



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - K141502

# DETEKSI PERILAKU KENDARAAN RODA EMPAT DI JALAN RAYA MENGGUNAKAN OPENCV

David Victor Giandly  
NRP 5113100033

Dosen Pembimbing  
Prof. Ir. Joko Lianto Buliali, M.Sc., Ph.D.  
Victor Hariadi, S.Si, M.Kom.

DEPARTEMEN INFORMATIKA  
Fakultas Teknologi Informasi & Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018

*[Halaman ini sengaja dikosongkan]*



**TUGAS AKHIR - K141502**

# **DETEKSI PERILAKU KENDARAAN RODA EMPAT DI JALAN RAYA MENGGUNAKAN OPENCV**

David Victor Giandly  
NRP 5113100033

Dosen Pembimbing  
Prof. Ir. Joko Lianto Buliali, M.Sc., Ph.D.  
Victor Hariadi, S.Si., M.Kom.

DEPARTEMEN INFORMATIKA  
Fakultas Teknologi Informasi & Komunikasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018

*[Halaman ini sengaja dikosongkan]*



**FINAL PROJECT - K141502**

# **BEHAVIOUR DETECTION FOUR-WHEEL VEHICLE ON THE STREET USING OPENCV**

**David Victor Giandy**  
NRP 5113100033

**Supervisor**  
Prof. Ir. Joko Lianto Buliali, M.Sc., Ph.D.  
Victor Hariadi, S.Si., M.Kom.

**Department of Informatics**  
**Faculty of Information Technology & Communication**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2018**

*[Halaman ini sengaja dikosongkan]*

**LEMBAR PENGESAHAN**  
**DETEKSI PERILAKU KENDARAAN RODA**  
**EMPAT DI JALAN RAYA MENGGUNAKAN**  
**OPENCV**

**TUGAS AKHIR**

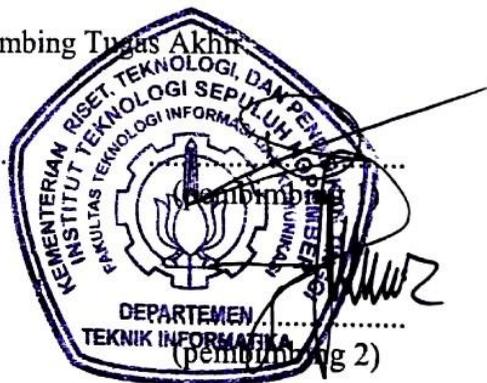
Diajukan Guna Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Manajemen Informasi  
Program Studi S-1 Departemen Informatika  
Fakultas Teknologi Informasi & Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh:  
**DAVID VICTOR GIANDLY**  
NRP: 5113 100 033

Disetujui oleh Dosen Pembimbing Tugas Akhir

Prof. Ir. Joko Lianto Buliali, M.Sc., Ph.D.  
NIP: 19670727 199203 1 002

Victor Hariadi, S.Si., M.Kom.  
NIP: 19691228 199412 1 001



**SURABAYA**  
**JANUARI 2018**

*[Halaman ini sengaja dikosongkan]*

# DETEKSI PERILAKU KENDARAAN RODA EMPAT DI JALAN RAYA MENGGUNAKAN OPENCV

Nama Mahasiswa : David Victor Giandly  
NRP : 5113 100 033  
Jurusan : Departemen Informatika FTIf- ITS  
Dosen Pembimbing 1 : Prof. Ir. Joko Lianto Buliali, M.Sc.,  
Ph.D.  
Dosen Pembimbing 2 : Victor Hariadi, S.Si., M.Kom.

## ABSTRAKSI

Dalam berkendara, banyak faktor yang dapat dilihat dari sebuah kendaraan saat berada di jalan dan hal yang diamati adalah kecepatan dan perpindahan lajur. Dari kecepatan dan perpindahan lajur kendaraan tersebut maka dapat dideteksi perilaku mengemudi sebuah kendaraan. Deteksi perilaku dapat dilakukan dari sebuah hasil rekaman yang dihasilkan oleh kamera pemantau lalu lintas yang terpasang di beberapa jalan. Hasil rekaman dari kamera pemantau lalu lintas yang digunakan dalam mendeteksi perilaku pengendara pada tugas akhir ini berupa file dengan format (.mp4). Pada tugas akhir ini, kendaraan yang akan diamati perilakunya hanya kendaraan roda empat saja. Tujuan Tugas Akhir ini yaitu untuk membuat sebuah perangkat lunak yang dapat mendeteksi perilaku kendaraan roda empat dengan video yang didapat dari kamera pemantau lalu lintas dengan bantuan *library* yaitu OpenCV dan menggunakan bahasa pemrograman C#.

Hasil rekaman dari kamera pemantau lalu lintas disimpan dalam format (.mp4) yang nantinya akan digunakan sebagai input dalam program. Pengambilan latar belakang dengan menggunakan metode background subtraction adalah tahap setelah video masuk ke dalam program dan digunakan untuk membantu dalam proses pendeteksian yang selanjutnya dilakukan proses penyimpanan gambar tiap frame dan pengambilan data XML. Dalam mendeteksi

kecepatan dilakukan dengan penggunaan rumus kecepatan yang ada dan untuk mengetahui perubahan lajur maka dilakukan pendeteksian dengan menyimpan tiap titik koordinat perpindahan tiap frame. Untuk melakukan kedua pendeteksian maka data XML diperlukan untuk melakukan proses tersebut.

Hasil dari pendeteksian dapat dilihat dalam file.txt yang menyimpan no frame, frame, jarak, kecepatan, perpindahan lajur, dan status kendaraan tersebut. Data XML yang tersimpan memengaruhi dalam proses pendeteksian kendaraan. XML yang tidak terdeteksi pada frame berikutnya akan menghentikan proses pendeteksian. Untuk mengetahui tingkat akurasi, maka dibagi menjadi tiga kondisi jalan, yaitu jalan dengan kondisi sepi, normal, dan ramai. Rata – rata nilai akurasi kecepatan pada kondisi jalan sepi 84,94%, jalan normal 75,27%, dan jalan ramai 91,22% sedangkan rata – rata nilai perpindahan lajur pada kondisi jalan sepi 86,67%, jalan normal 100%, dan jalan ramai 80%.

**Kata Kunci:** Background Subtraction, C#, Kamera Pemantau Lalu Lintas, Kecepatan, OpenCV

# Behaviour Detection Four-Wheel Vehicle On The Street Using OpenCV

Student Name : David Victor Giandly  
Student ID : 5113 100 033  
Major : Informatics Department FTIf-ITS  
Advisor 1 : Prof. Ir. Joko Lianto Buliali, M.Sc., Ph.D.  
Advisor 2 : Victor Hariadi, S.Si., M.Kom.

## ABSTRACT

*In driving, many factors can be seen from a vehicle while on the road and observed is speed and lane displacement. From the speed and lane displacement of the vehicle it can be detected driving behavior of a vehicle. Behavioral detection can be done from a recording produced by a traffic monitoring camera installed in some way. The recording of the traffic monitoring camera used in detecting the behavior of the rider in this final project is a file with .mp4 format. In this final project, the vehicle that will be observed its behavior is only four-wheeled vehicles only. The purpose of this Final Project is to create a software that can detect the behavior of four-wheeled vehicles with video obtained from traffic monitoring cameras with the help of OpenCV library and using C # programming language. The recording of the traffic monitoring camera is stored in the format (.mp4) which will be used as input in the program. Background retrieval using the background subtraction method is the step after the video enters the program and is used to assist in the detection process, which is then done in the image storage process of each frame and XML data retrieval. In detecting the speed is done by using the existing velocity formula and to know the change of lane then do the detection by storing each point coordinate the displacement of each frame. To do both detection then XML data is required to perform the process.*

*The results of the detection can be seen in file.txt which stores the no frame, frame, distance, speed, lane displacement, and vehicle status. The stored XML data affects the process of vehicle detection. Undetected XML in the next frame will stop detection. To determine the level of accuracy, then divided into three road conditions, namely the road with quiet conditions, normal, and crowded. The average speed accuracy value in the deserted road conditions is 84.94%, the normal road is 75.27%, and the road is crowded 91.22% while the mean of lane displacement rate in the deserted road conditions is 86.67%, the normal road is 100% and 80% crowded street.*

*Keywords: Background Subtraction, C #, Traffic Monitoring Camera, Speed, OpenCV*

## **KATA PENGANTAR**

Puji syukur kepada Tuhan Yesus Kristus atas segala kasih karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul:

### **DETEKSI PERILAKU KENDARAAN RODA EMPAT DI JALAN RAYA DENGAN OPENCV**

Melalui lembar ini, penulis ingin menyampaikan ucapan terimakasih dan penghormatan yang sebesar-besarnya kepada:

1. Ayah, Ibu, Adik dan keluarga besar yang selalu memberikan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Prof. Ir. Joko Lianto Buliali, M.sc., Ph.D., selaku dosen pembimbing 1 yang telah banyak membantu, membimbing, bahkan memotivasi penulis untuk menyelesaikan Tugas Akhir ini.
3. Bapak Victor Hariadi, S.Si., M.Kom. selaku dosen pembimbing 2 yang telah membimbing penulis dalam menyelesaikan Tugas Akhir ini.
4. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS lainnya yang telah banyak menyampaikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
5. Rekan – rekan gereja yang telah memberikan semangat dalam mengerjakan Tugas Akhir ini.
6. Rekan kerja yang telah memberikan semangat dan ijin penulis dalam bekerja dan mengerjakan Tugas Akhir ini.
7. Serta pihak-pihak lain yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari masih terdapat banyak kekurangan baik dalam pelaksanaan Tugas Akhir maupun penyusunan buku laporan ini, namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi. Penulis

mengharapkan kritik dan saran yang membangun untuk kesempurnaan penulisan buku Tugas Akhir ini.

