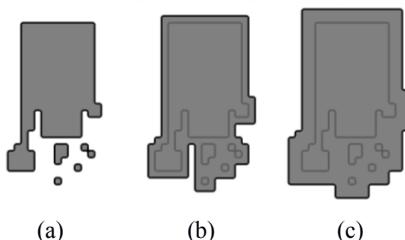
Merupakan jenis *thresholding* pada pengolahan citra digital yang mengonversi sebuah citra abu-abu (*grayscale*) menjadi citra biner (*binary image*). Citra biner adalah citra yang hanya memiliki dua macam nilai piksel, yaitu 0 dan 255. Nilai 0 merepresentasikan warna hitam, sedangkan nilai 255 merepresentasikan warna putih.

Proses binary thresholding dilakukan dengan cara membandingkan nilai piksel pada citra grayscale dengan suatu nilai *threshold*  $T$ . Jika nilai intensitas di suatu piksel kurang dari  $T$  maka nilai tersebut diubah menjadi 0 (hitam). Dan jika nilai intensitasnya lebih besar dari  $T$  maka nilai tersebut diubah menjadi 255 (putih). Dengan cara itu maka didapatkan sebuah citra biner dengan warna hitam-putih. Contoh citra biner dapat dilihat pada Gambar 2.3 (c).

## 2.6 Dilasi dan Erosi

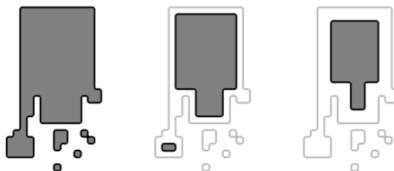
Dilasi (*dilation*) dan erosi (*erosion*) merupakan transformasi morfologis dasar, yang digunakan di berbagai macam konteks, seperti menghapus noise, mengisolasi elemen-elemen individual, dan menggabungkan elemen-elemen yang terpisah di dalam suatu citra [4].

Dilasi adalah konvolusi pada suatu citra dengan sebuah *kernel* di mana piksel-piksel pada area tersebut ditimpa atau digantikan oleh nilai maksimum dari semua nilai piksel yang dicakup oleh *kernel* tersebut.



**Gambar 2.4** Dilasi: (a) Piksel pada citra asli, (b) Piksel pada citra setelah dilasi sebanyak 1x, (c) Piksel pada citra setelah dilasi sebanyak 2x

Erosi adalah operasi yang berkebalikan dengan dilasi. Tindakan pada operator erosi sama dengan mengkomputasi sebuah nilai minimum di area yang dicakup oleh *kernel*.



**Gambar 2.5** Erosi: (a) Piksel pada citra asli, (b) Piksel pada citra setelah erosi sebanyak 1x, (c) Piksel pada citra setelah erosi sebanyak 2x

## 2.7 Traffic Management System

Sebagian besar *traffic management system* [2] saat ini dioperasikan dengan *pre-timed traffic lights*, yaitu lampu lalu-lintas yang telah ditentukan waktunya secara statis. Sistem ini tidak sensitif

terhadap perubahan-perubahan real time yang terjadi pada kondisi lalu-lintas.

Selain sistem yang statis, ada pula *traffic management system* yang bersifat dinamis. Sistem ini bisa menangani skenario yang berbeda-beda secara *real time*. Salah satu pendekatan yang dapat digunakan dalam sistem ini adalah memasang *inductive loop vehicle detector* di bawah permukaan jalan raya untuk mengukur atau menaksir jumlah lalu-lintas yang ada pada jalan tersebut. Metode tersebut termasuk metode yang bersifat intrusif, kompleks, mahal, dan penggunaannya terbatas pada area yang sempit pada jalan raya.

Pada perkembangan selanjutnya, *vision-based sensors* lebih banyak digunakan pada penerapan masalah lalu-lintas, karena bersifat non-intrusive, lebih hemat biaya dalam hal infrastruktur kamera dan penyebarannya, serta adanya dukungan berupa kecanggihan dalam teknologi pengolahan citra.

