

**PERANCANGAN SISTEM PENGHITUNG JUMLAH
KENDARAAN DUA ARAH DENGAN METODE
BACKGROUND SUBTRACTION DI JALAN TOL
JATINGALEH SEMARANG**

LAPORAN TUGAS AKHIR

Tugas Akhir ini disusun guna memenuhi salah satu syarat untuk menyelesaikan
Program studi Teknik Informatika S-1 pada Fakultas Teknologi Industri
Universitas Islam Sultan Agung



**DISUSUN OLEH :
M TAUFIQUR RAHMAN
NIM 32601601052**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG**

2022

FINAL PROJECT

***DESIGN OF TWO-WAY VEHICLE COUNTING SYSTEM WITH
BACKGROUND SUBTRACTION METHOD ON JATINGALEH
TOLL ROAD SEMARANG***

*Purposed to complete the requirement to obtain a bachelor's degree (S-1) at
Informatic Engeneering Departement of Industrial Technology Faculty Sultan
Agung Islamic University*



Arranged By:

M TAUFIQUR RAHMAN

NIM 32601601052

**MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG**

2022

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul “**PERANCANGAN SISTEM PENGHITUNG JUMLAH KENDARAAN DUA ARAH DENGAN METODE BACKGROUND SUBTRACTION DI JALAN TOL JATINGALEH SEMARANG**” ini disusun oleh :

Nama : M Taufiqur Rahman

NIM : 32601601052

Program Studi : Teknik Informatika

Telah disahkan oleh dosen pembimbing pada :

Hari :

Tanggal :

Pembimbing I

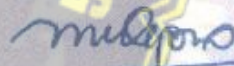


Sam Farisa Chaerul Haviana, ST., M.Kom

NIDN.0628028602

Mengesahkan,

Pembimbing II



Ir. Sri Mulyono, M.Eng

NIDN.0626066601

Mengetahui,

Ketua Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Islam Sultan Agung



Ir. Sri Mulyono, M.Eng

NIDN.0626066601

LEMBAR PENGESAHAN PENGUJI

Laporan tugas akhir dengan judul “**PERANCANGAN SISTEM PENGHITUNG JUMLAH KENDARAAN DUA ARAH DENGAN METODE BACKGROUND SUBTRACTION DI JALAN TOL JATINGALEH SEMARANG**” ini telah dipertahankan di depan dosen penguji Tugas Akhir pada :

Hari :

Tanggal :

TIM PENGUJI

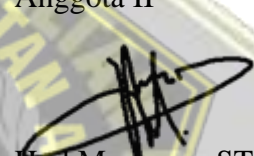
Anggota I



Andi Riansyah, ST, M.Kom.

NIDN. 0609108802

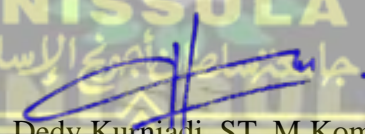
Anggota II



Hud Munawar, ST., MT.

NIK.210616052

Ketua Penguji



Dedy Kurniadi, ST, M.Kom.

NIDN.0622058802

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : M Taufiqur Rahman
NIM : 32601601052
Judul Tugas Akhir : PERANCANGAN SISTEM PENGHITUNG
JUMLAH KENDARAAN DUA ARAH
DENGAN METODE *BACKGROUND*
SUBTRACTION DI JALAN TOL
JATINGALEH SEMARANG

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 18 Juli 2022

Menyatakan



M Taufiqur Rahman

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : M Taufiqur Rahman
NIM : 32601601052
Program Studi : Teknik Informatika
Fakultas : Teknologi Industri
Alamat Asal : Jl.Ngablak Indah RT 11 / RW 04 Kel. Bangetayu
Kulon Kec. Genuk Kota Semarang

Dengan ini menyatakan Karya Ilmiah berupa Tugas Akhir dengan judul :
**PERANCANGAN SISTEM PENGHITUNG JUMLAH KENDARAAN DUA
ARAH DENGAN METODE *BACKGROUND SUBTRACTION* DI JALAN
TOL JATINGALEH SEMARANG**

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiatisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan Agung.

Semarang, 18 Juli 2022

Menyatakan

M Taufiqur Rahman

KATA PENGANTAR


Dengan mengucapkan puji syukur atas kehadiran Allah SWT berkat rahmat dan karunianya, penulis dapat menyelesaikan Tugas Akhir dengan judul **“Perancangan Sistem Penghitung Jumlah Kendaraan Dua Arah Dengan Metode *Background Subtraction* Di Jalan Tol Jatingaleh Semarang”** ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung.

Tugas akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak materis maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada:

1. Bapakku Arnawi, Ibuku Jinten dan Kakakku Dwi Agung Aryanto, S.Ak yang selalu memberikanku doa, nasihat dan semangat dalam menyelesaikan Tugas Akhir ini.
2. Kepada Bapak Sam Farisa Chaerul Haviana, ST., M.Kom, selaku Dosen Pembimbing I dan Bapak Ir. Sri Mulyono, M.Eng, selaku Dosen Pembimbing II yang telah memberikan saran, kritikan, masukan, motivasi serta kesabaran dalam penyusunan Tugas Akhir ini.
3. Kepada Rike Widia Astuti, S.Ak seorang perempuan yang selalu mensupport dan terus memberikan semangat kepada saya dalam menyelesaikan Tugas Akhir ini, saya ucapkan banyak-banyak terimakasih.

Dengan segala kerendahan hati, penulis menyadari masih banyak kekurangan dari ilmu pengetahuan dalam penyusunan laporan, sehingga penulis mengharapkan adanya saran dan kritikan bersifat membangun demi kesempurnaan laporan ini di masa mendatang.

Semarang, 18 Juli 2022



M Taufiqur Rahman

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN PEMBIMBING.....	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH..	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN.....	xiii
ABSTRAK.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Pembatasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	4
2.1 Tinjauan Pustaka.....	4
2.2 Dasar Teori.....	4
2.2.1 <i>Computer Vision</i>	4
2.2.2 <i>Open CV</i>	5
2.2.3 <i>Preprocessing</i>	5
2.2.4 <i>Segmentasi</i>	7
2.2.5 <i>Object Filtering</i>	8
2.2.6 <i>Deteksi Objek (Kendaraan)</i>	8
2.2.7 <i>Object Tracking</i>	9
2.2.8 <i>Vehicle Counting</i>	9

BAB III METODE PENELITIAN.....	11
3.1 Bahan Penelitian	11
3.2 Alat Penelitian	11
3.3 Pengumpulan Data.....	11
3.4 Konsep Sistem	12
3.5 Flowchart Sistem	14
3.6 Metode	16
3.7 Analisa Kebutuhan Sistem	17
3.7.1 Projek Model <i>Block Diagram</i>	17
3.7.2 Perancangan <i>Database</i>	18
3.8 Metode Pengujian.....	18
3.8.1 <i>Mean Square Error</i> (MSE).....	18
3.8.2 <i>Confusion Matrix</i>	19
3.8.3 Evaluasi Kinerja Sistem	21
3.9 Instalasi Sistem.....	21
3.10 Perancangan Prototipe	22
BAB IV HASIL DAN ANALISIS PENELITIAN	24
4.1 Cara Kerja Sistem.....	24
4.1.1 Pengambilan <i>Dataset</i>	24
4.1.2 Penentuan Daerah ROI	24
4.1.3 Proses Pengolahan Citra.....	25
4.1.4 Proses Pendeteksian Objek.....	27
4.1.5 Proses <i>Tracking Object and Capture</i>	27
4.1.6 Proses <i>Input Database</i>	29
4.2 Pembuatan Tampilan	29
4.2.1 Tampilan <i>GUI</i> Sistem.....	29
4.2.2 Tampilan Sistem Penghitung Kendaraan Keluar	30
4.2.3 Tampilan Sistem Penghitung Kendaraan Masuk	31
4.2.4 Tampilan Perhitungan Kendaraan 2 Arah	32
4.2.5 Tampilan <i>Capture</i>	33
4.2.6 Tampilan <i>Website</i>	34

4.3	Analisa Akurasi Sistem	35
4.3.1	Pengujian dan Analisa Pengaruh <i>Object Filter</i>	35
4.3.2	Pengujian dan Analisa pengaruh <i>Distance</i>	38
4.3.3	Analisa <i>Precision</i>	40
4.3.4	Analisa <i>Recall</i>	40
4.3.5	Analisa Akurasi	41
4.3.6	Evaluasi Kinerja Sistem	41
BAB V KESIMPULAN DAN SARAN.....		45
5.1	Kesimpulan.....	45
5.2	Saran	45
DAFTAR PUSTAKA		45
HALAMAN LAMPIRAN		46



DAFTAR TABEL

Tabel 3.1 Tabel Perancangan Database	18
Tabel 3.2 Tabel <i>Confusion Matrix</i>	19
Tabel 4.1 Tabel Analisa Pengaruh Object Filter pada ROI Kiri	35
Tabel 4.2 Tabel Analisa Pengaruh Object Filter pada ROI Kanan	36
Tabel 4.3 Tabel Analisa Distance pada ROI Kiri	39
Tabel 4.4 Tabel Analisa Distance pada ROI Kanan	39
Tabel 4.5 Tabel Evaluasi Kinerja Sistem.....	42