Surabaya, Januari 2018

David Victor Giandly

# DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAKSI.....	ix
ABSTRACT.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
DAFTAR KODE SUMBER.....	xxi
DAFTAR PERSAMAAN.....	xxiii
1. BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Tujuan.....	2
1.3. Rumusan Permasalahan.....	2
1.4. Batasan Permasalahan.....	2
1.5. Metodologi.....	3
1.6. Sistematika Penulisan.....	5
2. BAB II DASAR TEORI.....	7
2.1. <i>Open CV</i> .....	7
2.2. <i>Background Subtraction</i> .....	7
2.3. <i>Blob Detection</i> .....	8
2.4. <i>Median Filter</i> .....	8
2.5. <i>Binary Threshold</i> .....	9
2.6. <i>Speed Detection</i> .....	10
3. BAB III ANALISIS DAN PERANCANGAN SISTEM.....	11
3.1. Analisis.....	11
3.1.1. Analisis Permasalahan.....	11
3.1.2. Analisis Kebutuhan.....	12
3.1.3. Deskripsi Umum Sistem.....	13
3.2. Perancangan Sistem.....	19
3.2.1. Perancangan Antar Muka Aplikasi.....	20

3.2.2.	Perancangan Alur Proses Penggunaan Aplikasi ..	23
4.	BAB IV IMPLEMENTASI.....	25
4.1.	Lingkungan Implementasi.....	25
4.1.1.	Lingkungan Implementasi Perangkat Keras.....	25
4.1.2.	Lingkungan Implementasi Perangkat Lunak.....	25
4.2.	Implementasi Kode.....	25
4.2.1.	Implementasi Input.....	26
4.2.2.	Implementasi Processing Video .....	26
4.2.3.	Implementasi Output .....	31
5.	BAB V PENGUJIAN DAN EVALUASI .....	33
5.1.	Lingkungan Pengujian.....	33
5.2.	Skenario Pengujian.....	33
5.2.1.	Skenario 1 .....	34
5.2.2.	Skenario 2.....	34
5.3.	Akurasi Pengujian .....	37
5.3.1.	Skenario 1 .....	38
5.3.2.	Skenario 2.....	41
5.4.	Evaluasi Pengujian .....	45
6.	BAB VI KESIMPULAN DAN SARAN.....	48
6.1.	Kesimpulan.....	48
6.2.	Saran.....	48
	DAFTAR PUSTAKA.....	50
	BIODATA PENULIS.....	51

## DAFTAR GAMBAR

Gambar 2-1 Background Subtraction.....	8
Gambar 2-2 Median Filter.....	9
Gambar 3-1 Alur Jalan Sistem .....	13
Gambar 3-2 Background Jalan .....	15
Gambar 3-3 Data Frame dalam Video.....	16
Gambar 3-4 Batas Wilayah Deteksi .....	16
Gambar 3-5 Pendeteksian Kendaraan .....	17
Gambar 3-6 Perpindahan Lajur .....	18
Gambar 3-7 Hasil Deteksi .....	19
Gambar 3-8 Perancangan Tampilan Awal Sistem.....	20
Gambar 3-9 Pengambilan Background Jalan .....	21
Gambar 3-10 Data XML .....	21
Gambar 3-11 Tampilan Saat Mendeteksi Kendaraan.....	22
Gambar 3-12 Hasil Ouput Program.....	23
Gambar 3-13 Workflow Alur Penggunaan Aplikasi .....	24
Gambar 5-1 Pengambilam Data XML <b>Error! Bookmark not defined.</b>	
Gambar 5-2 Kondisi Jalan Sepi.....	35
Gambar 5-3 Kondisi Jalan Normal.....	36
Gambar 5-4 Kondisi Jalan Ramai .....	36
Gambar 5-5 Data XML Sebagian.....	38
Gambar 5-6 Hasil Pendeteksian Data XML Sebagian .....	39
Gambar 5-7 Data XML Sebagian.....	39
Gambar 5-8 Hasil Pendeteksian Data XML Penuh .....	40
Gambar 5-9 Data XML Penuh .....	41
Gambar 5-10 Dimensi Ukuran Mobil .....	42
Gambar 5-11 Output dalam berkas file.txt.....	47

*[Halaman ini sengaja dikosongkan]*

## **DAFTAR TABEL**

Tabel 3.1 Kondisi Kecepatan & Perpindahan Lajur.....	12
Tabel 3.2 Kebutuhan Fungsional Sistem.....	12
Tabel 5.1 Hasil Deteksi Cara Menegemudi.....	41
Tabel 5.2 Hasil Perhitungan Keadaan Jalan Sepi.....	43
Tabel 5.3 Hasil Perhitungan Keadaan Jalan Normal.....	44
Tabel 5.4 Hasil Perhitungan Keadaan Jalan Ramai.....	45
Tabel 5.5 Rangkuman Hasil Pengujian Skenario 2.....	46

*[Halaman ini sengaja dikosongkan]*

## **DAFTAR KODE SUMBER**

Kode Sumber 4.1 Input Video.....	26
Kode Sumber 4.2 Memulai Pendeteksian .....	27
Kode Sumber 4.3 Implementasi Perubahan Lajur.....	28
Kode Sumber 4.4 Menghitung Kecepatan.....	30
Kode Sumber 4.5 Penentuan Kesimpulan Deteksi.....	30
Kode Sumber 4.6 Menampilkan Hasil Deteksi .....	31

*[Halaman ini sengaja dikosongkan]*

## **DAFTAR PERSAMAAN**

Persamaan 2.1 Perhitungan Kecepatan.....	10
Persamaan 5.1 Perhitungan Akurasi.....	42

*[Halaman ini sengaja dikosongkan]*

# **BAB I**

## **PENDAHULUAN**

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

### **1.1. Latar Belakang**

Saat berkendara, banyak pengendara yang melakukan pelanggaran saat mengemudikan kendaraannya, baik kecepatan, berpindah lajur, melanggar rambu-rambu lalu lintas, dan lain-lain yang dapat menyebabkan bahaya bagi pengendara lain dan juga bisa menyebabkan kecelakaan. Pengemudi yang melanggar disebabkan karena kesempatan, terburu – buru dan tidak ada petugas kepolisian yang menjaga. Dari permasalahan itu, pemerintah dan petugas kepolisian memikirkan cara termudah untuk memantau dan dapat mengerahkan anggota kepolisian ke wilayah – wilayah yang sering terjadi pelanggaran – pelanggaran ataupun terjadi kecelakaan. Saat ini di setiap jalan yang ada di setiap kota sudah terpasang sebuah kamera pengawas (CCTV) yang dipergunakan untuk memantau arus kendaraan dan kejadian – kejadian yang terjadi di suatu tempat sebagai salah satu tindakan untuk meningkatkan keamanan dan ketertiban pengemudi. Dengan adanya video tersebut, maka dapat dicari perilaku kendaraan roda empat tersebut berdasarkan kecepatan dan perpindahan lajurnya.

Hasil dari rekaman CCTV yang ada pada setiap jalan dapat digunakan sebagai sebuah informasi untuk mengetahui karakteristik cara mengemudi seseorang dalam mengemudikan kendaraannya. Dalam mengetahui karakteristik cara mengemudi, diperlukan 2 hal yang akan diamati yaitu perpindahan lajur dan kecepatan sebuah kendaraan. Kedua hal ini sangat penting dalam mengetahui karakteristik cara mengemudi seseorang karena merupakan sebuah aturan yang sudah ada dalam peraturan lalu lintas sampai saat ini dan sering dilakukan oleh pengendara.

Dengan informasi tersebut dapat dijadikan sebuah ukuran untuk menentukan golongan kendaraan termasuk didalam cara mengemudi dengan batas aman atau bahaya.

Pada kali ini, Tugas Akhir ini akan menjawab implementasi kebutuhan menegenai karakteristik cara mengemudi suatu kendaraan di jalan raya. Tujuan dari Tugas Akhir ini adalah membuat sebuah perangkat lunak yang dapat mendeteksi karakteristik cara mengemudi sebuah kendaraan di jalan berdasarkan kecepatan dan perubahan lajur kendaraan tersebut.

## **1.2. Tujuan**

Tujuan dari pembuatan tugas akhir ini adalah:

1. Membuat perangkat lunak yang dapat mendeteksi dan menggolongkan cara mengemudi seseorang saat kendaraan di jalan berdasarkan video yang didapat dari rekaman kamera CCTV.
2. Membuat kesimpulan mengenai cara mengemudi sebuah kendaraan berdasarkan perpindahan lajur yang terjadi dan kecepatan kendaraan dalam video.

## **1.3. Rumusan Permasalahan**

Rumusan masalah yang diangkat dalam tugas akhir ini antara lain:

1. Bagaimana cara mendeteksi kendaraan yang ada pada video?
2. Bagaimana cara mengetahui perpindahan lajur mobil yang ada pada video?
3. Bagaimana cara mendapatkan kecepatan mobil yang ada pada video?

## **1.4. Batasan Permasalahan**

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, antara lain:

1. Kendaraan yang dideteksi adalah sebuah mobil.

2. Bahasa Pemrograman yang digunakan adalah C#
3. Library yang digunakan adalah OpenCV.
4. Program bersifat offline
5. Video yang digunakan berformat .mp4.
6. Kondisi cuaca cerah dan siang hari.
7. Sudut pandang kendaraan tampak depan dari tengah jalan.
8. Karakteristik cara mengemudi dipengaruhi oleh dua hal yaitu perubahan lajur dan kecepatan kendaraan.

### **1.5. Metodologi**

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

a. Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi latar belakang pembuatan tugas akhir, rumusan masalah, batasan masalah, tujuan pembuatan, manfaat, metodologi hingga jadwal kegiatan pembuatan tugas akhir. Selain itu proposal tugas akhir ini memberikan ringkasan dari tugas akhir. Proposal tugas akhir juga berisi tinjauan pustaka yang digunakan sebagai referensi pembuatan tugas akhir ini.

b. Studi literature

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan program yaitu mengenai *OpenCV*, *Background Subtraction*, *Blob Detection*, *Median Filter*, *Binary Threshold*, dan *Speed Detection* yang digunakan untuk mendeteksi kecepatan kendaraan dan perpindahan lajur dalam video agar mendapatkan karakteristik cara mengemudi.

c. Analisis dan desain perangkat lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat

lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Serta dilakukan desain suatu sistem dan desain proses-proses yang ada.

#### d. Implementasi perangkat lunak

Implementasi perangkat lunak ini dibangun dengan sistem perangkat bergerak yang berbasis bahasa pemrograman *C#* dengan *library* menggunakan OpenCV.

#### e. Pengujian dan evaluasi

Pada tahapan ini dilakukan uji coba terhadap perangkat lunak yang telah dibuat. Pengujian yang dimaksud adalah pengujian fungsionalitas aplikasi yang dibangun. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Tahap ini dimaksudkan juga untuk mengevaluasi jalannya aplikasi, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

#### f. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

##### 1. Pendahuluan

- a. Latar Belakang
- b. Rumusan Masalah
- c. Batasan Tugas Akhir
- d. Tujuan
- e. Metodologi
- f. Sistematika Penulisan

2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

## **1.6. Sistematika Penulisan**

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

### **Bab I Pendahuluan**

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

### **Bab II Dasar Teori**

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini. Teori yang terkait adalah *OpenCV*, *Background Subtraction*, *Blob Detection*, *Median Filter*, *Binary Threshold*, dan *Speed Detection* yang digunakan untuk mendeteksi kecepatan kendaraan dan perubahan lajur yang dilakukan dalam video agar mendapatkan karakteristik cara mengemudi.

### **Bab III Analisis dan Perancangan Sistem**

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka aplikasi.

**Bab IV Implementasi**

Bab ini berisi implementasi dari perancangan dan implementasi fitur-fitur penunjang aplikasi.

**Bab V Pengujian dan Evaluasi**

Membahas tentang lingkungan pengujian, skenario pengujian, dan evaluasi pengujian setelah aplikasi selesai dikembangkan.

**Bab VI Kesimpulan dan Saran**

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

**Daftar Pustaka**

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

**Lampiran**

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

## **BAB II**

### **DASAR TEORI**

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir.

#### **2.1. *Open CV***

OpenCV (*Open Source Computer Vision Library*) adalah sebuah library yang digunakan untuk pengolahan citra dinamis secara *realtime*[1]. OpenCV memiliki lisensi BSD sehingga dalam penggunaannya baik untuk akademik atau non akademik pun dapat digunakan secara gratis. OpenCV dapat digunakan pada Windows, Android, Linux, iOS, OS X, FreeBSD, OpenBSD, Maemo, Blackberry 10 dan dengan bahasa pemrograman Python, Java, C, C++. Karena program yang dibuat dalam bahasa C# maka diperlukan sebuah *wrapper* .Net untuk Open CV yaitu Emgu CV. Emgu CV memiliki library yang sama dengan Open CV hanya saja Emgu CV lebih mudah digunakan dalam bahasa pemrograman C#[2].