## **2.8 Queue Length Estimation**

*Queue length estimation* [2] merupakan suatu cara untuk mendapatkan perkiraan panjang antrean kendaraan di suatu ruas jalan pada persimpangan jalan raya. Panjang antrean pada sebuah persimpangan lalu-lintas memberikan informasi yang bermanfaat tentang tingkat kemacetan, dan merupakan sebuah parameter untuk *dynamic traffic light control*. Berdasarkan estimasi *current traffic queue length* di setiap ruas jalan pada persimpangan, lampu lalu-lintas dapat dijadwalkan sesuai dengan panjang antrean itu. Ruas jalan dengan antrean lalu-lintas lebih panjang dapat diberi lampu hijau untuk durasi yang lebih lama dan dapat diberi prioritas lebih tinggi. Dengan kata lain, sebuah *intelligent vision sensor* dapat menggantikan polisi lalu-lintas untuk mengerjakan tugas yang berupa mengelola lalu-lintas secara dinamis berdasarkan kondisi lalu-lintas saat itu.

M.Y. Siyal [5] mengemukakan sebuah metode untuk mengekstrak parameter-parameter antrean kendaraan yang akurat menggunakan pengolahan citra dan *neural network*. Metode ini memiliki kompleksitas komputasi yang tinggi disebabkan oleh penggunaan *neural network* di dalamnya, yang menjadikan metode ini kurang sesuai untuk *embedded system* yang *real-time* dan berbiaya rendah. Lebih jauh lagi, parameter antrean lalu-lintas berakurasi tinggi tidak terlalu

dibutuhkan untuk *dynamic traffic light control*. Hal ini disebabkan karena fungsionalitas dari *vision sensor* serupa dengan polisi lalu-lintas yang mendasarkan keputusannya dalam hal arus lalu-lintas, pada estimasi visualnya terhadap panjang antrean kendaraan, dan bukan pada panjang antrean yang sebenarnya maupun jumlah kendaraan yang sebenarnya.

R.K. Satzoda [2] mengemukakan sebuah teknik yang efisien secara komputasi untuk mengestimasi panjang antrean lalu-lintas di persimpangan, yang dapat membantu dalam pengaturan lalu-lintas dinamis (*dynamic traffic control*). Teknik yang dikemukakan tersebut, yang meliputi *progressive block processing* dan deteksi kendaraan menggunakan pendekatan dua-langkah (*two-step approach*), memberikan estimasi panjang antrean yang dapat diandalkan bahkan meskipun ada bayangan di permukaan jalan.

### **2.9 Progressive Block Processing**

*Progressive block processing* [2] merupakan sebuah metode untuk mengetahui panjang antrean lalu-lintas di suatu ruas jalan pada persimpangan, yang didasarkan pada pembagian jalan raya menjadi blok-blok dan memprosesnya secara progresif dimulai dari *stop line*. Kamera dipasang di persimpangan, menghadap ke jalan, dan posisinya diasumsikan tetap (*fixed*). *Region of interest* (ROI) dibatasi pada permukaan jalan saja, dengan demikian akan membuang daerah-daerah yang tidak diinginkan di sisi jalan pada citra, seperti pohon atau gedung. Ini juga dapat mengurangi ruang sampel yang akan diproses.

### **2.10 Penelitian Terkait**

#### **Menghitung Kendaraan Roda Empat**

Andi Muhammad Ali Mahdi Akbar [1] membangun sebuah sistem untuk menghitung kendaraan roda empat dengan metode luasan piksel. Dalam penelitian ini digunakan *IP Camera* untuk melakukan *streaming video* lalu-lintas. Video dikirimkan ke sebuah SBC (*single board computer*) untuk dilakukan proses pengolahan citra dengan ROI (*region of interest*) berbentuk segiempat yang ditempatkan pada setiap lajur pada jalan raya yang diteliti. Keluaran yang dihasilkan adalah jumlah kendaraan, yang dapat digunakan sebagai data untuk mengatur

lampu lalu-lintas di suatu persimpangan. Hasil dari penelitian ini menunjukkan bahwa sistem yang diajukan dapat menghasilkan perhitungan kendaraan roda empat dengan akurasi hingga 82,18 % pada siang hari dan 88,30 % pada malam hari.