DAFTAR GAMBAR

Gambar 2.1 <i>OpenCV</i>	5
Gambar 2.3 Garis Bantu.....	6
Gambar 3.1 Konsep Sistem.....	12
Gambar 3.2 <i>Flowchart</i> Sistem	15
Gambar 3.3 Metode sistem	16
Gambar 3.4 Projek Model <i>Block Diagram</i>	17
Gambar 3.5 Desain Sistem.....	22
Gambar 3.6 Desain Web	23
Gambar 4.1 <i>Extract ROI</i>	25
Gambar 4.2 Pengolahan Citra <i>Background Subtraction</i>	26
Gambar 4.3 Pengolahan Citra <i>Threshold</i>	26
Gambar 4.4 Pengolahan Citra <i>Dilation</i>	27
Gambar 4.5 Tampilan <i>GUI</i> Sistem	29
Gambar 4.6 <i>Output Original Frame ROI Kiri</i>	30
Gambar 4.7 <i>Output ROI Kiri</i>	30
Gambar 4.8 <i>Output Original Frame ROI Kanan</i>	31
Gambar 4.9 <i>Output ROI Kanan</i>	31
Gambar 4.10 <i>Output Original Frame ROI</i>	32
Gambar 4.11 <i>Output ROI</i>	33
Gambar 4.12 <i>Output Capture File</i>	33
Gambar 4.13 Tampilan <i>Website</i>	34
Gambar 4.15 <i>Output</i> Pendeteksian Salah.....	37
Gambar 4.14 <i>Output</i> Pendeteksian Benar	36
Gambar 4.16 Grafik Pengujian Proses <i>Filter</i>	38
Gambar 4.17 Grafik Pengujian <i>Distance</i>	40

DAFTAR LAMPIRAN

Lampiran 1 Formulir Pendaftaran Seminar Proposal.....	46
Lampiran 2 Lembar Persyaratan Pendaftaran Seminar Proposal.....	47
Lampiran 3 Formulir Pendaftaran Seminar Kemajuan	48
Lampiran 4 Lembar Persyaratan Pendaftaran Seminar Kemajuan	49
Lampiran 5 Formulir Pendaftaran Ujian Sarjana / Sidang Tugas Akhir.....	50
Lampiran 6 Lembar Persyaratan Ujian Sarjana / Sidang Tugas Akhir.....	51
Lampiran 7 Hasil Turnitin.....	52
Lampiran 8 <i>Logbook</i> Pra Seminar Proposal Dosen Pembimbing I.....	56
Lampiran 9 <i>Logbook</i> Pra Seminar Proposal Dosen Pembimbing II	57
Lampiran 10 <i>Logbook</i> Pra Seminar Kemajuan Dosen Pembimbing I.....	58
Lampiran 11 <i>Logbook</i> Pra Seminar Proposal Dosen Pembimbing II	59
Lampiran 12 <i>Logbook</i> Pra Sidang Tugas Akhir Dosen Pembimbing I.....	60
Lampiran 13 <i>Logbook</i> Pra Sidang Tugas Akhir Dosen Pembimbing II	61
Lampiran 14 <i>Logbook</i> Pasca Sidang Tugas Akhir Dosen Pembimbing I.....	62
Lampiran 15 <i>Logbook</i> Pasca Sidang Tugas Akhir Dosen Pembimbing II.....	63



ABSTRAK

Pertumbuhan jumlah penduduk di Indonesia berdampak pada jumlah kendaraan yang beroperasi di jalan tol sehingga menimbulkan kemacetan pada ruas-ruas jalan tol. Diperlukan adanya survey penghitungan jumlah kendaraan untuk mengetahui kepadatan pada ruas-ruas jalan tol, pelaksanaan penghitungan jumlah kendaraan masih dilakukan oleh manusia yang memungkinkan bisa terjadinya *human error*. Sistem penghitung kendaraan di jalan tol merupakan teknologi komputer untuk mendeteksi dan menghitung setiap kendaraan yang bisa diimplementasikan oleh sistem secara *real-time*. Penggunaan metode *Background Subtraction* diimplementasikan pada proses segmentasi dengan cara memisahkan antara objek dan latar belakang dengan *threshold* untuk mempertegas objek serta menghilangkan *shadow*. Dalam proses pendeteksian kendaraan dilakukan pengambilan *dataset* sebagai *input* sistem yang nantinya akan dilakukan pengujian dan menghasilkan *output* dari sistem berupa data id, arah, tanggal dan waktu yang tersimpan pada *database MySQL* dan ditampilkan pada *website*. Pengujian dilakukan dengan mendapatkan nilai *error* terendah. Evaluasi kinerja sistem mendapatkan hasil *precision* 91,07%, *recall* 87,09%, dan akurasi 84,52%. Metode ini kurang mendapatkan hasil yang optimal apabila diimplementasikan pada saat kondisi lalu lintas yang padat.

Kata Kunci: *Background Subtraction*, Jalan Tol, Deteksi Objek, *Object Filtering*, *Tracking*

ABSTRACT

Population growth in Indonesia has an impact on the number of vehicles operating on toll roads, causing congestion on toll road sections. A survey of the number of vehicles is needed to determine the density on toll road sections, the implementation of the calculation of the number of vehicles is still carried out by humans which allows human error to occur. Vehicle counter system on toll roads is a computer technology to detect and count every vehicle that can be implemented by the system in real-time. The use of the Background Subtraction method is implemented in the segmentation process by separating the object and the background with a threshold to emphasize the object and eliminate shadows. In the vehicle detection process, datasets are taken as system input which will later be tested and produce output from the system in the form of id, direction, date and time data stored in the MySQL database and displayed on the website. Testing is done by getting the lowest error value. Evaluation of system performance obtained 91.07% precision, 87.09% recall, and 84.52% accuracy. This method does not get optimal results when implemented during heavy traffic conditions.

Keywords: *Background Subtraction*, Toll roads, *Object Detection*, *Object Filtering*, *Tracking*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Semakin pesatnya pertumbuhan ekonomi dan jumlah penduduk di Indonesia setiap tahunnya berdampak pada jumlah kendaraan yang beroperasi di jalan tol sehingga menimbulkan tingkat kemacetan pada ruas-ruas jalan tol. Saat ini belum diberlakukan pada pengelola jaringan jalan untuk mengoptimalkan penggunaan infrastruktur yang ada dan untuk menyediakan pengguna dengan kondisi yang dapat diandalkan dan nyaman. Gangguan yang sering terjadi pada lalu lintas seperti kemacetan dan kecelakaan terutama di jalan tol, gangguan tersebut memerlukan kontrol kondisi lalu lintas secara *real-time* dan implementasi cepat dari manajemen lalu lintas. Diperlukan adanya survey penghitungan jumlah kendaraan untuk mengetahui kepadatan pada ruas-ruas jalan tol, pelaksanaan penghitungan jumlah kendaraan masih dilakukan oleh manusia yang memungkinkan bisa terjadinya *human error*. Selain rentan terjadinya *human error*, penghitungan yang dilakukan oleh manusia memerlukan biaya tersendiri dalam pelaksanaannya mengakibatkan kurang efisien (Fajar Mit Cahyana, 2014).

Beberapa persyaratan harus dipenuhi untuk memastikan kinerja video sistem pengawasan berdasarkan kamera dapat mendeteksi kendaraan dengan mempertimbangkan beberapa kendala yang memungkinkan dapat mempengaruhi kinerja sistem, seperti kendaraan oklusi, kemacetan lalu lintas, dan kondisi penerangan. Tujuan dari penelitian ini untuk merancang sistem penghitung jumlah kendaraan dua arah di jalan tol.

Sistem penghitung kendaraan di jalan tol merupakan teknologi komputer untuk mendeteksi dan menghitung setiap kendaraan yang bisa diimplementasikan oleh sistem secara *real-time* menjadi salah satu terobosan untuk mengatasi masalah *human error* dan efisien. Sistem ini dapat diimplementasikan menggunakan *webcam* atau *camera eksternal* sebagai alat bantu pengambilan gambar kendaraan yang melintas dan *webapp* sebagai *monitoring data*.

1.2 Perumusan Masalah

Rumusan masalah dalam penelitian ini adalah :

1. Bagaimana mengimplementasikan sebuah sistem penghitung jumlah kendaraan dengan menggunakan metode *Background Subtraction* ?
2. Bagaimana menyimpan data kendaraan dapat tersimpan secara otomatis ke dalam *database*?

1.3 Pembatasan Masalah

Batasan masalah ini bertujuan untuk memudahkan dan menghindari adanya kegiatan diluar sasaran, sehingga dalam pembuatan laporan ini perlu ditentukan batasan masalah. Adapun batasan masalah dalam tugas akhir ini meliputi :

1. Pendeteksian dilakukan pagi, siang, sore dengan tingkat kecerahan yang optimum
2. Sistem hanya dapat menghitung jumlah kendaraan keluar atau masuk di Jalan Tol Jatingaleh Semarang
3. Dataset video yang digunakan berukuran 1920x1080 piksel.

1.4 Tujuan

Tujuan dari penelitian yang akan dilakukan adalah sebagai berikut :

1. Mengimplementasikan sistem penghitung kendaraan otomatis dengan menerapkan metode *Background Subtraction*.
2. Mengirim data kendaraan berupa id kendaraan, arah kendaraan masuk atau keluar, tanggal dan waktu di jalan tol ke dalam basis data *MySQL*.

1.5 Manfaat

Manfaat dari penelitian ini adalah sebagai terobosan baru dalam memonitoring dan menghitung jumlah kendaraan yang melewati jalan tol, terutama jalan tol Jatingaleh Semarang, serta mengurangi *human eror* dalam penghitungan manual jumlah kendaraan yang kurang efisien.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan oleh penulis dalam pembuatan laporan tugas akhir adalah sebagai berikut :

BAB 1 : PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang pemilihan judul tugas akhir “Perancangan Sistem Penghitungan Kendaraan Dua Arah Dengan Metode *Background Subtraction* di Jalan Tol Jatingaleh Semarang”, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan.

BAB 2 : TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini memuat dasar teori yang berfungsi sebagai sumber atau alat dalam memahami permasalahan yang berkaitan dengan Perancangan Sistem Penghitungan Kendaraan Dua Arah di Jalan Tol Dengan Metode *Background Subtraction*

BAB 3 : METODE PENELITIAN

Bab ini mengungkapkan proses tahapan-tahapan penelitian dimulai dari Analisa kebutuhan sistem, kemudian perancangan sistem hingga *prototype* jadi dibuat.

BAB 4 : HASIL DAN ANALISIS PENELITIAN

Bab ini mengungkapkan hasil penelitian yakni hasil dari pembuatan *prototype* sistem penghitung jumlah kendaraan dua arah di jalan tol serta analisa kinerja sistem.