#### **2.2. *Background Subtraction***

Background Subtraction adalah teknik di bidang pengolahan citra dan visi komputer dengan latar depan gambar yang diekstraksi untuk pemrosesan lebih lanjut[3]. Pendekatan yang digunakan pada Background Subtraction untuk mendeteksi benda bergerak pada suatu video atau kamera. Pendekatan ini dengan mendeteksi benda bergerak dari selisih frame dengan *background model*.



**Gambar 2.1 Background Subtraction**

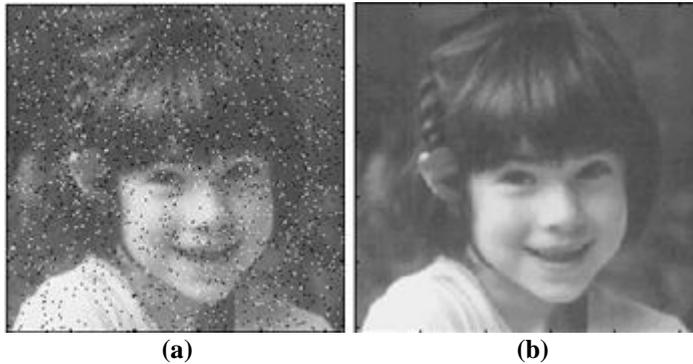
Pada Gambar 2.1, terdapat dua bagian. Bagian (a) merupakan gambar objek asli sedangkan bagian (b) merupakan gambar objek yang sudah mengalami *background subtraction*.

### 2.3. *Blob Detection*

Objek dalam bentuk dua atau tiga dimensi merupakan definisi dari Blob sedangkan Blob Detection merupakan kumpulan titik – titik pixel dengan warna yang berbeda dari latar belakang dan disatukan dalam suatu region[4]. Tujuan utama dari pendekatan ini untuk dapat menemukan daerah. Bentuk blob timbul dalam cara yang berbeda bergantung pada ukuran dan dapat dideteksi dengan menggunakan metode sederhana dalam representasi gambar (citation). Algoritma yang digunakan dalam Blob Detection ini adalah algoritma growing region. Algoritma growing region ini untuk menampilkan *image* sebagai matrix dengan nilai garis dan pixel yang sudah pasti.

### 2.4. *Median Filter*

Median Filter adalah teknik penyaringan yang digunakan untuk menghilangkan *noise*[5]. Pengurangan *noise* merupakan langkah awal yang berfungsi untuk memperbaiki hasil pemrosesan. Hasil dari median filter :



(a) (b)  
**Gambar 2.2 Median Filter**

Pada Gambar 2.2, terdapat 2 bagian yaitu bagian (a) merupakan gambar yang memiliki banyak bercak – bercak (*noise*) sedangkan bagian (b) merupakan gambar yang sama dan sudah diolah dengan median filter.

## 2.5. *Binary Threshold*

Dalam Open CV memiliki banyak fungsi dan salah satunya threshold. Threshold akan membuat gambar abu-abu menjadi gambar biner sesuai dengan ambang batas yang telah ditentukan. Target objek dapat ditentukan sebagai gelap atau terang. Threshold memiliki ambang batas atas dan bawah[6]. Sebenarnya banyak teknik yang bisa digunakan, tetapi pada program ini penulis menggunakan Binary Threshold.

**`cv2.threshold(img, 125, 255, cv2.THRESH_BINARY)`**

Angka 125 adalah ambang batas bawah sedangkan 255 adalah ambang batas atas. Open CV akan membaca setiap pixel gambar dan jika nilai pixel tersebut kurang dari 125 maka diubah menjadi 0 namun jika lebih besar dari 125 maka akan diubah menjadi 255.

## 2.6. *Speed Detection*

Kecepatan menurut Kamus Besar Bahasa Indonesia (KBBI) adalah waktu yang diperlukan untuk menempuh jarak tertentu[7]. Untuk mendapatkan kecepatan maka perlu mengetahui posisi kendaraan pada setiap frame. Persamaan berikut adalah rumus perhitungan kecepatan.

### **Persamaan 2.1 Perhitungan Kecepatan**

$$Kecepatan = \frac{\text{Jarak}}{\text{Waktu}}$$

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

Bab ini membahas tahap analisis dan perancangan sistem yang akan dibangun. Analisis membahas semua persiapan yang akan menjadi pokok pikiran pembuatan aplikasi ini. Mulai dari masalah yang melatarbelakangi, hingga analisis gambaran awal sistem yang akan dibuat. Perancangan sistem membahas hal-hal yang berkaitan dengan pondasi atau dasar pembuatan aplikasi, yang meliputi perancangan basis data, tampilan antar muka halaman aplikasi, hingga perancangan alur proses yang akan diimplementasikan di dalam aplikasi.

#### **3.1. Analisis**

Tahap analisis meliputi analisis masalah, analisis kebutuhan, deskripsi umum sistem, dan kasus penggunaan sistem yang dibuat.

##### **3.1.1. Analisis Permasalahan**

Di jalan, sudah banyak CCTV yang terpasang di setiap negara terutama Indonesia. CCTV yang ada hanya digunakan untuk memantau keadaan lalu lintas dan kejadian yang terjadi pada jalan tersebut. Oleh karena itu, aplikasi ini dibuat untuk menangani permasalahan di atas.

Dalam mengemudi, hal yang banyak dilakukan oleh pengendara adalah mempercepat kecepatan dan melakukan perpindahan lajur. Karena adanya sebuah hasil dari rekaman CCTV dan ingin mengetahui kedua hal tersebut maka terbentuklah aplikasi ini. Pada Tabel 3.1, kecepatan dibagi menjadi tiga kondisi yaitu pelan, normal dan cepat sedangkan untuk perpindahan lajur hanya memiliki dua kondisi yaitu aman dan bahaya seperti pada Tabel 3.1.

**Tabel 3.1 Kondisi Kecepatan & Perpindahan Lajur**

Kecepatan	Kondisi
0 km/jam – 40 km/jam	Pelan
41 km/jam – 60 km/jam	Normal
> 60 km/jam	Cepat

Perpindahan Lajur	Kondisi
0 – 1	Normal
>= 2	Bahaya

Dengan adanya dua hal utama dalam pendeteksian maka program ini dapat digunakan untuk menentukan mengetahui kecepatan dan perpindahan lajur yang dapat menggambarkan perilaku pengemudi saat berkendara.

### 3.1.2. Analisis Kebutuhan

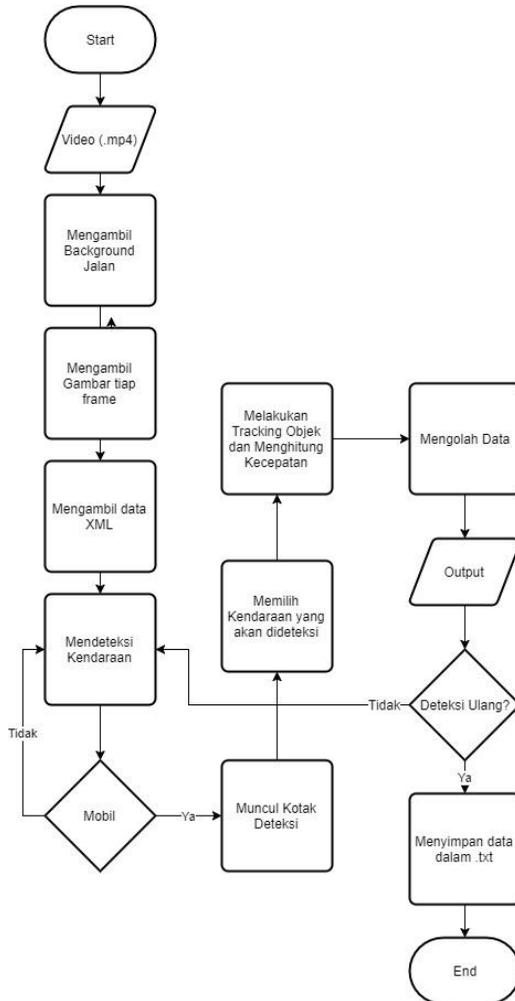
Kebutuhan utama dalam aplikasi ini difokuskan untuk mengetahui perilaku pengemudi saat berkendara pada video yang didapatkan pada CCTV yang terpasang pada jalan. Pada Tabel 3.2, Kebutuhan fungsional sistem adalah hal yang diperlukan dalam menentukan perilaku pengemudi berdasarkan

**Tabel 3.2 Kebutuhan Fungsional Sistem**

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-1	Mendeteksi kendaraan	Partisipan dapat mengetahui mobil yang terdeteksi dalam video
F-2	Mendeteksi perubahan lajur	Partisipan dapat mengetahui perpindahan lajur sebuah mobil yang terdeteksi dalam video
F-3	Mendeteksi kecepatan mobil yang terdeteksi	Partisipan dapat mengetahui kecepatan mobil yang terdeteksi dalam video.
F-4	Menentukan cara mengemudi	Partisipan dapat mendapatkan hasil karakteristik cara mengemudi pada mobil yang terdeteksi dalam video.

### 3.1.3. Deskripsi Umum Sistem

Aplikasi yang akan dibuat pada Tugas Akhir ini adalah program aplikasi perangkat bergerak. Gambar 3.1 adalah gambaran alur jalan sistem.



**Gambar 3.1 Alur Jalan Sistem**

Untuk penjelasan secara rinci akan dijelaskan di bawah mulai dari proses *input*, proses, hingga *output*.

### **3.1.3.1. *Input***

*Input* video ini didapat dari hasil perekaman CCTV yang dibutuhkan dalam program ini adalah video dengan input berformat MPEG Layer-4 Audio (.mp4). Video yang digunakan dalam kasus ini bernama video.mp4.

### **3.1.3.2. Proses**

Pada tahap ini dijelaskan secara bertahap langkah-langkah yang dilakukan pada sistem dari proses pengambilan background jalan, pengambilan XML tiap frame, hingga mendapatkan kecepatan dan perpindahan lajur dari kendaraan yang dideteksi yang nantinya akan didapatkan sebuah kesimpulan atau hasil *output*.

#### **3.1.3.2.1 Background Jalan**

Tahap ini merupakan tahap yang dilakukan sebelum melakukan ke tahap yang berikutnya, yaitu tahap proses. Pada tahap ini, video yang telah di masukkan dalam sistem akan mendeteksi background jalan dalam video tersebut yang akan digunakan sebagai model atau pembanding dari tiap frame yang dihasilkan dari video tersebut. Gambar 3.2 merupakan hasil dari proses awal memasukkan video dan terlihat bahwa tidak ada kendaraan yang ada didaerah deteksi nantinya. Untuk mendapatkan hasil tidak ada objek atau kendaraan diwilayah deteksi dalam background jalan ini, penulis menggunakan background subtraction dalam melakukan prosenya.