### **Real-Time Vehicle Counter System**

Pada penelitian yang dilakukan oleh I.K.E. Purnama, A. Zaini, M. Hariadi, dan B.N. Putra [6] dibangun sebuah sistem penghitung kendaraan secara *real time* untuk diterapkan pada ITS (*Intelligent Transportation System*). Masukan yang digunakan dalam penelitian ini berupa video. Frame-frame dari video diproses dengan serangkaian prosedur yaitu *foreground extraction*, *object segmentation* dan *labeling*, serta *object classification* untuk membedakan antara sepeda motor dan mobil. Pada citra yang di dalamnya tidak terdapat bayangan dari objek-objek yang tidak diinginkan, sistem ini memberikan tingkat keberhasilan hingga 97 % untuk mengenali sepeda motor, dan 95 % untuk mengenali mobil.

### **Deteksi Kecepatan Kendaraan Bermotor**

Pribadi Hartoto [7] melakukan penelitian untuk membangun sistem yang dapat mendeteksi kecepatan kendaraan menggunakan metode *background subtraction* dan *frame difference*. Masukan yang digunakan berupa rekaman video lalu-lintas, kemudian dilakukan proses *background reconstruction* sehingga didapatkan citra latar, yang kemudian dijadikan referensi pada proses *foreground extraction*. Objek bergerak akan diukur perpindahannya dalam menempuh jarak tertentu, sebagai referensi perhitungan kecepatan. Akurasi hasil deteksi tidak mencapai 100 % jika dibandingkan dengan acuan speedometer pada kendaraan uji, namun cukup mewakili estimasi kecepatan dengan perbedaan maksimal 4,9 km/jam dan minimal 0,3 km/jam.

*Halaman ini sengaja dikosongkan*

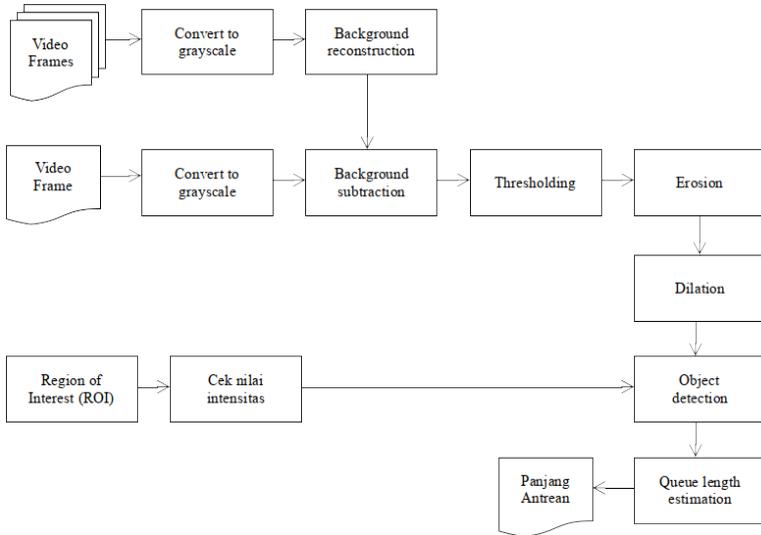
## **BAB 3**

# **DESAIN DAN IMPLEMENTASI SISTEM**

Penelitian ini dilaksanakan sesuai dengan desain sistem beserta implementasinya. Desain sistem merupakan konsep dan perancangan infrastruktur, kemudian diwujudkan dalam bentuk blok-blok alur yang harus dilakukan. Bagian implementasi merupakan pelaksanaan teknis untuk setiap blok pada desain sistem.

### **3.1 Desain Sistem**

Pada penelitian ini, kamera digunakan untuk mengambil data berupa rekaman video lalu-lintas di persimpangan jalan raya. Perekaman video dimulai saat lampu merah menyala hingga detik terakhir lampu tersebut menyala. Kemudian video tersebut diproses menggunakan program yang dirancang untuk mendeteksi objek berupa antrian kendaraan, kemudian melakukan perhitungan untuk mendapatkan estimasi panjang antrian kendaraan. Pemrosesan ini meliputi *background reconstruction*, *background subtraction*, *thresholding*, erosi, dilasi, deteksi objek, dan *queue length estimation*. Diagram blok untuk sistem ini dapat dilihat pada gambar 3.1.