BAB 5 : KESIMPULAN DAN SARAN

Bab ini mengungkapkan kesimpulan isi dari keseluruhan uraian bab-bab sebelumnya dan saran yang diperoleh dan diharapkan dapat bermanfaat dalam pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Citra merupakan suatu representasi (gambaran), kemiripan atau imitasi dari suatu objek. Citra digunakan sebagai *output* suatu sistem perekaman data dapat bersifat optic seperti foto, dapat bersifat analog seperti video pada televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan (Sutoyo, 2009). Proses pengolahan citra pada pustaka *OpenCV (Open Computer Vision)* telah dikembangkan beberapa metode yang dapat digunakan untuk memproses pengenalan objek bergerak.

Penelitian sebelumnya meneliti tentang sistem pendeteksi kendaraan dalam *Intelligent Transport System (ITS)* dengan metode *Viola-Jones* dapat menghasilkan akurasi sebesar 92,86%, akurasi terendah sebesar 52,78%. Metode *Viola-Jones* dapat digunakan untuk mendeteksi kendaraan (Harwendhani, 2016). Penelitian lainnya meneliti tentang sistem *tracking* dan klasifikasi kendaraan dengan metode *Background Subtraction* dapat menghasilkan akurasi sebesar 96% dan akurasi terendah 86%. (Sayidul, 2019)

Berdasarkan hasil dari penelitian sebelumnya maka, pada penelitian kali ini membuat perancangan sistem pendeteksi kendaraan dua arah di jalan tol menggunakan metode *Background Subtraction* sebagai algoritma pemisahan antara objek di depan (*foreground*) dan *background* serta *tracking* kendaraan untuk mengetahui arah kendaraan yang terdeteksi.

2.2 Dasar Teori

2.2.1 Computer Vision

Computer Vision merupakan cabang ilmu pengetahuan yang mempelajari bagaimana sistem dapat mengenali objek yang diamati dan dapat menambahkan perintah sesuai apa yang diinginkan. Pada ilmu ini mempelajari bagaimana mesin mengekstrak informasi dari suatu gambar dan menggunakannya untuk menyelesaikan tugas tertentu. Data gambar dapat diambil dari berbagai bentuk seperti gambar video atau gambar dari berbagai sudut (Purwanto, 2015). Dengan

cabang ilmu ini dapat dikembangkan lagi untuk membantu kegiatan manusia serta dapat mengatasi permasalahan seperti penghitungan jumlah kendaraan di jalan tol.

2.2.2 Open CV

OpenCV (Open Source Computer Vision) merupakan pustaka *computer vision* yang digagas oleh *intel* pada tahun 1999. Pustaka ini didesain untuk digunakan diberbagai *platform* dengan berfokus pada pemrosesan gambar secara *real-time* dengan berbagai algoritma *computer vision* yang bebas dari hak cipta sehingga dapat dikembangkan kembali secara bebas. *OpenCV* telah disediakan untuk dapat digunakan diberbagai bahasa pemrograman seperti bahasa : C, C++, Python dan Android.



Gambar 2.1 OpenCV

2.2.3 Preprocessing

Preprocessing merupakan tahapan awal sebelum melakukan pendeteksian. *Preprocessing* digunakan untuk menyiapkan beberapa komponen yang dibutuhkan. Terdapat tiga komponen dalam *preprocessing* yaitu: komponen pertama citra RGB (*red, green, blue*) yang nantinya akan diubah menjadi citra *greyscale*, menentukan daerah *region of interest* untuk menentukan daerah mana yang akan dilakukan pendeteksian dan membuat garis hitung untuk menghitung jumlah kendaraan yang melewati garis tersebut.

a. Grayscale

Grayscale adalah citra kanal dalam piksel (x, y) yang mendefinisikan tingkatan dari hitam ke putih. *Grayscale* mempunyai intensitas 256 level artinya skala abu-abu dari 0-255, dalam intensitas 0 mendefinisikan warna hitam dan intensitas 255 mendefinisikan warna putih (Sayidul, 2019). Proses konversi citra warna menjadi *grayscale* menggunakan Persamaan 1 (Harwendhani, 2016).

$$X = (0,299 * R) + (0,587 * G) + (0,114 * B) \quad (1)$$

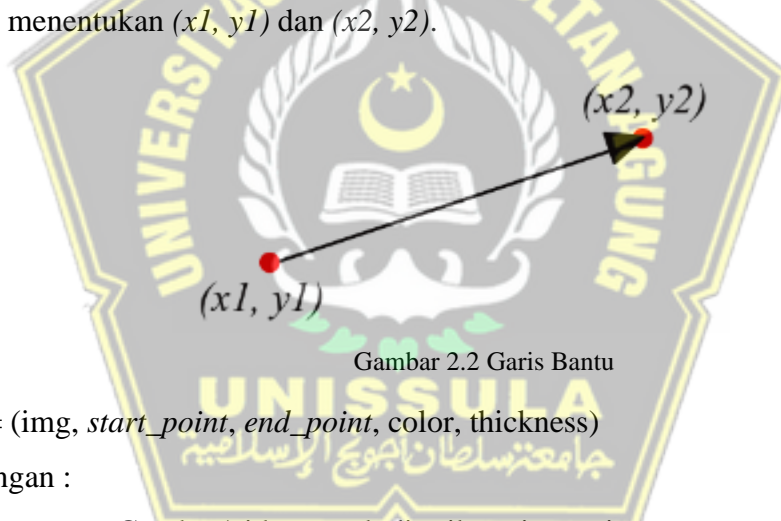
Keterangan :

- X : Nilai *grayscale*
 R : Nilai intensitas *red*
 G : Nilai intensitas *green*
 B : Nilai intensitas *blue*

Greyscale merupakan tahapan awal dari *preprocessing* yang berfungsi untuk mengubah citra warna ke citra keabuan serta mempermudah untuk tahap pemisahan antara objek (*foreground*) dan *background*.

b. **Garis Bantu**

Merupakan *code* tambahan untuk mempermudah dalam perhitungan jumlah kendaraan masuk maupun keluar dengan memanfaatkan pustaka dari *OpenCV* dengan menentukan $(x1, y1)$ dan $(x2, y2)$.



Gambar 2.2 Garis Bantu

Garis = (*img, start_point, end_point, color, thickness*)

Keterangan :

- Img* : Gambar/video untuk ditarik garis nantinya
Start point : Koordinat awal garis, dipresentasikan oleh $(x1, y1)$
End_point : Koordinat awal garis, dipresentasikan oleh $(x2, y2)$
Color : Warna direpresentasikan oleh RGB (255,0,0)
Thickness : Ketebalan garis

Dalam pembuatan garis bantu langkah pertama kali yaitu mengetahui *resolusi output* kamera yang dipakai berguna untuk menentukan *start_point* dan *end point* kemudian menentukan warna dan ketebalan garis tersebut.

c. **Region of Interest (ROI)**

Region of Interest (ROI) adalah bagian citra yang dipilih untuk di proses (Pratomo, 2020). Sistem pendeteksi kendaraan dua arah di jalan tol dilakukan dengan menempatkan daerah ROI pada area lajur keluar atau masuknya kendaraan. Kendaraan yang melintas pada area ROI tersebut yang akan dihitung dan dideteksi tanpa memisahkan area ROI dari *frame* aslinya.

2.2.4 Segmentasi

Segmentasi adalah proses yang bertujuan untuk memisahkan antara objek (*foreground*) dengan objek *background*. Pengurangan latar belakang sebagai langkah pertama dalam fase segmentasi untuk mengisolasi objek bergerak. Pengurangan latar belakang diikuti oleh serangkaian operasi penyaringan untuk menghilangkan artefak segmentasi. Pada penelitian ini menggunakan segmentasi *Background Subtraction*.

a. *Background Subtraction*

Background Subtraction adalah metode untuk mendeteksi gerakan berdasarkan perbedaan antara frame video saat ini dan latar belakang kosong. Metode ini mengharuskan gambar latar belakang kosong diperoleh sebelumnya. Dengan mengambil *frame* saat ini dan mengurangi latar belakangnya, piksel demi piksel, hasilnya akan menjadi *masking* yang menunjukkan perubahan. Jika intensitas *masking* ini melampaui *threshold* (ambang batas) yang telah ditentukan, maka itu dianggap sebagai gerakan.

Background Subtraction sering digunakan dalam aplikasi berbasis pengolahan citra. Cara kerja dari Algoritma *Background Subtraction* dapat dilihat pada Persamaan 2 (Sayidul, 2019).

$$S(n) = |F(n) - B| > T \quad (2)$$

Keterangan :

$S(n)$: Selisih *frame* ke- n

$F(n)$: *Frame* ke- n

B : Citra *Background*

T : *Threshold*

Pada penelitian ini metode *Background Subtraction* digunakan untuk memisahkan antara objek (*foreground*) dan *background*, dimana dalam proses

segmentasi pengolahan citra menjadi *greyscale* serta memanfaatkan nilai ambang batas (*threshold*) yang sudah ditentukan.

2.2.5 Object Filtering

Object filtering adalah proses untuk menonjolkan objek dari hasil segmentasi pada citra, sehingga dapat lebih mudah untuk membedakan objek dengan lainnya.

a. *Thresholding*

Thresholding adalah salah satu teknik segmentasi yang digunakan untuk pembeda nilai ambang batas yang signifikan antara objek dan latar belakang (Ardhianto, 2013). Dalam pelaksanaannya, *thresholding* membutuhkan nilai yang digunakan sebagai ambang batas antara objek dengan latar belakang.

b. *Dilation*

Dilation merupakan pengolahan citra morfologi OpenCV yang bertujuan untuk memodifikasi struktur geometrik pada citra. *Dilation* merupakan perbaikan citra dari interaksi *kernel* yang dapat memperjelas objek dari hasil segmentasi dengan memanfaatkan nilai piksel tetangganya. *Dilation* dihitung dengan menggunakan persamaan 3 (Menegaz, 2017).

$$A \oplus B = \{z | (B)_z \cap A \neq \emptyset\} \quad (3)$$

Keterangan :

A : Citra masukan

B : Citra *structuring element*

z : Nilai piksel bagian dari citra A

2.2.6 Deteksi Objek (Kendaraan)

Deteksi Objek adalah teknologi komputer yang terkait dengan visi komputer dan pemrosesan gambar. Pada penelitian ini pendeteksian objek dilakukan untuk mengenali objek yang bergerak seperti manusia, mobil dan lainnya.