**Gambar 3.2 Background Jalan**

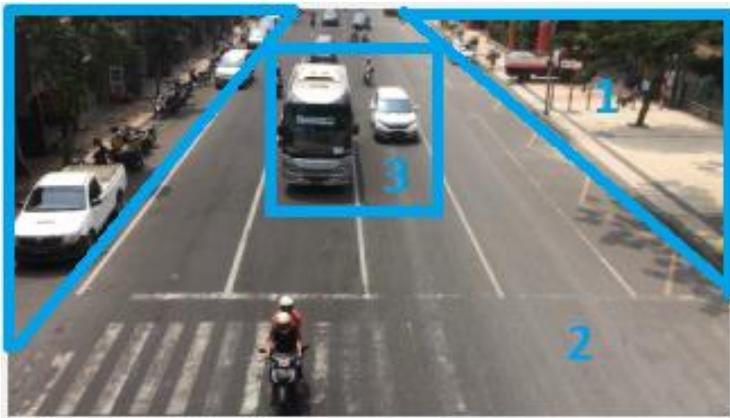
### **3.1.3.2.2 Pengambilan Data XML**

Jumlah gambar yang ada bergantung dari seberapa jumlah frame yang ada dalam satu file video. Pada Gambar 3.3 merupakan kumpulan frame yang didapatkan dari suatu video. Gambar – gambar inilah yang akan dijadikan sebuah acuan dalam menentukan perpindahan lajur tiap kendaraan dan pengambilan data xml untuk mendapatkan kecepatan kendaraan tersebut. Untuk mendeteksi objek maka diperlukan data XML. Data XML ini diambil dari pendeteksian yang dilakukan dari tiap frame tersebut yang kemudian informasi dari tiap frame disimpan didalam file XML. Data yang ada di file XML ini nantinya dapat digunakan untuk medeteksi objek di wilayah yang telah ditetapkan untuk dilakukan pendeteksian. Dalam tahap ini, diperlukan sebuah penetapan wilayah deteksi agar hanya jalan utama saja yang akan diteliti dan meningkatkan hasil dari proses perhitungan kendaraan tersebut.



**Gambar 3.3 Data Frame dalam Video**

Pada Gambar 3.4, wilayah yang akan fokuskan untuk diamati ada dibagian no 2. Wilayah dengan segitiga dibagian kiri dan kanan merupakan wilayah yang tidak akan dideteksi.



**Gambar 3.4 Batas Wilayah Deteksi**

Adanya penetapan wilayah deteksi ini bertujuan untuk memudahkan dalam perhitungan dan menghindari *error* yang akan terjadi pada sistem saat program ini dijalankan.

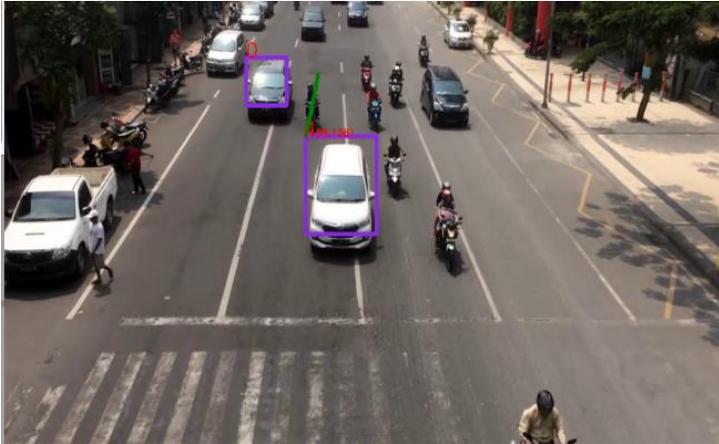
### 3.1.3.2.3 Pendeteksian Kecepatan dan Perpindahan Lajur

Pada tahap ini, adalah lanjutan dari bagian diatas. Setelah ditetapkan batasan wilayah yang akan diamati, tahap ini akan memulai pendeteksian kendaraan yang ada dalam video. Pendeteksian kendaraan dalam video ini dibantu dengan file xml yang sudah terisi dengan beberapa sample mobil. Dengan adanya file tersebut maka akan membantu proses pendeteksian mobil dan dengan hasil tersebut akan didapatkan kecepatan dari kendaraan tersebut dengan rumus yang sudah ditetapkan dan perpindahan lajur yang terjadi berdasarkan perubahan titik koordinat tiap frame.



**Gambar 3.5 Pendeteksian Kendaraan**

Pada Gambar 3.5, menunjukkan mobil yang dideteksi dengan ditandai kotak berwarna ungu. Kendaraan yang dipilih untuk dideteksi akan berjalan selama 29 fps / 29 frame dan setelah 29 frame akan muncul hasil kecepatan yang dilakukan oleh kendaraan tersebut.



**Gambar 3.6 Perpindahan Lajur**

Proses perpindahan lajur kendaraan sama dengan proses dari kecepatan. Hasil dari perpindahan lajur akan dapat terlihat setelah 29 frame diproses oleh program. Saat pendeteksian, setiap perpindahan lajur dapat diketahui dari tanda garis berwarna hijau dibelakangnya mengikuti setiap perpindahan kendaraan, Gambar 3.6. Dengan didapatkannya hasil perpindahan lajur dan kecepatan tersebut maka akan dilakukan pengolahan data. Pengolahan data ini adalah pengelompokkan hasil dengan batas bawah dan batas atas kecepatan sebuah kendaraan yang terdeteksi sehingga akan didapatkan sebuah kesimpulan cara mengemudi seseorang saat berkendara.

### **3.1.3.3. Output**

Tahap akhir ini adalah tahap dimana hasil yang dikeluarkan oleh program tersebut. Pada Gambar 3.7, menunjukkan hasil deteksi dari kendaraan yang sudah dipilih. Hasil yang ditampilkan pada aplikasi dibagi menjadi tiga bagian yaitu Kecepatan kendaraan, yang kedua Lajur yang menunjukkan jumlah perubahan lajur yang terjadi, dan yang ketiga adalah kesimpulan

yang menjelaskan cara mengemudi pengendara tersebut saat di jalan raya. Untuk uji coba, akan diperinci lebih lagi pada Bab V. Seluruh hasil dari pendeteksian kendaraan tersebut dapat dilihat kembali setelah program selesai dijalankan dengan format ekstensi berekstensi file.txt.

Parameter	
Jarak Px	74 Px/ 6 M
Frame	29 frm/ 0 Dtk
Kecepatan	6Meter/Detik
Kecepatan	21Km/Jam
Lajur	0
Kesimpulan	Aman

**Gambar 3.7 Hasil Deteksi**

### 3.2. Perancangan Sistem

Tahap ini meliputi perancangan basis data, tampilan antarmuka, dan perancangan alur proses penggunaan sistem yang diharapkan dapat memenuhi tujuan dari pengembangan aplikasi ini. Perlu diketahui bahwa aplikasi ini dibangun dalam kondisi lingkungan tertentu, dan dapat dioperasikan dalam lingkungan tertentu pula. Lingkungan pengembangan aplikasi adalah sebagai berikut.

Komputer	: Prosesor Intel® Core™ i3-CPU (2.53GHz), RAM 8 GB.
Sistem operasi	: Windows 10
IDE	: Microsoft Visual Studio Profesional 2012
Bahasa Pemrograman	: C#
Library	: EmguCV 3.1

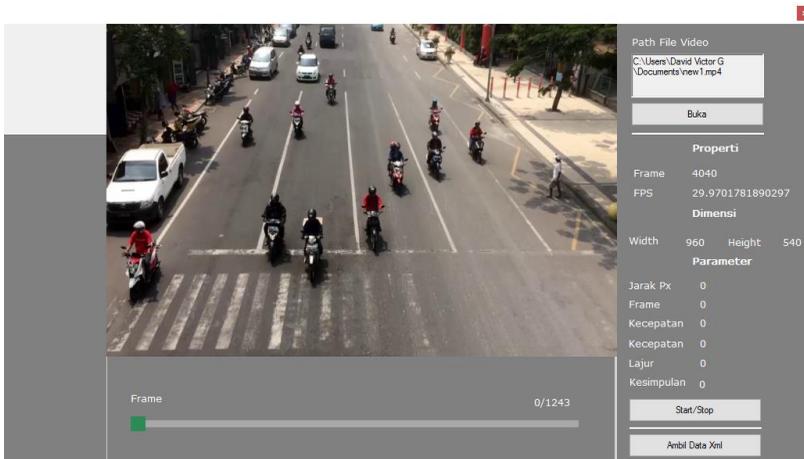
### 3.2.1. Perancangan Antar Muka Aplikasi

Subbab ini menjelaskan bagaimana rancangan antarmuka yang akan berinteraksi secara langsung dengan pengguna pada saat tahap implementasi.

#### 3.2.1.1. Perancangan Tampilan Awal Aplikasi Dibuka

Pada tampilan awal ketika pengguna membuka aplikasi terdapat tiga tombol yaitu : Buka, Start/Stop, dan ambil data XML yang ketiganya ini adalah fungsionalitas dalam aplikasi ini.

Pengerjaan Tugas Akhir ini memiliki tiga langkah utama yang harus dilakukan sebelum nantinya melakukan pendeteksian yaitu pengambilan background jalan, pengambilan gambar tiap frame, dan pengambilan data XML. Perancangan tampilan awal sistem dapat dilihat pada Gambar 3.8.



**Gambar 3.8 Perancangan Tampilan Awal Sistem**

Sesuai yang telah dijelaskan sebelumnya, terdapat sub menu atau bagian untuk melakukan proses pengambilan data sebelum melakukan pendeteksian.

Tahap awal yang dilakukan adalah mengambil background jalan. Pengambilan background jalan dilakukan saat menginputkan video dengan memilih tombol buka. Setelah memilih video yang akan digunakan, maka sistem akan langsung menampilkan proses pengambilan background jalan dari video tersebut. Pada Gambar 3.9 menunjukkan proses pengambilan background jalan berdasarkan video yang diinputkan.



**Gambar 3.9 Pengambilan Background Jalan**

Tujuan dari pengambilan background sebagai pembandingan dengan frame – frame yang nantinya akan didapatkan di proses selanjutnya. Pengambilan gambar tiap frame video dan data XML yang dibutuhkan untuk pendeteksian dengan memilih tombol Ambil Data XML. Sistem akan langsung melakukan pengambilan gambar tiap frame atau pemisahan gambar video.

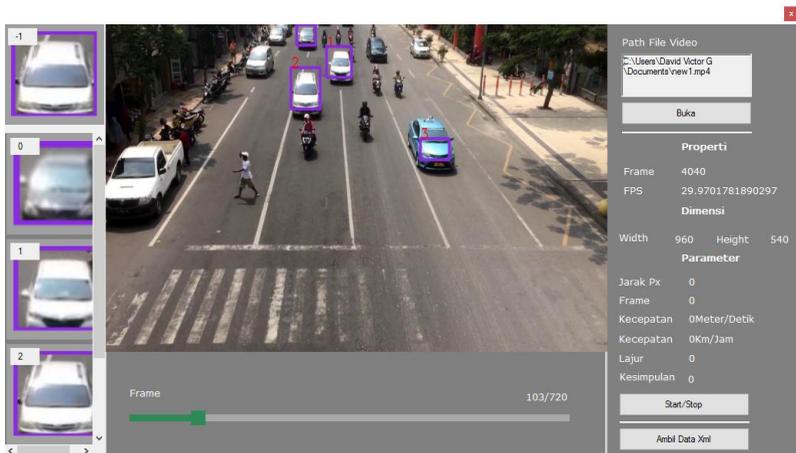


**Gambar 3.10 Data XML**

Gambar 3.10 menunjukkan proses pemisahan gambar video dan nantinya digunakan untuk pengambilan data XML. Sebelum melakukan pengambilan data XML dapat dilakukan filter objek agar mendapatkan hasil yang baik. Untuk pengambilan data XML dengan memilih tombol start dan tombol stop untuk memberhentikan pengambilan data XML.