**Gambar 3.1** Diagram blok sistem keseluruhan

Saat memproses video rekaman lalu-lintas, program akan melakukan rekonstruksi citra latar (*background reconstruction*) secara berkala untuk mendapatkan citra latar (*background*) yang nantinya akan digunakan dalam proses selanjutnya untuk mendeteksi kendaraan. Rekonstruksi tersebut perlu diulangi secara berkala karena kondisi jalan akan berubah sesuai waktu dan cuaca. Dengan demikian, *background* yang didapatkan akan selalu sesuai dengan kondisi terkini dari jalan raya yang diteliti.

Setelah didapatkan *background* dalam bentuk *grayscale image*, kemudian kondisi jalan pada detik terakhir lampu merah menyala di-capture untuk diproses pada tahap *background subtraction*. Sebelumnya, *frame* tersebut harus diubah terlebih dahulu ke dalam bentuk *grayscale image* agar dapat diproses. Prosesnya berupa menghitung nilai perbedaan absolut antara *frame* tersebut dengan *background* yang telah didapatkan sebelumnya. Dari perhitungan tersebut akan didapatkan *foreground image* berupa antrean kendaraan, dalam bentuk *grayscale* pula.

Selanjutnya, dilakukan segmentasi citra, yang meliputi *binary thresholding*, erosi, dan dilasi. Proses *binary thresholding* terhadap *foreground image* akan menghasilkan citra biner (*binary image*). Kemudian dilakukan proses erosi dan dilasi. Segmentasi tersebut berfungsi untuk mengurangi *noise* berupa piksel-piksel putih yang bukan merupakan kendaraan, sekaligus memperjelas piksel-piksel putih yang terdapat pada objek yang akan dideteksi, yaitu berupa kendaraan.

Dari hasil segmentasi di atas, program kemudian melakukan identifikasi piksel-piksel putih yang berada pada *region of interest* (ROI). ROI yang digunakan pada penelitian ini adalah ROI berupa garis, yang diletakkan pada 2 lajur kendaraan di jalan raya. Sedangkan lajur paling kanan dari kamera tidak termasuk dalam ROI karena umumnya lajur tersebut dilalui oleh kendaraan yang belok kiri dan tidak berhenti pada saat lampu merah menyala.

### **3.2 Alur Kerja**

Penelitian ini dilakukan dengan menempuh beberapa tahap, yaitu sebagai berikut:

1. *Background reconstruction*
2. *Background subtraction*
3. *Thresholding*
4. Erosi
5. Dilasi
6. Deteksi objek
7. Penghitungan *queue length estimation*

### **3.3 Background Reconstruction**

Pada tahap ini, program akan membaca sejumlah *frame* dari video uji. Hasil dari proses ini adalah sekumpulan citra dalam bentuk RGB. Kemudian dilakukan proses konversi dari citra RGB menjadi citra grayscale. Setelah itu, dilakukan proses rata-rata berjalan (*running average*) pada citra tersebut sehingga dihasilkan citra latar atau *background image*.

Jumlah *frame* yang digunakan dalam proses ini adalah 300 *frame* agar terdapat cukup banyak data untuk menghasilkan citra latar yang cukup jelas atau halus.

Citra latar yang didapatkan dari proses ini akan disimpan untuk digunakan pada tahap selanjutnya yaitu *background subtraction*.

### **3.4 Background Subtraction**

Tahap ini diawali dengan meng-*capture* suatu *frame* dari video uji, yang akan dideteksi objeknya. Citra yang didapatkan dari proses *capture* tersebut berupa citra RGB, kemudian dilakukan konversi ke dalam bentuk citra *grayscale*. Setelah didapatkan citra *grayscale*, kemudian dilakukan proses pengurangan antara citra tersebut dengan citra latar yang telah didapatkan dari tahap *background reconstruction*. Proses ini berupa menghitung nilai *absolute difference* antara *frame* tersebut dengan citra latar. Dengan proses tersebut, akan didapatkan sebuah citra *grayscale* yang berisi objek yang dikehendaki, yang disebut juga dengan *foreground image*.