Tujuan Deteksi Objek adalah mendeteksi semua instance objek dari yang diketahui, seperti orang, mobil atau wajah dalam suatu gambar. Biasanya hanya sedikit contoh objek hadir di dalam gambar, tetapi ada banyak kemungkinan lokasi dan skala dimana mereka dapat terjadi dan bagaimana dieksplorasi. Setiap pendeteksian objek dilaporkan dalam beberapa bentuk informasi, biasanya lokasi objek dan skala objek didefinisikan dalam kotak pembatas.

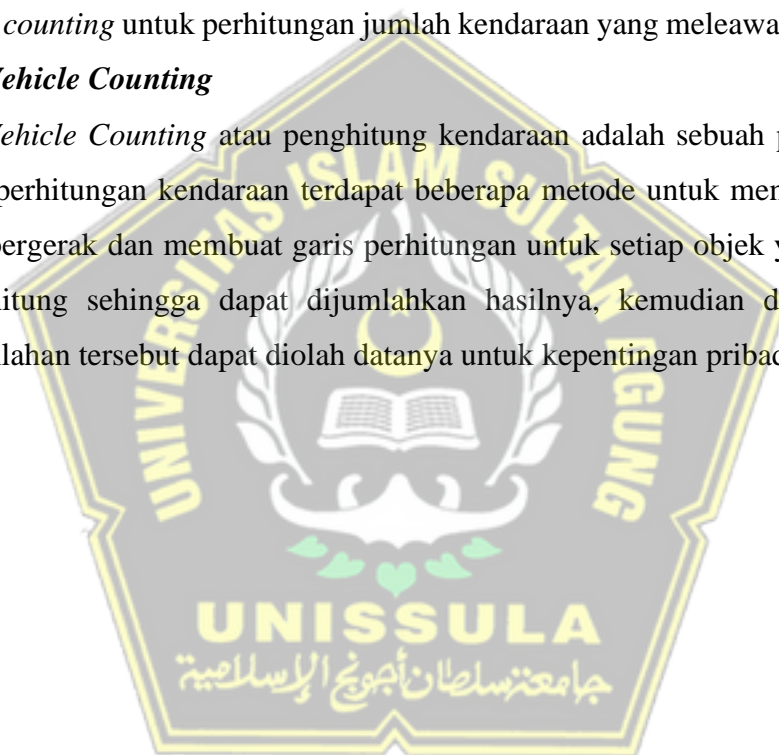
Deteksi objek bertujuan untuk memisahkan objek atau (*foreground*) dan latar belakang atau (*background*). Biasanya untuk melakukan deteksi objek, akan dibuat sebuah model latar pada sebuah citra berberak berdasarkan nilai pixel yang masuk seiring waktu (M. Piccardi, 2004).

2.2.7 Object Tracking

Object Tracking atau pelacakan objek adalah sebuah sistem dimana dalam objek yang telah terdeteksi selanjutnya dilakukan *tracking object* supaya objek yang terdeteksi mendapatkan akurasi yang tepat yang nantinya akan dilakukan *vehicle counting* untuk perhitungan jumlah kendaraan yang melewati garis hitung.

2.2.8 Vehicle Counting

Vehicle Counting atau penghitung kendaraan adalah sebuah proses dimana dalam perhitungan kendaraan terdapat beberapa metode untuk mengenali sebuah objek bergerak dan membuat garis perhitungan untuk setiap objek yang melewati garis hitung sehingga dapat dijumlahkan hasilnya, kemudian dari data hasil penjumlahan tersebut dapat diolah datanya untuk kepentingan pribadi atau umum.



BAB III

METODE PENELITIAN

3.1 Bahan Penelitian

Bahan penelitian dalam penelitian ini adalah dataset video di jalan tol Jatingaleh, Semarang. Terdapat dataset video sample berdurasi 1 menit untuk menguji tingkat keakurasian dan ketepatan pendeteksian yaitu dataset diambil menggunakan kamera dengan kondisi penerangan yang cukup yang dilakukan pada pagi, siang dan sore.

3.2 Alat Penelitian

Alat yang dibutuhkan dalam penelitian ini adalah berupa perangkat keras dan perangkat lunak dengan spesifikasi sebagai berikut :

1. Perangkat Keras
 - a. Komputer
 - b. Kamera / Webcam
2. Perangkat Lunak
 - a. Database server *MySql*

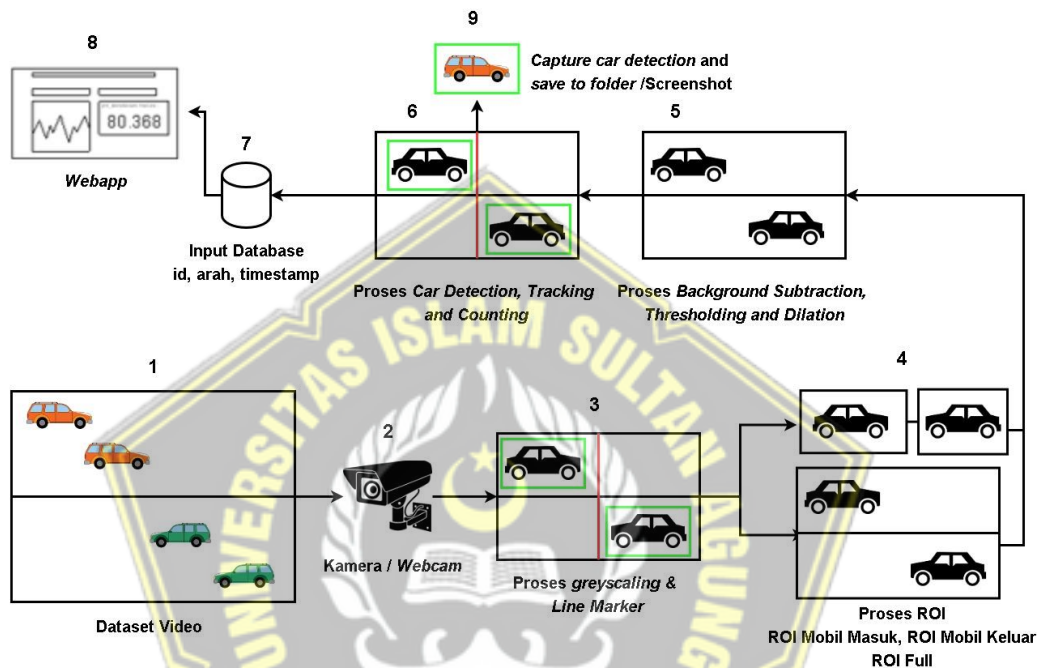
3.3 Pengumpulan Data

Pengumpulan data yang dilakukan peneliti untuk penyusunan tugas akhir adalah:

1. Video kendaraan yang melewati jalan Tol Jatingaleh Semarang yang merupakan data yang harus dipenuhi sebagai sampel dalam pengujian untuk dilakukan pendeteksian kendaraan dan penghitung kendaraan (*Car detection and Car Counting*) di Jalan Tol Jatingaleh Semarang.
2. Studi Literatur digunakan untuk mencari dasar teori sebagai referensi dalam penyusunan laporan penelitian yang dapat memudahkan peneliti dalam membangun prototipe yang bersangkutan dengan metode *Background Subtraction*. Sumber yang dijadikan literatur yaitu jurnal, karya ilmiah dan

situs-situs untuk mempermudah dalam peneliti membuat laporan dan prototipe.

3.4 Konsep Sistem



Gambar 3.1 Konsep Sistem

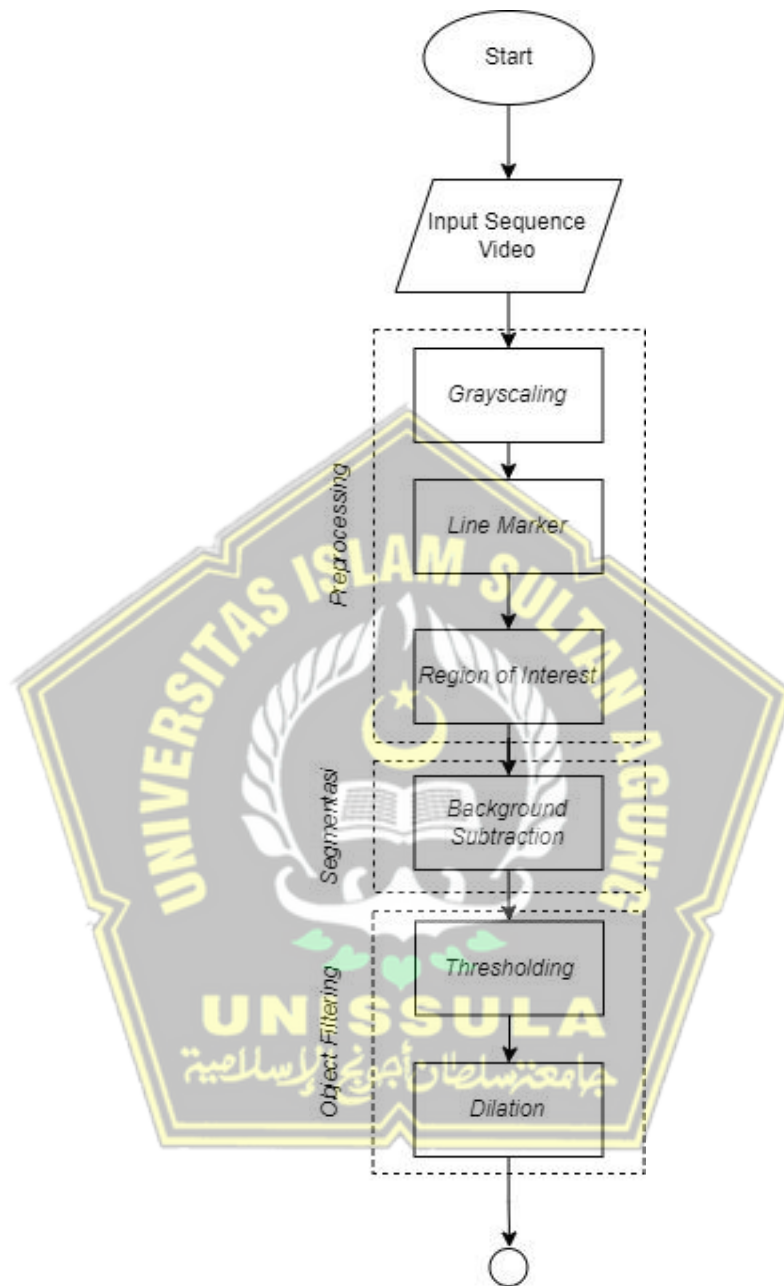
Gambar 3.1 menunjukkan gambaran konsep sistem keseluruhan dari sistem penghitung jumlah kendaraan dua arah di jalan tol. berikut keterangan Gambar 3.1:

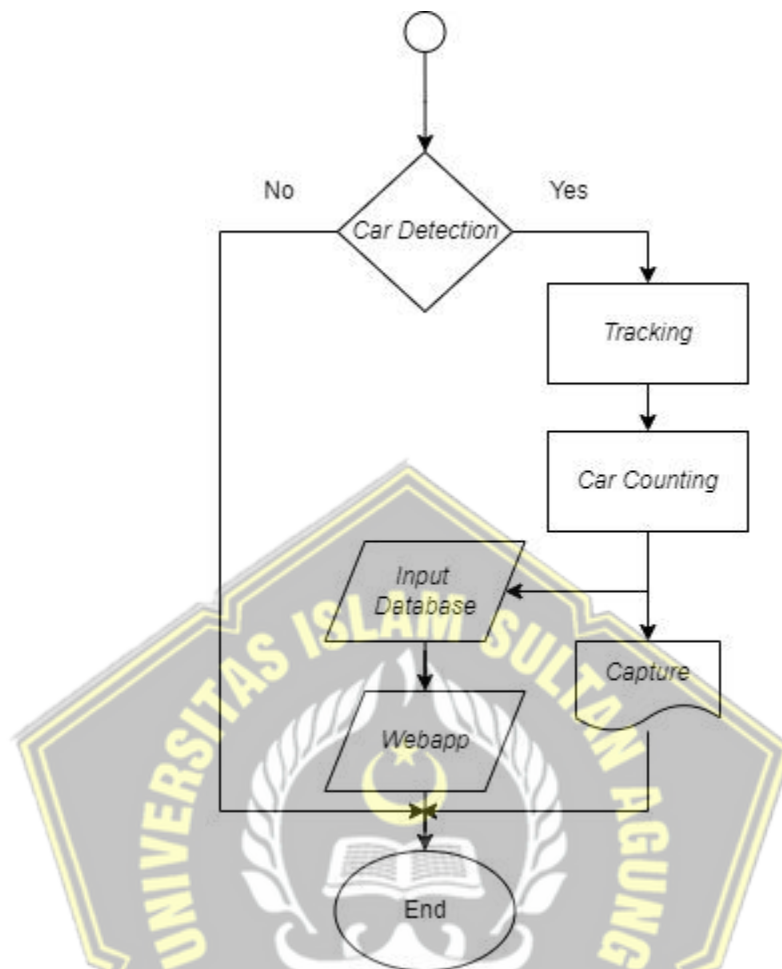
1. *Dataset* video yang diambil dari jalan tol Jatingaleh Semarang
2. Proses pendeteksian objek bergerak yang terdeteksi oleh kamera.
3. Proses *greyscaling* dan pembuatan *line marker* untuk garis hitung
4. Proses *Region of Interest* untuk membagi 3 bagian pendeteksian kendaraan yaitu: kendaraan masuk, kendaraan keluar dan kendaraan dua arah
5. Proses pemisahan antara objek (*foreground*) dan *background* menggunakan metode *Background Subtraction* kemudian mengatur ambang batas (*threshold*) dan selanjutnya proses penebalan (*dilation*).

6. Proses pengenalan objek setelah melakukan proses segmentasi dan *object filtering* kemudian *tracking object* dan menghitung jumlah kendaraan (*counting*) dengan garis hitung.
7. Proses penyimpanan data yang telah dihitung dan arah kendaraan kedalam *database*
8. Tampilan akhir *webapp* grafik data kendaraan hasil pengolahan data dari *database*
9. *Capture object* apabila objek yang terdeteksi melewati garis hitung.



3.5 Flowchart Sistem





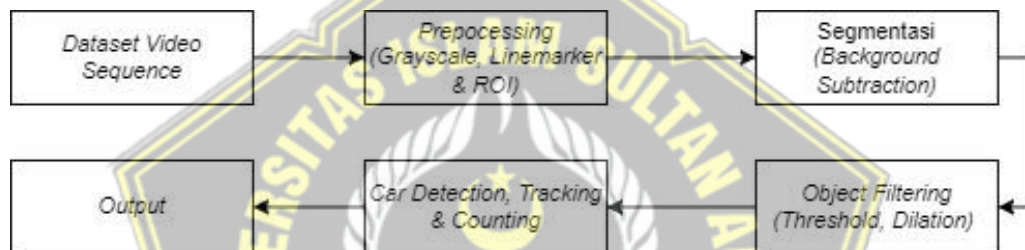
Gambar 3.2 *Flowchart* Sistem

Pada Gambar 3.2 *flowchart* diatas dijelaskan skema untuk mendapatkan bagian citra yang dikehendaki sebagai objek deteksi, mulanya pada tahapan memasukan dataset *video/sequence* yang kemudian diolah pada tahap *preprocessing* merupakan tahapan awal melakukan penyaringan awal terhadap bagian dari frame video dengan tujuan membuat citra keabuaan dan memntukan daerah ROI serta membuat garis bantu bantu yang nantinya digunakan untuk menghitung kendaraan yang lewat. Kemudian pada tahapan segmentasi merupakan tahapan untuk memisahkan antara sub-citra yang mengandung objek atau tidak dengan menggunakan metode *background subtraction*. Kemudian apabila object yang dikehendaki sudah terdeteksi tahapan selanjutnya yaitu melakukan *object filtering* dengan tujuan untuk mengurangi noise dan penebalan pada citra. Kemudian proses *tracking & counting* proses ini bertujuan untuk *tracking* objek

pendeteksian serta melakukan penghitungan object kendaraan yang terdeteksi. Kemudian object yang terdeteksi akan di *capture* dan disimpan pada folder *capture* serta objek yang terdeteksi akan dihitung datanya dan dimasukkan kedalam *database* yang nantinya akan diolah untuk pembuatan *webapp*.

3.6 Metode

Metode yang digunakan pada penelitian ini adalah metode *background subtraction and blob detection*. *Dataset* yang digunakan berupa video yang diambil menggunakan kamera/webcam. Adapun urutan proses metode dapat dilihat pada Gambar 3.3 dibawah ini:

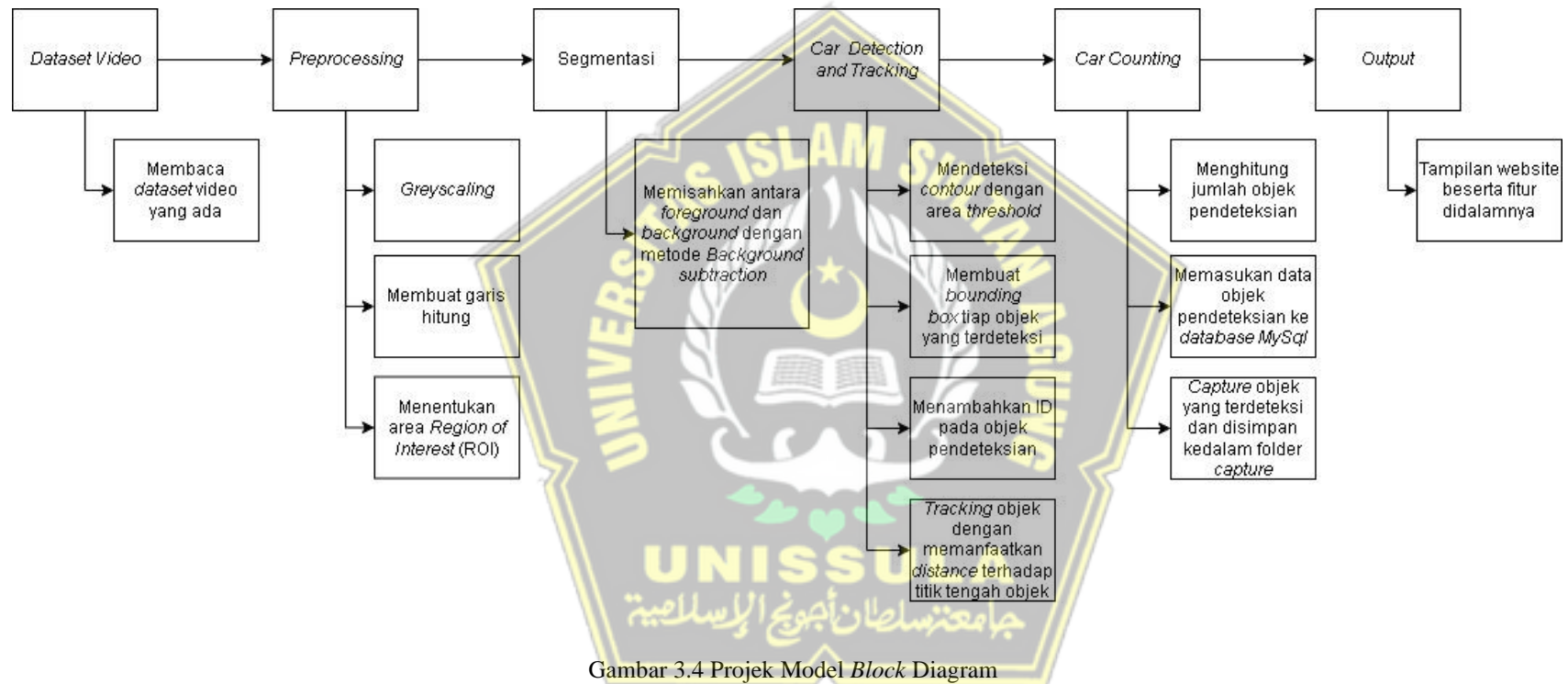


Gambar 3.3 Metode sistem

Gambar 3.3 merupakan proses *Background subtraction* yang diterapkan dalam sistem. Langkah pertama yaitu pengambilan *dataset* video yang diambil menggunakan kamera kemudian dilakukan *grayscaleing* untuk membantu dalam pemisahan antara *background* dan *foreground*. Kemudian dilakukan pemisahan antara *background* dan *foreground*. Pada tahap *Object Filtering* dilakukan untuk mengurangi noise dan mempertebal objek sebelum dilakukan pendeteksian dan *tracking* objek, selanjutnya hasil output pada metode ini berupa data yang akan di *input* kedalam *database MySQL* yang nantinya akan diolah untuk kebutuhan *website*.