### 3.2.1.2. Perancangan Tampilan Saat Proses Analisa Berjalan

Gambar 3.11 merupakan tampilan saat program dijalankan dan pada saat proses analisa berjalan, maka program akan menampilkan kotak yang menunjukkan sebuah mobil yang terdeteksi karena tersimpan dalam data XML. Program akan menampilkan mobil yang terdeteksi pada bagian samping kiri dari program. Kotak yang terdeteksi akan memiliki nomer yang akan memudahkan dalam memilih kendaraan mana yang ingin dideteksi.



**Gambar 3.11 Tampilan Saat Mendeteksi Kendaraan**

Saat ingin memilih kendaraan yang ingin dideteksi, maka pilih gambar pada samping kiri dari program dan gambar yang terpilih akan ditampilkan pada kotak dibagian kiri atas. Gambar kendaraan yang muncul merupakan gambar dari hasil pendeteksian XML. Saat gambar yang terpilih sudah ada didalam kotak bagian kiri atas, maka pilih tombol start program akan melakukan penghitungan.

### 3.2.1.3. Perancangan Tampilan Hasil

Hasil yang ditampilkan setelah aplikasi selesai dijalankan berbentuk berkas note dengan format .txt yang berisi data-data yang didapatkan dari kendaraan yang terdeteksi. Hasil deteksi juga dapat dilihat pada program setelah proses deteksi perpindahan lajur dan kecepatan selesai. Pada Gambar 3.12, terlihat tampilan hasil deteksi berformat .txt dengan menggunakan aplikasi notepad++. Hasil yang di catat oleh program adalah Nomer Frame (Nomer kendaraan saat terpilih pertama), Frame, Jarak (Km), Kecepatan, (Km/Jam).

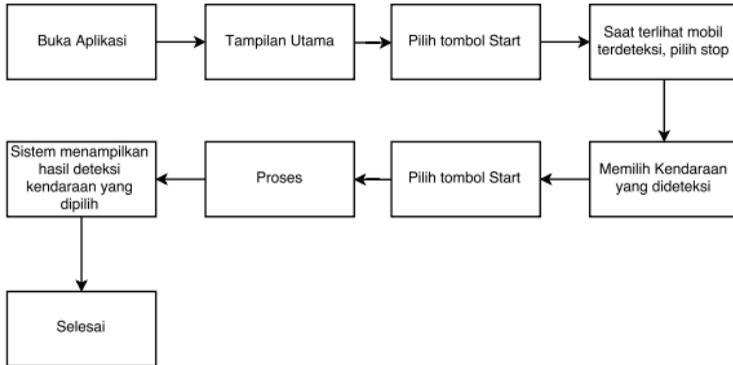
Nomor	Frame	Frame	Jarak (Km)	Kecepatan (Km/Jam)
0		1	8	28,8
1		1	5	18
2		1	13	43,2
3		1	4	14,4
4		1	6	18
5		1	11	39,6
6		1	10	36
7		1	12	39,6
8		1	21	75,6
9		1	5	18
10		1	12	43,2

**Gambar 3.12 Hasil Ouput Program**

### 3.2.2. Perancangan Alur Proses Penggunaan Aplikasi

Pada tahap ini akan dijelaskan mengenai rancangan alur proses penggunaan aplikasi yang digunakan sebagai acuan dalam

pembangunan aplikasi. Gambar 3.13 adalah penjelasan alur penggunaan aplikasi dalam bentuk *workflow*.



**Gambar 3.13 Workflow Alur Penggunaan Aplikasi**

Alur proses ini membantu memperlihatkan langkah demi langkah yang akan dilakukan pengguna dalam menjalankan perangkat lunak ini sehingga terlihat jelas antara *input*, *proses*, dan *output*. Langkah-langkah yang dilakukan sama semua. Pertama memasukkan video yang ingin diamati. Setelah video terdeteksi maka selanjutnya adalah pengambilan background jalan saat tidak ada objek. Setelah itu pengambilan gambar tiap frame dan melakukan pengambilan data xml. Kemudian selesai maka program akan melakukan proses pendeteksi kendaraan pada video dan akan didapatkan perubahan lajur yang terjadi dan kecepatan mobil yang terdeteksi dan akan terdapat hasil kesimpulan apakah kendaraan yang terdeteksi tersebut ini mengemudikan kendaraannya dengan Aman atau Bahaya. Setelah selesai melakukan pengamatan terhadap video tersebut maka akan muncul hasil deteksi dari video yang sudah diamati dalam ekstensi hasil .txt.

## **BAB IV IMPLEMENTASI**

Bab ini membahas implementasi dari analisis dan perancangan sistem yang telah dibahas pada Bab III. Namun dalam penerapannya, rancangan tersebut dapat mengalami perubahan minor sewaktu-waktu apabila dibutuhkan.

### **4.1. Lingkungan Implementasi**

Dalam implementasinya, lingkungan yang digunakan sama seperti yang dituliskan pada rancangan, yakni menggunakan beberapa perangkat pendukung sebagai berikut.

#### **4.1.1. Lingkungan Implementasi Perangkat Keras**

Perangkat keras yang digunakan dalam implementasi pengembangan aplikasi ini adalah Komputer. Spesifikasi komputer yang digunakan adalah laptop dengan Prosesor Intel® Core™ i3-CPU (2.53GHz), RAM 8 GB.

#### **4.1.2. Lingkungan Implementasi Perangkat Lunak**

Penjelasan perangkat lunak yang digunakan dalam implementasi aplikasi ini adalah sebagai berikut:

- Microsoft Windows 10 Pro sebagai sistem operasi pada *notebook*.
- Microsoft Visual Basic Profesional 2012 untuk IDE perancangan sistem.
- EmguCV 3.1 sebagai *library* pendukung dalam perancangan sistem.

### **4.2. Implementasi Kode**

Subbab ini membahas tentang implementasi tampilan antarmuka yang telah dirancang dan dibahas pada Bab III. Selanjutnya akan dirinci dengan detail sebagai berikut.

### 4.2.1. Implementasi Input

Kode Sumber 4.1 merupakan implementasi untuk melakukan input video yang akan diproses ke tahap berikutnya. Video yang diinputkan secara otomatis akan langsung melakukan background subtraction. Hasilnya nanti akan dijadikan sebagai acuan dalam proses selanjutnya.

```

1. private void btn_open_Click(object sender, EventArgs e)
2.     {
3.         if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
4.         {
5.             this.xmlParameterTemp = this.param.getDataVideo(openFileDialog1.FileName);
6.             this.xmlParameter.xmlsaves_.deleteAll();
7.             this.xmlParameter.xmlsaves_.add(this.xmlParameterTemp
8.             );
9.             this.formAnalisisBackground.loadData();
10.            this.formAnalisisBackground.Show();
11.        }
12.    }
13.

```

**Kode Sumber 4.1 Input Video**

### 4.2.2. Implementasi Processing Video

Pada tahap ini processing video terbagi menjadi 3 fungsi, yaitu mendeteksi mobil, menghitung perubahan lajur yang terjadi, mendapatkan kecepatan mobil, dan menentukan cara mengemudi kendaraan tersebut.

#### 4.2.2.1. Implementasi Mendeteksi Kendaraan

Kode Sumber 4.2 adalah untuk mendeteksi kendaraan yang ada selama video berjalan. Agar dapat mendeteksi kendaraan maka diperlukan persiapan – persiapan sebelum melakukan pendeteksian. Gambar tiap frame dan XML

sangat diperlukan dalam proses ini. Setelah didapatkan seperti yang telah dijelaskan diatas maka proses selanjutnya adalah melakukan pendeteksian. Untuk memulai mendeteksi kendaraan dapat dilakukan dengan menekan tombol “Start” dan saat itulah kendaraan yang lewat akan terdeteksi tanpa melakukan proses penghitungan.

```

1. Image<Gray, byte> temps = new Image<Gray, byte>(
2.         this.GrayBin.ToBitmap()
3.         );
4.         CvInvoke.Resize(
5.             temps,
6.             temps,
7.             new System.Drawing.Size(tem
8. ps.Width , temps.Height ), 0, 0,
9.             Emgu.CV.CvEnum.Inter.Linear
10.        );
11.        CvBlobs blobs = new CvBlobs();
12.        _blobDetector.Detect(temps, blobs);
13.        blobs.FilterByArea(100, int.MaxValue);
14.
15.        int cnt = 1;
16.        foreach (var pair in blobs)
17.        {
18.            CvBlob b = pair.Value;
19.            CvInvoke.Rectangle(temps, b.Boundin
20. gBox, new MCvScalar(255.0, 0.0, 0.0), 2);
21.            CvInvoke.PutText(temps, cnt.ToStrin
22. g(), new Point(b.BoundingBox.X, b.BoundingBox.Y), F
23. ontFace.HersheyPlain, 4.0, new MCvScalar(255.0, 0.0
24. , 0.0));
25.            cnt++;
26.        }

```

#### Kode Sumber 4.2 Memulai Pendeteksian

##### 4.2.2.1. Implementasi Perubahan Lajur

Kode Sumber 4.3 ini adalah implementasi untuk melakukan pencatatan tiap perpindahan kendaraan hingga

waktu pendeteksian selesai. Setiap perpindahan akan di perlihatkan dengan garis yang mengikuti kendaraan tersebut.

```

1. int diffY_objeck = Math.Abs(this.trackingLajur.y -
   this.tempCorTracking.y);
2.         if (diffY_objeck > this.tempCorTrac
   king.w)
3.         {
4.             this.trackingLajur = this.tempC
   orTracking;
5.             this.lajur++;
6.         }

```

#### Kode Sumber 4.3 Implementasi Perubahan Lajur

#### 4.2.2.2. Implementasi Perhitungan Kecepatan

Kode Sumber 4.4 berikut ini adalah implementasi untuk melakukan perhitungan dalam mendapatkan kecepatan terhadap kendaraan yang terdeteksi. Perhitungan ini didapat dari pengumpulan pergerakan selama 29 fps.

```

1. double _x1 = (this.firstCors.x - tempCorTracking.x)
   * (this.firstCors.x - tempCorTracking.x);
2. double _y1 = (this.firstCors.y - tempCorTracking.y)
   * (this.firstCors.y - tempCorTracking.y);
3. _x1 = Math.Sqrt(_x1 + _y1);
4.
5. int diffY_objeck = Math.Abs(this.trackingLajur.y -
   this.tempCorTracking.y);
6.
7. if (diffY_objeck > this.tempCorTracking.w)
8. {
9.     this.trackingLajur = this.tempCorTracking;
10.    this.lajur++;
11. }
12.
13. if (this.updateDataPosisiObjeckCalclajur > 10)