### **3.5 Thresholding**

Pada tahap ini, dilakukan proses *binary thresholding* pada *foreground image* yang telah didapatkan dari tahap *background subtraction*. Nilai *threshold* yang digunakan dalam penelitian ini adalah 60. Piksel-piksel yang nilainya kurang dari 60 akan dijadikan 0 (nol), sedangkan piksel-piksel yang nilainya lebih dari 60 akan dijadikan 255. Nilai 0 berarti hitam, sedangkan nilai 255 berarti putih. Dengan proses *binary thresholding* ini maka akan didapatkan citra biner (*binary image*) yang berwarna hitam-putih.

### **3.6 Erosi**

Proses erosi yang diterapkan pada penelitian ini menggunakan *structuring element* yang berupa segiempat (*rectangle*) dengan ukuran 3x3. Ukuran yang digunakan tidak terlalu besar agar tidak terlalu banyak piksel yang hilang setelah proses erosi dilakukan. Sedangkan perulangan yang diterapkan adalah sebanyak 2x. Dengan adanya proses erosi maka *noise* berupa piksel-piksel putih yang tersebar di dalam citra dapat dihilangkan atau dikurangi.

### 3.7 Dilasi

Proses dilasi yang diterapkan pada penelitian ini menggunakan structuring element yang berupa segiempat (rectangle) dengan ukuran  $5 \times 5$ . Ukuran yang digunakan lebih besar daripada ukuran structuring element pada erosi agar objek yang dikehendaki semakin besar atau jelas. Sedangkan perulangan yang diterapkan adalah sebanyak  $2x$ .

Setelah tahap ini maka didapatkan sebuah citra yang tersegmentasi (*segmented image*). Citra inilah yang akan diproses lebih lanjut berdasarkan ROI (*region of interest*) yang telah ditetapkan.

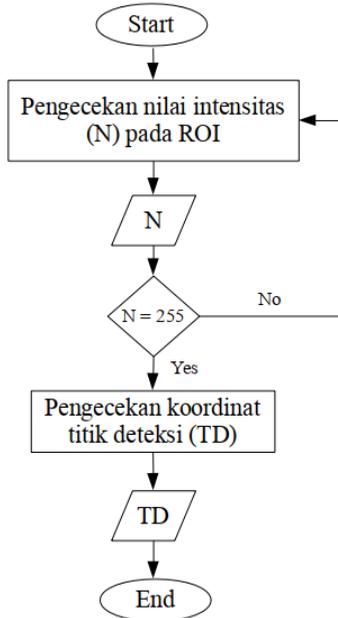
### 3.8 Deteksi Objek

Deteksi objek berupa antrean kendaraan dilakukan dengan memproses ROI (*region of interest*) yang berupa garis pada lajur kanan dan tengah dari ruas jalan yang diamati. Garis pada ROI ditetapkan dengan cara menentukan titik awal  $(x_1, y_1)$  dan titik akhir  $(x_2, y_2)$ . Dari kedua titik tersebut, didapatkan persamaan garis:

$$y = \frac{(y_2 - y_1)(x - x_1)}{(x_2 - x_1)} + y_1 \quad (3.1)$$

Dengan menggunakan persamaan garis seperti pada persamaan (3.1) maka didapatkan *array* berisi pasangan nilai-nilai  $x$  dan  $y$  di sepanjang garis pada ROI. Kemudian dilakukan pengecekan nilai intensitas di sepanjang garis tersebut, apakah bernilai 0 atau 255. Pengecekan dimulai dari titik di ujung belakang dari antrean kendaraan. Apabila bernilai 0 (hitam), artinya belum terdeteksi adanya kendaraan. Maka pengecekan dilanjutkan ke titik di depannya. Demikian seterusnya hingga ditemukan adanya titik bernilai 255 (putih) yang merepresentasikan adanya objek berupa kendaraan. Apabila telah ditemukan titik putih maka pengecekan dihentikan. Hal ini dilakukan karena objeknya adalah antrean kendaraan sehingga titik-titik di depannya dapat dianggap berwarna putih dan ada kendaraan di titik-titik tersebut.

Diagram alir untuk proses deteksi objek berupa antrean kendaraan dapat dilihat pada Gambar 3.2.