3.7 Analisa Kebutuhan Sistem

3.7.1 Projek Model *Block Diagram*



Pada Gambar 3.4 adalah projek model *block* diagram, dimana dalam *block* diagram menggambarkan fungsi dan hubungan timbal balik dari suatu sistem. Terdapat beberapa urutan fungsional sistem yang memiliki induk dari parameter yang dibutuhkan oleh sistem untuk melakukan pendeteksian hingga menghasilkan *output* sistem.

3.7.2 Perancangan Database

Untuk membuat prototipe ini, hal selanjutnya yang perlu dibuat adalah *database*. *Database* dirancang untuk membantu menjalankan fungsi pada sistem yaitu menyimpan data yang berkaitan dengan sistem. Adapun perancangan *database* pada prototipe sistem pendeteksi kendaraan adalah :

Tabel 3.1 Tabel Perancangan Database

Field	Tipe	Keterangan
Id	Int(11)	Primary Key, AUTO INCREMENT
Arah	Varchar(100)	Digunakan untuk jenis arah hasil pendeteksian objek
Tanggal	Date	Current_timestamp()
Waktu	Time	Current_timestamp()

Pada Tabel 3.1 adalah tabel pendeteksian kendaraan yang berfungsi menampilkan jumlah kendaraan, arah kendaraan dan waktu saat pendeteksian dilakukan. Tabel terdiri dari id, arah, tanggal dan waktu.

3.8 Metode Pengujian

3.8.1 Mean Square Error (MSE)

Mean Square Error (MSE) merupakan metode pengujian yang digunakan sebagai pemeriksaan antara nilai yang dihasilkan oleh sistem yang tercemar dengan kesalahan dengan nilai yang seharusnya dicapai oleh sistem, dengan cara memberikan skor kuantitatif yang menggambarkan tingkat kemiripan atau kesalahan diantara keduanya, MSE dihitung dengan menggunakan Persamaan 4 (Wang, 2009).

$$MSE(x, y) = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2 \quad (4)$$

Keterangan:

N : Jumlah pengujian *dataset*

(x, y) : Perbedaan antara nilai x dan y pada *dataset*

i : Jumlah percobaan

x_i : Nilai yang dihasilkan oleh sistem yang tercemar dengan kesalahan

y_i : Nilai yang seharusnya dicapai oleh sistem

Dalam pengujian MSE nilai *error* dapat dikatakan baik apabila memiliki nilai *error* paling rendah.

3.8.2 Confusion Matrix

Confusion Matrix adalah metode pengujian yang digunakan untuk menggambarkan penilaian pengelompokan. Beberapa metode pengujian *confusion matrix* seperti *precision*, *recall* dan akurasi. Tabel 3.2 merupakan gambaran dari *confusion matrix* (Adinugroho, 2018).

Tabel 3.2 Tabel *Confusion Matrix*

		Nilai yang sebenarnya	
		TRUE	FALSE
Hasil Prediksi	POSITIVE	TP (<i>True Positive</i>)	FP (<i>False Positive</i>)
	NEGATIVE	TN (<i>True Negative</i>)	FN (<i>False Negative</i>)

Keterangan:

TP : Nilai yang seharusnya benar dan Nilai yang dihasilkan sistem benar

TN : Nilai yang seharusnya benar dan Nilai yang dihasilkan sistem salah

FP : Nilai yang seharusnya salah dan Nilai yang dihasilkan sistem benar

FN : Nilai yang seharusnya salah dan Nilai yang dihasilkan sistem salah

a. *Precision*

Precision dapat ditentukan dari jumlah keseluruhan nilai yang seharusnya benar dan nilai yang dihasilkan sistem benar dibagi dengan keseluruhan nilai yang dihasilkan sistem benar. Nilai *precision* tertinggi jika nilai yang dihasilkan sama dengan 100% dan Nilai *precision* terendah jika nilai yang dihasilkan sama dengan

0%. Pengujian *precision* dihitung dengan menggunakan Persamaan 5 (Adinugroho, 2018).

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (5)$$

Keterangan:

TP : Nilai yang seharusnya benar dan Nilai yang dihasilkan sistem benar

FP : Nilai yang seharusnya salah dan Nilai yang dihasilkan sistem benar

Pengujian *precision* digunakan untuk menghitung presentase nilai yang dihasilkan benar dari total keseluruhan nilai prediksi benar

b. *Recall*

Recall ditentukan dari jumlah keseluruhan nilai yang seharusnya benar dan nilai yang dihasilkan sistem benar dibagi dengan *True Positive* ditambah *False Negative*. Nilai *recall* tertinggi apabila nilai yang dihasilkan sama dengan 100% dan Nilai *recall* terendah jika nilai yang dihasilkan sama dengan 0%. Pengujian *recall* dihitung dengan menggunakan Persamaan 6 (Adinugroho, 2018).

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (6)$$

Keterangan:

TP : Nilai yang seharusnya benar dan Nilai yang dihasilkan sistem benar

FN : Nilai yang seharusnya salah dan Nilai yang dihasilkan sistem salah

Pengujian *recall* digunakan untuk menghitung presentase dari total kendaraan terdeteksi yang memiliki nilai positif.

c. Akurasi

Akurasi ditentukan dari nilai yang seharusnya benar dibagi dengan keseluruhan data. Nilai akurasi tertinggi apabila nilai akurasi yang dihasilkan sama dengan 100% dan Nilai akurasi terendah apabila nilai akurasi yang dihasilkan sama dengan 0%. Akurasi dihitung dengan menggunakan Persamaan 7 (Adinugroho, 2018).

$$Akurasi = \frac{TP+TN}{TP+TN+FN+FP} \times 100\% \quad (7)$$

Keterangan:

TP : Nilai yang seharusnya benar dan Nilai yang dihasilkan sistem benar

TN : Nilai yang seharusnya benar dan Nilai yang dihasilkan sistem salah

FP : Nilai yang seharusnya salah dan Nilai yang dihasilkan sistem benar

FN : Nilai yang seharusnya salah dan Nilai yang dihasilkan sistem salah

Pengujian akurasi digunakan untuk menghitung presentase nilai prediksi yang benar tanpa mempertimbangkan yang nilai positif dan negative.

3.8.3 Evaluasi Kinerja Sistem

Evaluasi merupakan proses untuk mempermudah dalam menarik kesimpulan. Evaluasi didapatkan setelah pengujian dan analisa terhadap sistem. Dalam penelitian ini dilakukan dua evaluasi kinerja sistem yaitu, evaluasi membandingkan jumlah objek yang seharusnya dengan jumlah objek yang dihasilkan sistem dan evaluasi kesalahan dalam mendeteksi objek.

3.9 Instalasi Sistem

Untuk membangun prototipe agar berjalan dengan lancar, maka diperlukan untuk menginstall beberapa *library python* yaitu :

1. *OpenCV* v4.1.2

OpenCV (Open Source Computer Vision) merupakan pustaka *computer vision* yang bertujuan untuk pengolahan citra. Metode yang digunakan berdasarkan penggunaan *open source library OpenCV*.

2. *PyMySQL* v1.0.2

PyMysql merupakan library klient *MySQL python*. *MySQL* adalah sistem manajemen *database open source* terkemuka dengan manajemen basis data *multiuser* dan *multithreaded* system. Library *PyMySQL* digunakan untuk mengkoneksikan data dapat dikirimkan kedalam *database MySQL*.

3. *Numpy* v1.18.1

Numpy merupakan *library python* yang menyediakan objek *array* multidimensi, berbagai objek turunan dan operasi cepat pada *array*. *Numpy* digunakan untuk bekerja dengan data berbentuk *array*.

4. *Tkinter* v.0.1.0

Tkinter merupakan *library python* yang digunakan untuk membuat *GUI* atau antarmuka pada *python*.

3.10 Perancangan Prototipe

Perancangan desain prototipe memiliki tujuan untuk menyampaikan rancangan mengenai sistem yang akan dibangun melalui pendeteksi dan penghitung kendaraan dengan di Jalan Tol Jatingaleh Semarang. Pada perancangan desain *prototype* memiliki dua *interface* atau tampilan yaitu tampilan sistem dan tampilan website, berikut rancangan prototipe yang akan dibuat:

a. Perancangan Desain Sistem



Gambar 3.5 Desain Sistem

Pada Gambar 3.5 merupakan perancangan desain sistem yang berisikan *button* untuk menjalankan *method* sistem. Terdapat empat *button* dalam perancangan desain sistem yaitu: *button* keluar atau pendeteksian lajur keluar kendaraan, *button* masuk atau pendeteksian lajur masuk kendaraan, *button* dua arah atau pendeteksian lajur dua arah dan *button* *open capture* berguna untuk melihat hasil *capture* setelah dilakukan pendeteksian.

b. Perancangan Desain *Web*Gambar 3.6 Desain *Web*

Pada Gambar 3.6 merupakan *dashboard* dari perancangan desain web dimana setelah sistem dijalankan maka data (id, arah, tanggal dan waktu) akan otomatis masuk kedalam *database* yang kemudian diolah dan ditampilkan kedalam website sebagai *monitoring*.

BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 Cara Kerja Sistem

Pada tahap metodologi proses akan dijelaskan bagaimana langkah-langkah atau urutan bagaimana cara kerja sistem.

4.1.1 Pengambilan *Dataset*

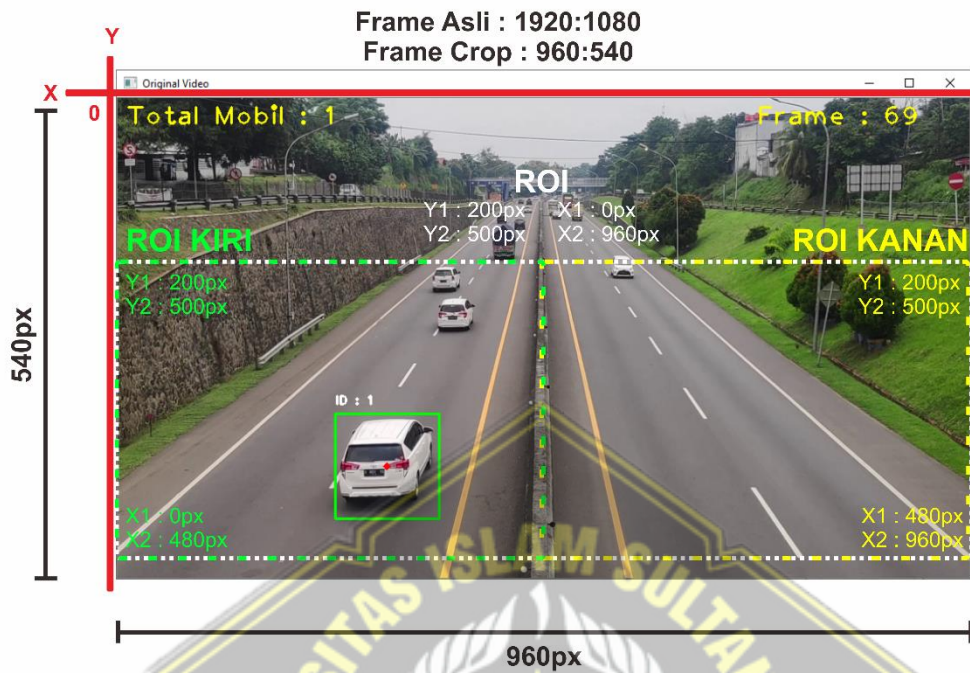
Pada tahap ini, pengambilan *dataset* dilakukan pada Jalan Tol Jatingaleh Semarang. Dimana *dataset* yang digunakan berdurasi 1menit dengan resolusi 1920x1080 piksel. Dalam proses pengambilan *dataset* video ini diperlukan adanya *tripod* kamera agar citra yang dihasilkan stabil. Kemudian sistem akan membaca *frame* atau citra yang tertangkap oleh kamera selanjutnya dilakukan *looping* atau perulangan hingga *frame* terakhir.

4.1.2 Penentuan Daerah ROI

Pada tahap sebelum menentukan ROI, karena *dataset* yang digunakan beresolusi 1920x1080 piksel sedangkan resolusi layar monitor yang digunakan peneliti yaitu 1366x768 maka dilakukan *resize* terhadap *dataset* yang digunakan sebanyak 0,5 tiap panjang dan lebarnya.

Proses ROI (*Region of Interest*) dalam penelitian ini digunakan untuk menentukan area lajur keluar masuknya kendaraan yang akan di proses. Penentuan area lajur keluar masuknya kendaraan berfungsi untuk objek yang berada diluar area ROI tidak menjadi *noise* dan tidak akan di proses serta penentuan ROI dapat memfokuskan pendeteksian objek yang diinginkan. Pada penelitian ini ROI yang digunakan nantinya akan dibagi menjadi 3 yaitu roi untuk frame keseluruhan, roiKiri untuk pendeteksian kendaraan keluar, dan roiKanan untuk pendeteksian kendaraan masuk.

```
# Extract Region of interest
roi = frame[200: 500, 0: 960]
roiKiri = frame[200: 500, 0: 480]
roiKanan = frame[200: 500, 480: 960]
```



Gambar 4.1 Extract ROI

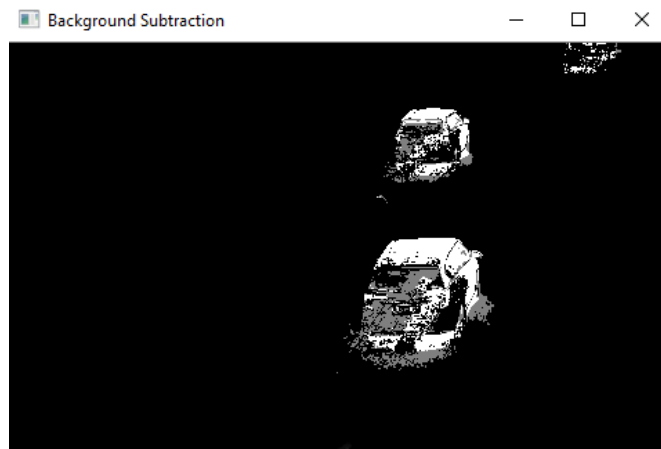
Pada gambar 4.1 merupakan proses *extract* ROI dari frame original menghasilkan ROI Kiri atau arah kendaraan keluar, ROI Kanan atau arah kendaraan masuk dan ROI atau kendaraan 2 arah.

4.1.3 Proses Pengolahan Citra

Area ROI yang sudah ditentukan sebelumnya agar area yang dilakukan proses pengolahan citra berfokus pada *frame* ROI bukan pada keseluruhan *frame*. Pengolahan citra merupakan sebuah disiplin ilmu tentang memanipulasi sebuah citra untuk meningkatkan kualitas dari citra (Kurniawan, 2015). Dalam pelaksanaannya dilakukan beberapa tahapan cara yang di sebut teknik pengolahan citra (Pratomo, 2020).

Pengolahan citra yang digunakan pada peneliti terdapat 3 tahapan yaitu:

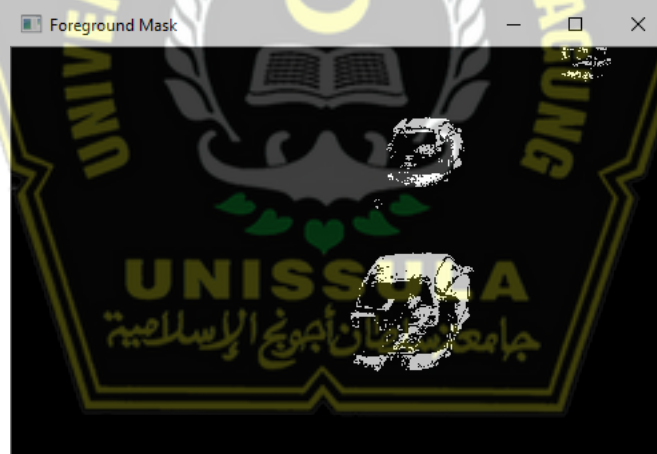
1. Tahapan Awal yaitu pemisahan objek dengan latar belakangnya dengan memanggil *library* dari *OpenCV* yaitu *Background Subtraction MOG2*.



Gambar 4.2 Pengolahan Citra *Background Subtraction*

Pada Gambar 4.2 merupakan hasil dari pemisahan objek dengan latar belakangnya dengan menggunakan *Background Subtraction MOG2*.

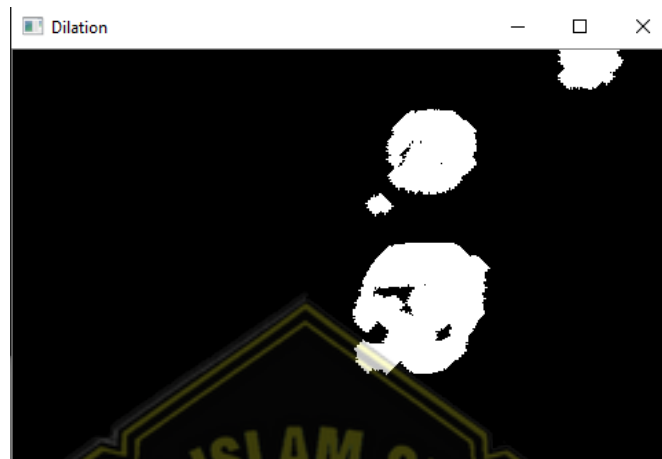
2. Tahapan selanjutnya yaitu penggunaan *Thresholding* (ambang batas), untuk memepertegas objek serta menghilangkan *shadow* atau warna keabu-abuan (hanya warna hitam dan putih) seperti pada Gambar 4.5



Gambar 4.3 Pengolahan Citra *Threshold*

Pada gambar 4.3 merupakan hasil dari penggunaan *Thresholding* pada tahapan *object filtering*.

3. Tahap terakhir yaitu penggunaan *Dilation (Morphology)* untuk penebalan objek yang terdeteksi supaya mendapatkan *masking* yang maksimal seperti pada Gambar 4.6



Gambar 4.4 Pengolahan Citra *Dilation*

Pada gambar 4.6 merupakan hasil dari penggunaan *dilation* pada tahapan *object filtering*.

4.1.4 Proses Pendeteksian Objek

Proses pendeteksian objek dilakukan dengan memanggil *library* dari *OpenCV* yaitu *Contouring*. Fungsi *Contour* digunakan untuk mendeteksi dan mengenali suatu objek dengan menggabungkan titik-titik yang terbentuk pada suatu piksel. Pada penelitian ini *contour* di setting sebesar >7000 pixel untuk menghiraukan titik-titik kecil yang dapat mengganggu pendeteksian objek. Objek yang terdeteksi ditandai dengan *bounding box* dengan posisi *center bounding box* koordinat “X” dan “Y” yang dapat secara otomatis mengikuti setiap gerakan objek.