```

```

14. {
15.     this.updateDataPosisiObjeckCalcLajur = 0;
16.
17.     this.trackingLajur = this.tempCorTracking;
18. }
19. this.updateDataPosisiObjeckCalcLajur++;
20. this.Text = diffY_objeck.ToString() + "," + this.tempCorTracking.w.ToString() + "," + this.lajur.ToString() + "," + this.trackingLajur.y.ToString() + "," + this.tempCorTracking.y.ToString();
21.
22. if (this.countTracking < this.xmlParameterTemp.getdata_typeDouble(this.param.videoType.fps))
23. {
24.
25.     this.jarak = _x1 / 12;
26.
27.     this.lbl_jarak_.Text = Math.Floor(_x1).ToString() + " Px" + "/" + Math.Floor(jarak).ToString() + " M";
28.
29.     this.time = ((double)this.countTracking / this.xmlParameterTemp.getdata_typeDouble(this.param.videoType.fps));
30.     this.lbl_frame_.Text = this.countTracking.ToString() + " frm" + "/" + Math.Floor(time).ToString() + " Dtk";
31.
32.     this.lbl_lajur.Text = this.lajur.ToString();
33.
34.     for (int i = 0; i < (this.cortracking.Count - 1); i++)
35.     {
36.         Point start = new Point(this.cortracking[i].x, this.cortracking[i].y);
37.         Point second = new Point(this.cortracking[i + 1].x, this.cortracking[i + 1].y);
38.         LineSegment2D line5 = new LineSegment2D(start, second);
39.         CvInvoke.Line(imtemp, start, second, new Bgr(Color.Green).MCvScalar, 2);
40.
41.     }

```

```

42.
43.     this.isAlreadyCalculateSpeed = false;
44. }
45. else
46. {
47.     if (!this.isAlreadyCalculateSpeed)
48.     {
49.         try
50.         {
51.             if (this.countTracking > 0)
52.             {
53.                 kec = Math.Floor(jarak / time);
54.             }
55.         }

```

**Kode Sumber 4.4 Menghitung Kecepatan**

### 4.2.2.3. Implementasi Penentuan Golongan

Kode Sumber 4.5 berikut ini adalah implementasi untuk melakukan pengelompokan kecepatan dan perubahan lajur berdasarkan golongan yang telah ditentukan. Pengelompokan terdiri dari 2, yaitu: Aman dan Bahaya.

```

1. bool syaratbahaya_1 = (((kec) * 3.6) < 60 && (this.
   lajur > 0));
2. bool syaratbahaya_2 = (((kec) * 3.6) > 60 && (this.
   lajur == 0));
3. bool syaratbahaya_3 = (((kec) * 3.6) > 60 && (this.
   lajur > 0));
4. if (syaratbahaya_1 || syaratbahaya_2 || syaratbahay
   a_3){
5.     this.lbl_kesimpulan.Text = "Bahaya";
6. }
7. else{
8.     this.lbl_kesimpulan.Text = "Aman";
9. }

```

**Kode Sumber 4.5 Penentuan Kesimpulan Deteksi**

### 4.2.3. Implementasi Output

Berikut ini adalah implementasi untuk menampilkan hasil dari proses pendeteksian kendaraan dan kecepatan kendaraan berdasarkan golongan yang ada. Hasil pendeteksian dapat dilihat pada aplikasi dan dalam berkas bernama file.txt. Kode Sumber 4.6 merupakan proses hasil dari pendeteksian dan penetapan kesimpulan yang dapat dilihat setelah program selesai mendeteksi suatu kendaraan. Penyimpanan data – data kendaraan yang berhasil diolah oleh program yang nantinya hasil tersebut akan dilihat dalam berkas bernama file.txt.

```

1.  this.isAlreadyCalculateSpeed = true;
2.  this.lbl_kecepatan_.Text = Math.Floor(kec).ToString
    () + "Meter/Detik";
3.  this.lbl_kecepatan2.Text = Math.Floor((kec) * 3.6).
    ToString() + "Km/Jam";
4.
5.  cls_speedData tempspeed = new cls_speedData();
6.  tempspeed.numberFrame = 0 ;
7.  tempspeed.jarak = jarak ;
8.  tempspeed.time = time ;
9.  tempspeed.kecepatan = (kec) * 3.6;
10. this.LogSpeed.add(tempspeed);
11.
12. bool syaratbahaya_1 = (((kec) * 3.6) < 60 && (this.
    lajur > 0));
13. bool syaratbahaya_2 = (((kec) * 3.6) > 60 && (this.
    lajur == 0));
14. bool syaratbahaya_3 = (((kec) * 3.6) > 60 && (this.
    lajur > 0));
15. if (syaratbahaya_1 || syaratbahaya_2 || syaratbahaya_3){
16. this.lbl_kesimpulan.Text = "Bahaya";
17. }else{
18. this.lbl_kesimpulan.Text = "Aman";
19. }
20. this.LogSpeedToFileTxt();

```

**Kode Sumber 4.6 Menampilkan Hasil Deteksi**

*[Halaman ini sengaja dikosongkan]*

## **BAB V**

### **PENGUJIAN DAN EVALUASI**

Bab ini membahas pengujian dan evaluasi pada aplikasi yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem yang telah dijabarkan pada Bab III dan terhadap tujuan dibuatnya aplikasi ini, yakni agar dapat mengetahui cara mengemudi seseorang saat berada di jalan.

#### **5.1. Lingkungan Pengujian**

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kakas sebagai berikut:

Prosesor	: Prosesor Intel® Core™ i3-CPU
RAM	: 8 GB
Jenis <i>Device</i>	: Personal Computer
Sistem Operasi	: Windows 10

#### **5.2. Skenario Pengujian**

Pada bagian ini akan dijelaskan skenario pengujian yang dilakukan. Uji coba dilakukan untuk mengetahui karakteristik pengemudi saat berkendara melalui input video. Hasil *output* dipengaruhi skenario yang digunakan, maka penulis membuat 2 skenario yang menguji tentang data XML dan kondisi jalan. Bagian yang diamati penulis adalah pengaruh Data XML dalam proses pendeteksian kecepatan dan perpindahan lajur. Setelah memiliki data XML tersebut, maka dapat dilanjutkan ke proses berikutnya yaitu deteksi perpindahan lajur dan kecepatan kendaraan. Dalam pendeteksian kecepatan dan perpindahan lajur ini, penulis membagi menjadi tiga kondisi jalan, yaitu : sepi, normal, dan ramai. Hasil dari sistem nantinya akan dibandingkan dengan hasil yang asli dan akan dihitung tingkat akurasi dari sistem tersebut. Hasilnya tersebut dapat menggambarkan perilaku pengemudi. Hasil dari sistem merupakan hasil yang dikeluarkan oleh program berdasarkan kendaraan yang dipilih untuk dideteksi sedangkan

Hasil Asli merupakan hasil dari pengamatan dan perhitungan berdasarkan pengelihan dari video yang ada.

### 5.2.1. Skenario 1

Data XML dalam program ini diperlukan untuk membantu proses mendeteksi objek kendaraan yang ada dalam sebuah video. Skenario ini adalah untuk mengetahui pengaruh dalam proses pendeteksian kendaraan dari pengambilan data XML. Setiap frame dalam video akan dibaca dan disimpan dalam XML tersebut. Gambar 5.1 terlihat data video yang sudah diinputkan kedalam program. Untuk mendapatkan jumlah pengambilan data XML pada program, dapat dilihat pada index dalam properti dalam gambar tersebut.

C:\Users\David Victor G\Documents\new1.mp4

	Properti		Dimensi
Frame	4040	Width	960
FPS	29	Height	540
index	127		

**Gambar 5.1 Pengambilam Data XML**

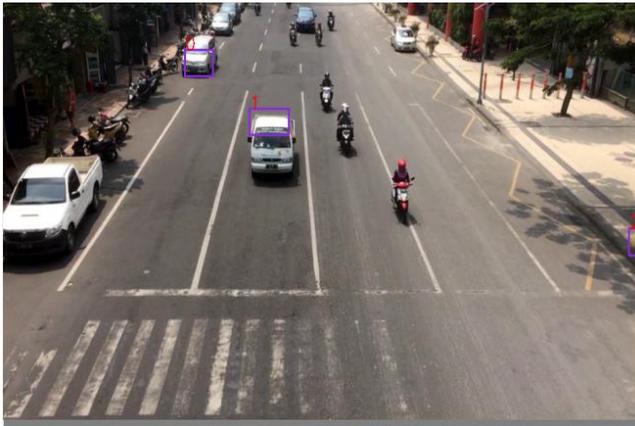
Pada Skenario 1 ini, penulis membagi menjadi dua kondisi yaitu pengambilan data penuh dan pengambilan data tidak penuh dari video.

### 5.2.2. Skenario 2

Pada skenario ini, penulis membagi menjadi tiga kondisi jalan yang ada, yaitu kondisi jalan sepi, normal, ramai. Ketiga kondisi jalan ini adalah keadaan yang paling sering terjadi dalam berkendara dijalan.

### 5.2.2.1. Kondisi Jalan Sepi

Kriteria jalan sepi adalah kondisi jalan yang tidak terdapat banyak kendaraan dalam video tersebut. Jalan dapat dikatakan sepi jika jumlah kendaraan (mobil) pada video berjumlah 0 – 3 mobil tanpa menghitung kendaraan yang berhenti. Pada Gambar 5.2, merupakan salah satu contoh keadaan jalan raya yang termasuk dalam kondisi jalan sepi.



**Gambar 5.2 Kondisi Jalan Sepi**

### 5.2.2.2. Kondisi Jalan Normal

Untuk kriteria jalan normal adalah kondisi jalan yang jumlah kendaraan saat di jalan tidak terlalu sepi dan tidak terlalu ramai. Jalan dapat dikatakan normal jika jumlah kendaraan (mobil) pada video berjumlah 4 – 5 mobil tanpa menghitung kendaraan yang berhenti. Pada Gambar 5.3, merupakan salah satu contoh keadaan jalan raya yang termasuk dalam kondisi jalan Normal.



**Gambar 5.3 Kondisi Jalan Normal**

### **5.2.2.3. Kondisi Jalan Ramai**

Untuk kriteria jalan ramai adalah kondisi jalan yang jumlah kendaraan yang padat. Jalan dapat dikatakan ramai jika jumlah kendaraan (mobil) pada video berjumlah  $\geq 6$  mobil tanpa menghitung kendaraan yang berhenti. Pada Gambar 5.4, merupakan salah satu contoh keadaan jalan raya yang termasuk dalam kondisi jalan ramai.



**Gambar 5.4 Kondisi Jalan Ramai**

Kondisi jalan sepi, normal, dan ramai memiliki langkah pengujian yang sama. Berikut ini adalah langkah – langkah pengujian untuk semua skenario:

- Melakukan uji coba dengan wilayah jalan sepi
- Memasukkan video berformat .mp4
- Aplikasi mengambil background jalan
- Aplikasi membaca video frame demi frame.
- Aplikasi memuat file XML.
- Aplikasi mendeteksi kendaraan
- User Memilih kendaraan yang akan dideteksi
- Aplikasi mendeteksi perubahan lajur
- Aplikasi menghitung kecepatan kendaraan
- Aplikasi menggolongkan cara mengemudi berdasarkan kecepatan dan perubahan lajur kendaraan yang terdeteksi.
- Aplikasi akan menampilkan hasil pencatatan dalam file berekstensi .txt.