**Gambar 3.2** Proses deteksi antrean kendaraan

Keterangan:

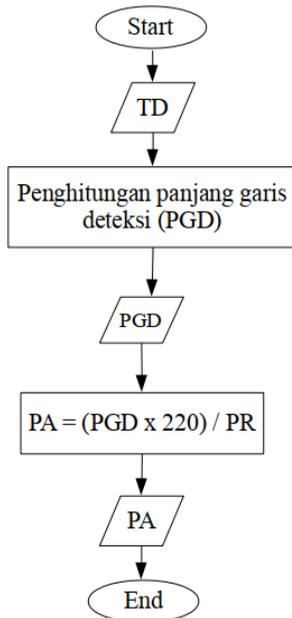
N = nilai intensitas titik-titik di sepanjang ROI

TD = koordinat (x, y) titik deteksi

### 3.9 Penghitungan *Queue Length Estimation*

Setelah ditemukan titik putih pada tahap deteksi objek, maka dilakukan perhitungan banyaknya titik mulai dari titik awal (bawah) pada ROI hingga titik yang dideteksi (TD). Kemudian banyaknya titik tersebut dikonversi ke dalam meter berdasarkan rasio antara panjang ROI pada citra dengan panjang sebenarnya di lapangan. Hasil konversi tersebut adalah berupa panjang antrean kendaraan, dalam satuan meter.

Diagram alir untuk proses penghitungan queue length estimation dapat dilihat pada Gambar 3.3.



**Gambar 3.3** Penghitungan panjang antrean kendaraan

Keterangan:

TD = koordinat (x, y) titik deteksi

PGD = panjang garis deteksi, dari titik terdepan sampai TD

PR = panjang ROI pada citra

220 = panjang ROI sebenarnya

PA = panjang antrean kendaraan (meter)

*Halaman ini sengaja dikosongkan*

## BAB 4

### PENGUJIAN DAN ANALISIS

#### 4.1 *Background Reconstruction*

Pengujian *background reconstruction* dilakukan dengan membedakan jumlah frame yang diproses, untuk mendapatkan *background*. Tujuan dari pengujian ini adalah untuk mendapatkan citra latar yang paling jelas, yang nantinya akan digunakan pada proses selanjutnya. Hasilnya adalah sebagai berikut:



**Gambar 4.1** *Background reconstruction* menggunakan 100 frame



**Gambar 4.2** *Background reconstruction* menggunakan 200 frame



**Gambar 4.3** *Background reconstruction* menggunakan 300 frame

Dari Gambar 4.1, 4.2, dan 4.3 dapat diketahui bahwa citra latar yang paling jelas adalah citra hasil rekonstruksi menggunakan 300 frame.

#### **4.2 Background Subtraction dan Image Segmentation**

Pengujian ini dilakukan untuk mengetahui seberapa baik citra latar (*background image*) yang didapatkan pada tahap sebelumnya, untuk digunakan sebagai *background* sehingga menghasilkan *foreground image* yang paling jelas. Dengan demikian, deteksi objeknya juga menjadi lebih akurat.



**Gambar 4.4** Segmentasi citra menggunakan 100 frame



**Gambar 4.5** Segmentasi citra menggunakan 200 frame



**Gambar 4.6** Segmentasi citra menggunakan 300 frame

Dari hasil pengujian pada Gambar 4.4, 4.5, dan 4.6 tampak bahwa penggunaan 100 frame dalam *background reconstruction* akan menghasilkan citra tersegmentasi (*segmented image*) yang kurang jelas pada objek yang dikehendaki. Pada penggunaan 200 frame, citra akhir yang didapatkan tampak lebih jelas pada bagian objek yang dikehendaki. Pada penggunaan 300 frame, citra akhir yang dihasilkan tidak jauh berbeda dengan penggunaan 200 frame.

### 4.3 Penentuan *Region of Interest* (ROI)

Tujuan dari pengujian ini adalah untuk mengetahui posisi koordinat yang paling tepat untuk menempatkan ROI sehingga identifikasi kendaraan juga lebih tepat.

ROI yang digunakan pada penelitian ini adalah berupa 2 garis yang berada di lajur tengah dan kiri dari kamera, seperti tampak pada Gambar 4.7 berikut. Sedangkan video uji yang digunakan memiliki resolusi 1024 x 600 piksel.



**Gambar 4.7** ROI yang terdiri dari dua garis

Dari hasil pengujian program, didapatkan posisi koordinat ROI yang paling tepat adalah sebagai berikut:

- Garis A (merah) = (495, 100) hingga (390, 590)
- Garis B (hijau) = (513, 100) hingga (610, 590)

Sedangkan panjang sebenarnya dari ROI tersebut adalah 220 meter, yang didapatkan dengan cara melakukan pengukuran langsung di lapangan.

Setelah dilakukan segmentasi, citra pada Gambar 4.7 akan menjadi seperti Gambar 4.8.

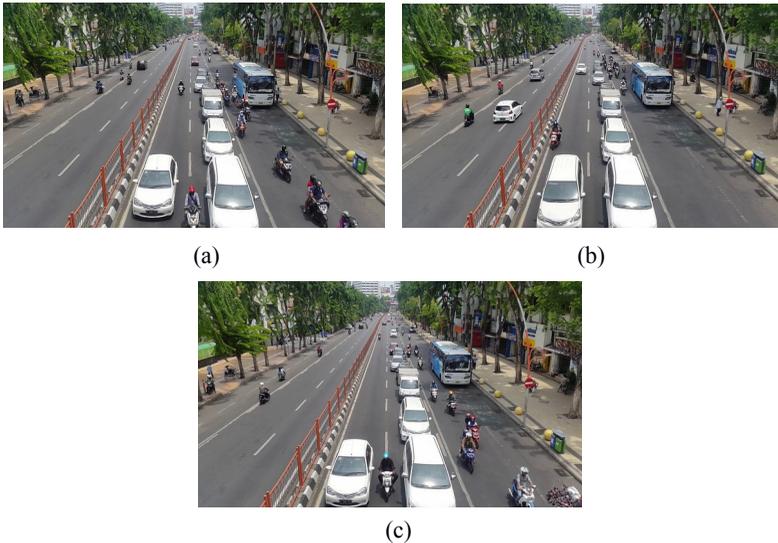


mobil. Sedangkan pada bagian belakang garis A tidak terdapat objek kendaraan sehingga titik-titiknya berwarna hitam, yaitu titik-titik dengan nilai intensitas 0 (nol).

#### 4.5 *Queue Length Estimation*

Berdasarkan pengecekan nilai intensitas titik-titik di sepanjang ROI sebagaimana tampak pada Gambar 4.9 yang pengecekannya dimulai dari titik di ujung belakang, kemudian program memeriksa koordinat dari titik putih yang terdeteksi, serta menghitung panjang garis mulai dari titik awal antrean (bawah) hingga titik putih tersebut, kemudian nilainya dikalikan dengan 220 meter yang merupakan jarak sebenarnya.

Pengujian ini dilakukan pada 3 frame yang berbeda, sebagaimana tampak pada gambar 4.10.



**Gambar 4.10** *Frame untuk pengujian queue length estimation*

Hasil penghitungan untuk garis A (merah) dan garis B (hijau) dapat dilihat pada Tabel 4.1.

**Tabel 4.1** Hasil penghitungan *queue length estimation*

Pengujian	Perhitungan Program		Perhitungan Manual	
	ROI A (m)	ROI B (m)	ROI A	ROI B
Pengujian 1	169	220	35,2	220
Pengujian 2	192	217	48,4	167,2
Pengujian 3	215	217	158,4	167,2

Dengan membandingkan data yang dihasilkan melalui program dan melalui cara manual sebagaimana pada Tabel 4.1, dapat diketahui bahwa estimasi panjang antrian pada penelitian ini belum cukup akurat, disebabkan karena citra yang digunakan bersifat perspektif. Dibutuhkan tambahan teknik atau metode lain agar hasil estimasinya lebih mendekati panjang sebenarnya.

*Halaman ini sengaja dikosongkan*

## **BAB 5**

### **PENUTUP**

#### **5.1 Kesimpulan**

Dari penelitian ini, didapatkan kesimpulan bahwa estimasi panjang antrean dengan hanya melakukan perhitungan panjang garis, memberikan hasil yang kurang akurat jika diterapkan pada citra yang bersifat perspektif.