Setelah langkah – langkah ini dilakukan maka akan keluar hasil dari pendeteksian kendaraan tersebut dan nantinya pengguna akan dapat mengetahui karakteristik cara mengemudi kendaraan tersebut beserta dengan kecepatan dan perubahan lajur yang terjadi. Pengujian akurasi aplikasi ini dilakukan secara mandiri dengan melakukan langkah – langkah diatas yang sama dengan skenario berbeda yang proses rancangan alur proses aplikasi sebagai tolak ukur keberhasilan pengujian, dan mengacu pada kasus penggunaan yang sebelumnya telah dijelaskan pada Bab III. Pengujian pada kebutuhan fungsionalitas dapat dijabarkan pada subbab berikut.

### **5.3. Akurasi Pengujian**

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang

diberikan meliputi dua skenario pengujian yang sudah dijelaskan pada subbab 5.2. Skenario pertama mengenai pengaruh Data XML terhadap pendeteksian kendaraan dan skenario kedua adalah Evaluasi akurasi hasil kecepatan dan perubahan lajur sesuai dengan modul yang terdapat aplikasi.

### 5.3.1. Skenario 1

Seperti yang sudah dijelaskan pada subbab 5.2.1, Skenario ini dibagi menjadi dua kali pengambilan data, yang pertama dengan mengambil data sebagian dari video dan yang kedua dengan mengambil sepenuhnya data dari video tersebut.

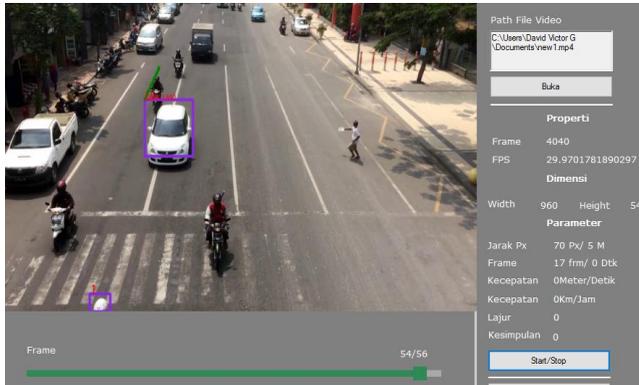
#### 5.3.1.1. Pengambilan Data Tidak Penuh

Proses yang dilakukan dalam pengambilan data XML ini hanya sebagian saja. Pada Gambar 5.5, index menunjukkan angka 55 yang berarti hanya ada 55 Data XML yang tersimpan oleh program. Saat pendeteksian dilakukan dan belum memenuhi 29 fps maka program akan terhenti dan tidak dapat menampilkan hasil pendeteksian.

	Properti		Dimensi
Frame	4040	Width	960
FPS	29	Height	540
index	55		

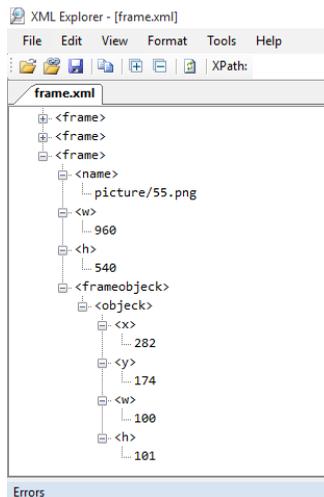
**Gambar 5.5 Data XML Sebagian**

Dengan adanya 55 data XML dalam program maka hasil yang terjadi saat pendeteksian kendaraan yang belum mencapai 29 fps adalah program terhenti dan tidak dapat menampilkan hasil pendeteksian seperti pada Gambar 5.6



**Gambar 5.6 Hasil Pendeteksian Data XML Sebagian**

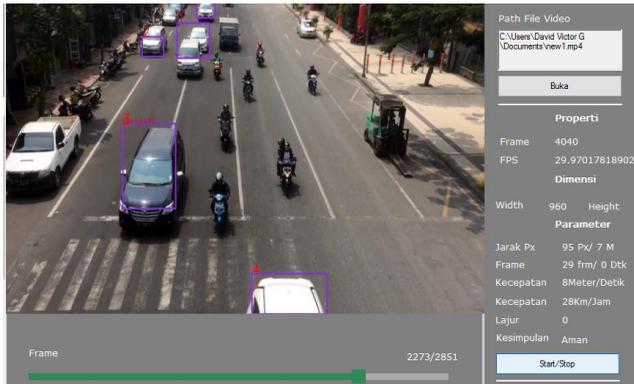
Pendeteksian pada Gambar 5.6 terhenti karena tidak di dapati data XML di frame berikutnya. Agar lebih jelas, pada Gambar 5.7 merupakan contoh data XML yang ada. Pada gambar tersebut menunjukkan bahwa data yang ada hanya sampai pada Frame 55, maka objek yang terdeteksi akan terhenti dan tidak dapat melanjutkan pendeteksian objek dan kecepatan.



**Gambar 5.7 Data XML Sebagian**

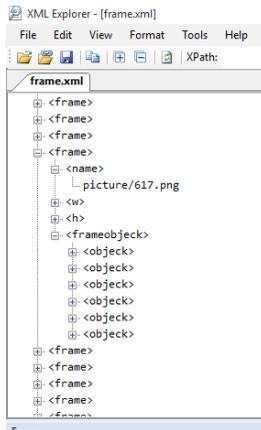
### 5.3.1.2. Pengambilan Data Penuh

Proses yang dilakukan dalam pengambilan data XML ini sama seperti pada pengambilan data sebagian hanya saja data diambil sampai video selesai. Jumlah frame pada video ini berjumlah 4040 maka data XML yang diambil berjumlah 4040. Hasil dari proses pendeteksian jika memiliki Data XML penuh maka akan dapat menyelesaikan pendeteksian dan menampilkan hasil tersebut seperti Gambar 5.8.



**Gambar 5.8 Hasil Pendeteksian Data XML Penuh**

Pada Gambar 5.9 merupakan hasil dari data XML yang di dapatkan program dengan pengambilan data hingga video berakhir. Pada pendeteksian di atas, proses selesai dilakukan hingga memunculkan hasil kecepatan dan pendeteksian karena data XML yang ada tersimpan secara keseluruhan tiap frame yang ada di dalam video. Dalam file XML tersebut berisi data file frame, ukuran frame, objek yang terdeteksi sebagai kendaraan. Dalam objek yang terdeteksi menyimpan data titik koordinat (x,y) dan ukuran objek kendaraan yang terdeteksi sebagai kendaraan. Jika objek yang terdeteksi sebagai kendaraan lebih dari satu maka akan ditampilkan keseluruhan data yang terdeteksi sebagai objek.



**Gambar 5.9 Data XML Penuh**

### 5.3.2. Skenario 2

Pada pengujian, terdapat tiga kondisi jalan yaitu sepi, padat, dan ramai. Ketiga kondisi ini mewakili kondisi nyata yang terjadi di jalan raya pada umumnya. Tabel 5.1 memperlihatkan hasil pengujian dari tiga kondisi jalan raya, jenis kendaraan yang terdeteksi, mana yang sesuai dengan definisi, mana yang tidak sesuai dengan definisi, dan mana yang tidak terdapat pada video inputan.

**Tabel 5.1 Hasil Deteksi Cara Menegemudi**

<b>Kondisi</b>	<b>Aman</b>	<b>Bahaya</b>
<b>Sepi</b>	<b>V</b>	<b>V</b>
<b>Normal</b>	<b>V</b>	<b>X</b>
<b>Padat</b>	<b>V</b>	<b>V</b>

Keterangan simbol:

V = Sesuai dan dipakai.

X = Tidak dipakai karena tidak ditemukan.

Akurasi pengenalan jenis kendaraan dihitung menggunakan Persamaan 5.1 yang merupakan perhitungan dengan cara sebagai berikut:

$$Akurasi\ Kecepatan = \frac{HK}{HB} * 100\%$$

$$Akurasi\ Perubahan\ Lajur = \frac{1}{(HB - HK) + 1} * 100\%$$

**Persamaan 5.1 Perhitungan Akurasi**

Di mana HK atau Hasil Kecil adalah hasil yang didapatkan dari sistem maupun asli yang memiliki jumlah lebih kecil sedangkan HB atau Hasil Besar adalah hasil yang didapatkan dari sistem maupun asli yang memiliki jumlah yang lebih besar. Keduanya digunakan dalam pencarian akurasi kecepatan maupun perpindahan lajur.

NO	MOBIL	PANJANG	LEBAR	TINGGI
1	Yaris	4115 mm	1700 mm	1475 mm
2	Honda Jazz	3920 mm	1695 mm	1525 mm
3	Avanza	4190 mm	1660 mm	1695 mm
4	Xenia	4190 mm	1660 mm	1685 mm
5	Ertiga	4265 mm	1695 mm	1685 mm
6	Terios	4405 mm	1695 mm	1740 mm
7	Rush	4420 mm	1745 mm	1740 mm
8	Livina	4485 mm	1695 mm	1595 mm
9	Inova	4485 mm	1775 mm	1750 mm
10	Kijang LGX	4495 mm	1670 mm	1775 mm

**Gambar 5.10 Dimensi Ukuran Mobil**

Dalam penghitungan kecepatan kendaraan yang sebenarnya, penulis menggunakan metode perhitungan panjang jalan berdasarkan berapa jumlah mobil yang berbaris sepanjang ukuran video yang kemudian dikalikan dengan 4300 Milimeter atau 4,3 Meter. Pada Gambar 5.10 merupakan data – data mobil yang mayoritas sering dijumpai di jalan raya. 4,3 Meter adalah

angka rata-rata ukuran panjang mobil sesuai dengan data pada gambar diatas. Dalam menentukan waktu asli kendaraan, penulis menggunakan perangkat lunak Sony Vegas Pro 13 untuk dapat menentukan waktu yang diperlukan sebuah mobil saat terdeteksi. Persamaan 2.1 merupakan rumus yang digunakan dalam perhitungan kecepatan ini. Dalam menentukan perubahan lajur sesungguhnya dengan melihat secara langsung dalam video.

### 5.3.2.1. Hasil Percobaan Kondisi Jalan Sepi

Berikut ini adalah hasil pengujian yang diperoleh untuk kondisi jalan sepi. Dalam pengujian ini, kendaraan yang dideteksi hanya dihitung selama 29 fps. Setelah 29 fps maka sistem akan menampilkan hasil deteksi. Kendaraan yang dideteksi hanya satu kali dan hanya lima kendaraan yang akan diambil sebagai pengujian. Hasil yang diperoleh dilampirkan pada Tabel 5.2.

**Tabel 5.2 Hasil Perhitungan Keadaan Jalan Sepi**

No Frame	No Mobil	Kecepatan		Akurasi (%)	Perubahan Lajur		Akurasi (%)
		Asli	Sistem		Asli	Sistem	
573	1	46	54	85,91	0	0	100,00
689	3	40	46	87,00	0	0	100,00
829	1	54	75	72,57	0	0	100,00
880	1	40	36	89,95	0	0	100,00
932	1	56	50	89,24	0	2	33,33

Dari data di atas, maka dilakukan penghitungan akurasi dengan Persamaan 5.1. Kendaraan dengan cara mengemudi aman ada 4 dan kendaraan dengan cara mengemudi berbahaya ada 1 berdasarkan sistem. Hasil akurasi kecepatan dan perpindahan lajur dengan kondisi jalan sepi yang didapatkan dari pengujian sistem dan keadaan asli adalah sebagai berikut : 84,94% untuk akurasi kecepatan dan 86,67% untuk akurasi perpindahan lajur.

### 5.3.2.2. Hasil Percobaan Kondisi Jalan Normal

Berikut ini adalah hasil pegujian yang diperoleh untuk kondisi jalan Normal. Dalam pengujian ini, kendaraan yang dideteksi hanya dihitung selama 29 fps. Setelah 29 fps maka sistem akan menampilkan hasil deteksi. Kendaraan yang dideteksi hanya satu kali dan hanya lima kendaraan yang akan diambil sebagai pengujian. Hasil yang diperoleh dilampirkan pada Tabel 5.3.

**Tabel 5.3 Hasil Perhitungan Keadaan Jalan Normal**

No Frame	No Mobil	Kecepatan		Akurasi (%)	Perpindahan Lajur		Akurasi
		Asli	Sistem		Asli	Sistem	
188	3	43	50	86,36	0	0	100,00
253	1	40	46	87,00	0	0	100,00
282	2	50	54	86,48	0	0	100,00
330	0	48	75	63,97	0	0	100,00
391	1	40	21	52,53	0	0	100,00

Dari data di atas, maka dilakukan penghitungan akurasi Persamaan 5.1. Kendaraan dengan cara mengemudi aman ada 5 dan kendaraan dengan cara mengemudi berbahaya terdapat 1 kendaraan. Berdasarkan sistem, nomer frame 330 memiliki kecepatan 75 km/jam. Hasil akurasi kecepatan dan perpindahan lajur dengan kondisi jalan normal yang didapatkan dari pengujian sistem dan keadaan asli adalah sebagai berikut : 75,27% untuk akurasi kecepatan dan 100,00% untuk akurasi perpindahan lajur.

### 5.3.2.3. Hasil Percobaan Kondisi Jalan Ramai

Berikut ini adalah hasil pegujian yang diperoleh untuk kondisi jalan Ramai. Dalam pengujian ini, kendaraan yang dideteksi hanya dihitung selama 29 fps. Setelah 29 fps maka sistem akan menampilkan hasil deteksi. Kendaraan yang dideteksi hanya satu kali dan hanya lima kendaraan yang akan diambil sebagai pengujian. Hasil yang diperoleh dilampirkan pada Tabel 5.4.

**Tabel 5.4 Hasil Perhitungan Keadaan Jalan Ramai**

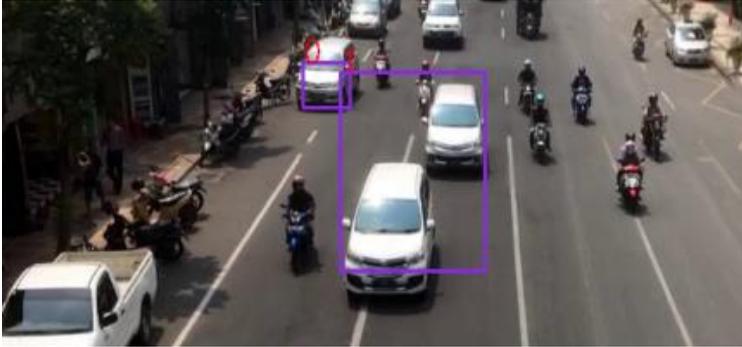
No Frame	No Mobil	Kecepatan		Akurasi (%)	Perubahan Lajur		Akurasi (%)
		Asli	Sistem		Asli	Sistem	
2068	1	40	28	70,04	0	1	50,00
2149	3	40	43	93,07	0	0	100,00
2251	1	40	39	97,55	0	0	100,00
2328	2	37	36	97,88	0	0	100,00
2361	1	40	39	97,55	1	0	50,00

Dari data di atas, maka dilakukan penghitungan akurasi dengan Persamaan 5.1. Kendaraan dengan cara mengemudi aman ada 5 dengan terdeteksi 1 kali berpindah lajur berdasarkan sistem dengan nomer frame 2068 dan berdasarkan keadaan asli dengan nomer frame 2361. Kendaraan dengan cara mengemudi berbahaya tidak ada. Hasil akurasi kecepatan dan perpindahan lajur dengan kondisi jalan normal yang didapatkan dari pengujian sistem dan keadaan asli adalah sebagai berikut : 91,22% untuk akurasi kecepatan dan 80,00% untuk akurasi perpindahan lajur.

#### **5.4. Evaluasi Pengujian**

Pada subbab ini membahas hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian kebutuhan fungsional beserta evaluasi berupa akurasi pada masing-masing skenario.

Hasil dari Skenario 1 dapat disimpulkan bahwa keterbatasan XML dapat mempengaruhi pendeteksian kendaraan dan mempengaruhi sistem dalam melakukan proses perhitungan kecepatan maupun perpindahan lajur. Dalam proses pengambilan XML juga dapat terlihat bahwa kendaraan yang berdekatan dan memiliki warna yang sama akan terdeteksi sebagai satu objek. Gambar 5.11, pada kotak ungu bernomer satu menunjukkan bahwa kendaraan yang berdekatan dan memiliki warna sama dapat terdeteksi sebagai satu objek.



**Gambar 5.11 Dua Objek Terdeteksi Menjadi Satu**

Untuk Skenario 2, rangkuman mengenai hasil akurasi dapat dilihat pada Tabel 5.5. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik. Sehingga bisa ditarik kesimpulan bahwa fungsionalitas dari aplikasi telah dapat bekerja sesuai dengan yang diharapkan.

**Tabel 5.5 Rangkuman Hasil Pengujian Skenario 2**

Skenario Pengujian	Hasil
Pengujian dengan kondisi jalan sepi	Berhasil
Pengujian dengan kondisi jalan normal	Berhasil
Pengujian dengan kondisi jalan ramai	Berhasil

Hasil *output* keseluruhan hasil pendeteksian program yang dapat dilihat di berkas file.txt. Gambar 5.12 merupakan hasil tampilan output yang terdapat informasi Nomer Frame, Frame, Jarak, dan kecepatan.

<u>Nomor</u>	<u>Frame</u>	<u>Frame</u>	<u>Jarak (Km)</u>	<u>Kecepatan (Km/Jam)</u>
0		1	8	28,8
1		1	5	18
2		1	13	43,2
3		1	4	14,4
4		1	6	18
5		1	11	39,6
6		1	10	36
7		1	12	39,6
8		1	21	75,6
9		1	5	18
10		1	12	43,2

**Gambar 5.12 Output dalam berkas file.txt**

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan Tugas Akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

#### **6.1. Kesimpulan**

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. XML yang didapat mempengaruhi keberhasilan pendeteksian sebuah kendaraan. Data XML yang disimpan adalah gambar tiap frame. Jika saat deteksi tidak terdapat data XML frame selanjutnya maka pendeteksian terhenti.
2. Tingkat Akurasi di ketiga kondisi jalan memiliki tingkat akurasi yang berbeda beda. Rata – rata nilai akurasi kecepatan pada kondisi jalan sepi 84,94%, jalan normal 75,27%, dan jalan ramai 91,22%. Rata – rata nilai perpindahan lajur pada kondisi jalan sepi 86,67%, jalan normal 100%, dan jalan ramai 80%.
3. Kendaraan yang saling berdekatan dan memiliki warna sama sulit untuk dideteksi. Warna sama dan berdekatan akan terdeteksi sebagai satu objek.

#### **6.2. Saran**

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan:

1. Peningkatan akurasi untuk masing-masing modul agar hasil semakin membaik
2. Meningkatkan hasil pendeteksian mobil ketika kendaraan yang terdeteksi lebih dari satu.

3. Pengambilan video dari sisi yang lain agar lebih memudahkan pendeteksian dan meningkatkan hasil.
4. Melakukan pengembangan system agar memungkinkan untuk melakukan deteksi kendaraan secara *real time*.
5. Mencari alternatif untuk mendeteksi kendaraan yang berwarna gelap.
6. Video jalan sebaiknya tidak terlalu lebar agar mendapatkan tingkat kepadatan yang lebih baik.

## DAFTAR PUSTAKA

- [1] “About - OpenCV library.” [Online]. Available: <http://opencv.org/about.html>. [Accessed: 30-May-2017].
- [2] “Emgu CV: OpenCV in .NET (C#, VB, C++ and more).” [Online]. Available: [http://www.emgu.com/wiki/index.php/Main\\_Page](http://www.emgu.com/wiki/index.php/Main_Page). [Accessed: 13-Dec-2017].
- [3] “OpenCV: How to Use Background Subtraction Methods.” [Online]. Available: [https://docs.opencv.org/3.2.0/d1/dc5/tutorial\\_background\\_subtraction.html](https://docs.opencv.org/3.2.0/d1/dc5/tutorial_background_subtraction.html). [Accessed: 13-Dec-2017].
- [4] T. B. Nguyen and S. T. Chung, “An Improved Real-Time Blob Detection for Visual Surveillance,” in *2009 2nd International Congress on Image and Signal Processing*, 2009, pp. 1–5.
- [5] “Median Filter.” [Online]. Available: [http://fourier.eng.hmc.edu/e161/lectures/smooth\\_sharpen/node2.html](http://fourier.eng.hmc.edu/e161/lectures/smooth_sharpen/node2.html). [Accessed: 13-Dec-2017].
- [6] “Basic Thresholding Operations — OpenCV 2.4.13.4 documentation.” [Online]. Available: <https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html>. [Accessed: 13-Dec-2017].
- [7] “Arti kata cepat - Kamus Besar Bahasa Indonesia (KBBI) Online.” [Online]. Available: <https://kbbi.web.id/cepat>. [Accessed: 19-Dec-2017].

## BIODATA PENULIS



David Victor Giandly, lahir pada tanggal 23 September 1995 di Surabaya. Penulis menempuh pendidikan perguruan tinggi di Institut Teknologi Sepuluh Nopember Surabaya di Departemen Teknik Informatika Fakultas Teknologi Informasi dan Komunikasi pada tahun 2013. Penulis memiliki pengalaman menjadi asisten dosen pada mata kuliah Organisasi Komputer dan Komputasi Numerik.

Penulis terlibat aktif dalam organisasi kemahasiswaan serta kegiatan kepanitiaan selama berkuliah, antara lain Badan Eksekutif Mahasiswa Fakultas Teknologi Informasi (BEM FTIf) ITS pada tahun 2015-2016. Dalam melakukan pengerjaan Tugas Akhir, penulis memiliki ketertarikan pada bidang Dasar dan Terapan Komputasi (DTK). Untuk menghubungi penulis dapat melalui *email*: [david.victor.giandly13@mhs.if.its.ac.id](mailto:david.victor.giandly13@mhs.if.its.ac.id).