#### **5.2 Saran**

Berdasarkan penelitian dan kesimpulan yang didapatkan, maka diberikan saran untuk penelitian selanjutnya:

1. Menambahkan teknik lain untuk menghitung estimasi panjang antrean, di samping perhitungan titik yang telah diterapkan pada penelitian ini, agar hasilnya lebih akurat.
2. Merancang program dalam bentuk GUI (*Graphical User Interface*) agar lebih mudah dan interaktif untuk digunakan.

*Halaman ini sengaja dikosongkan*

## DAFTAR PUSTAKA

- [1] Andi Muhammad Ali Mahdi Akbar. *Traffic IP Camera untuk Menghitung Kendaraan Roda Empat Menggunakan Metode Luasan Piksel*. Institut Teknologi Sepuluh Nopember. 2016.
- [2] R. K. Satzoda, S. Suchitra, T. Srikanthan, dan J. Y. Chia. *Vision-based Vehicle Queue Detection at Traffic Junctions*. Nanyang Technological University.
- [3] M. Wiering, J. Van Veenen, J. Vreeken, dan A. Koopman. *Intelligent Traffic Light Control*. Institute of Information and Computing Sciences. Utrecht University. 2004.
- [4] Adrian Kaehler, dan Gary Bradsky. *Learning OpenCV, 2<sup>nd</sup> Edition*. O'Reilly Media, Inc. 2014.
- [5] M.Y. Siyal dan M. Fathy. "A Neural-vision Based Approach to Measure Traffic Queue Parameters in Real-time," Pattern Recognition Letters, vol. 20, no. 8, pp. 761-770, Aug 1999.
- [6] I.K.E. Purnama, A. Zaini, M. Hariadi, dan B.N. Putra. *Real Time Vehicle Counter System for Intelligent Transportation System*. Institut Teknologi Sepuluh Nopember. 2009.
- [7] Pribadi Hartoto. *Sistem Deteksi Kecepatan Kendaraan Bermotor pada Real Time Traffic Information System*. Institut Teknologi Sepuluh Nopember. 2009.
- [8] N.A. Zainuddin, Y.M. Mustafah, A.A. Shafie, M.A. Rashidan, dan N.N.A. Aziz. *Adaptive Background Reconstruction for Street Surveillance*. International Islamic University Malaysia. 2014.
- [9] A. Elgammal, R. Duraiswami, D. Harwood, dan L. Davis. "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance," Proceeding of the IEEE, vol. 90, no. 7, pp. 1151-1162, 2002.

- [10] Z. Hou dan C. Han. "*A Background Reconstruction Algorithm Based on Pixel Intensity Classification in Remote Video Surveillance System*," Proceedings of the Seventh International Conference on Information Fusion, vol. 2, 2004.
- [11] Kiran Pande, S.C. Yadav, dan Anurag Kumar Srivastava. "*Implementation and Analysis of Various Background Subtraction Techniques for IR Target Detection Using Different Filters*," International Conference on Intelligent Computing and Control Systems, 2017.

## BIOGRAFI PENULIS



Dwi Kuncoro dilahirkan pada 22 Agustus 1988 di Purworejo, Jawa Tengah. Penulis lulus dari SMP Negeri 20 Purworejo pada 2004 kemudian melanjutkan pendidikan ke SMA Negeri 1 Purworejo. Pada 2009, penulis melanjutkan pendidikan strata satu di Jurusan Teknik Elektro ITS Surabaya, bidang studi Teknik Komputer dan Telematika. Selama kuliah, penulis pernah aktif sebagai sekretaris umum Kalam (kajian Islam) Himatektro ITS. Penulis juga pernah aktif sebagai asisten direktur bidang bisnis di Koperasi Mahasiswa "Dr. Angka" ITS. Selain itu, penulis juga pernah mengikuti ajang perlombaan seperti PKM (Program Kreatifitas Mahasiswa) serta PMW (Program Mahasiswa Wirausaha) ITS bersama rekan-rekan sejurusan maupun lain jurusan.

*Halaman ini sengaja dikosongkan*