4.1.5 Proses *Tracking Object and Capture*

Tracking adalah teknik untuk mengikuti pergerakan suatu objek. Fungsi *tracking* dilakukan dengan mendapatkan titik tengah objek agar dapat disimpan dalam suatu *array*. Fungsi *tracking* dilaksanakan dengan memanggil *class* “*tracker.py*” dimana dalam *class* tersebut terdapat *method update*, *capture* dan *stop*.

1. *Method update* berguna untuk menghitung jarak / *distance* antara *center bounding box* dengan *new center bounding box* dan menambahkan id jumlah kendaraan tiap objek yang terdeteksi. Pada penelitian ini *method update*

menggunakan jarak / *distance* antara *center bounding box* dengan *new center bounding box* dengan jarak ≤ 150 piksel.

```
same_object_detected = False
```

```
for id, pt in self.center_points.items():
    dist = math.hypot(cx - pt[0], cy - pt[1])
    if dist <= 150:
        self.center_points[id] = (cx, cy)
        objects_bbs_ids.append([x, y, w, h, id])
        same_object_detected = True

# New object is detected we assign the ID to that object
if same_object_detected is False:
    self.center_points[self.id_count] = (cx, cy)
    objects_bbs_ids.append([x, y, w, h, self.id_count])
    self.id_count += 1
```

2. *Method capture berguna* untuk setiap *bounding box* objek yang terdeteksi pada sistem akan dilakukan *capture frame* jika nilai $cy \leq 210$ dan $cy \geq 170$.

```
if (cy <= 210 and cy > 170):
    self.flag[id] = 1
else:
    break

def capture(self, img, x, y, h, w, id):
    if (self.capflag[id] == 0):
        self.capflag[id] = 1
        self.flag[id] = 0
        crop_img = img[y:y + h, x:x + w]
        file = "Capture/Keluar/"+str(id)+".jpg"
        cv2.imwrite(file, crop_img)
```

3. *Method stop* digunakan untuk menutup sistem setelah proses pendeteksian dan *tracking* kendaraan telah selesai supaya nilai pendeteksian selanjutnya dimulai dari 0.

```
def stop(self):
    msg = print("Jumlah mobil di tambahkan :", self.id_count)
    sys.exit(msg)
```

4.1.6 Proses *Input Database*

Pada tahap ini objek yang terdeteksi pada *bounding box* memiliki id sendiri-sendiri setelah melalui proses *tracking*, kemudian dilakukan *counting* atau perhitungan jumlah kendaraan yang terdeteksi dan dimasukkan datanya kedalam *database mysql* dimana *query* yang digunakan adalah *INSERT INTO* dengan *Value (string)* sesuai dengan perintah, apabila $cx > 0$ dan $cx \leq 480$ maka *value* “Keluar” dan apabila $cx > 480$ dan $cx \leq 960$ maka *value* “Masuk” kemudian memanggil *class method* “*mysql.py*” yang merupakan *mysql connector* dalam *database* peneliti.

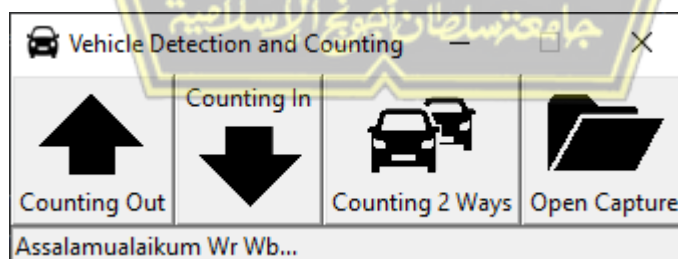
```

if (cx > 0 and cx <= 480):
    val = 'Keluar'
    sql = ("INSERT INTO cardetection (arah) VALUES (%s)")
    DB.insert(database, sql, val)
    print("Status : Keluar")
elif (cx > 480 and cx <= 960):
    val = 'Masuk'
    sql = ("INSERT INTO cardetection (arah) VALUES (%s)")
    DB.insert(database, sql, val)
    print("Status : Masuk")

```

4.2 Pembuatan Tampilan

4.2.1 Tampilan *GUI* Sistem

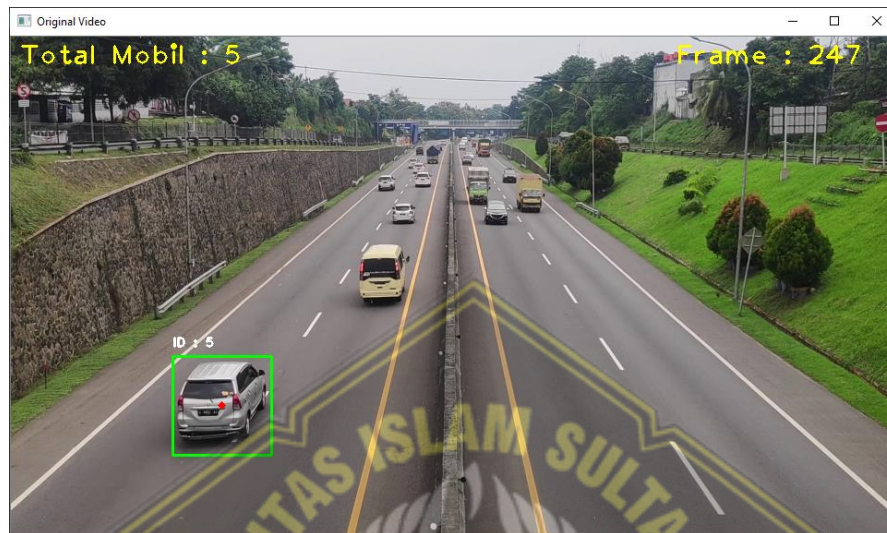


Gambar 4.5 Tampilan *GUI* Sistem

Pada Gambar 4.5 merupakan tampilan *interface* sistem penghitung kendaraan dua arah di jalan tol menggunakan *background subtraction*. Tampilan sistem dibuat menggunakan *library tkinter* pada *python*. Pada tampilan ini operator dapat mengoperasikan seluruh fitur yang ada, mulai dari perhitungan kendaraan keluar,

perhitungan kendaraan masuk, perhitungan kendaraan dua arah dan membuka hasil *capture*.

4.2.2 Tampilan Sistem Penghitung Kendaraan Keluar



Gambar 4.6 Output Original Frame ROI Kiri

Pada Gambar 4.6 merupakan tampilan *original frame* pada sistem penghitung kendaraan menggunakan ROI kiri.



Gambar 4.7 Output ROI Kiri

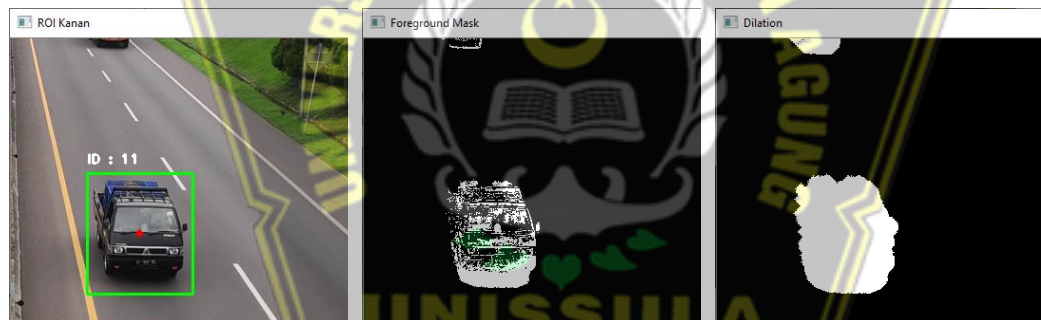
Pada Gambar 4.7 merupakan hasil dari *foreground mask*, *dilation* dan ROI kiri pada sistem penghitung kendaraan menggunakan ROI kiri.

4.2.3 Tampilan Sistem Penghitung Kendaraan Masuk



Gambar 4.8 Output Original Frame ROI Kanan

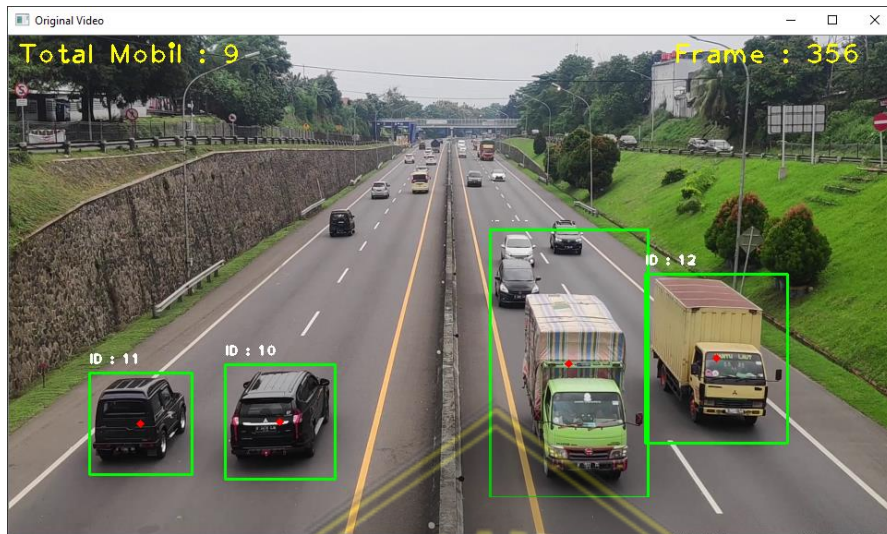
Pada Gambar 4.8 merupakan tampilan *original frame* pada sistem penghitung kendaraan menggunakan ROI kanan.



Gambar 4.9 Output ROI Kanan

Pada Gambar 4.9 merupakan hasil dari *foreground mask*, *dilation* dan ROI kanan pada sistem penghitung kendaraan menggunakan ROI kanan.

4.2.4 Tampilan Perhitungan Kendaraan 2 Arah



Gambar 4.10 Output Original Frame ROI

Pada Gambar 4.10 merupakan tampilan *original frame* pada sistem penghitung kendaraan menggunakan ROI dua arah untuk melakukan pendeteksian kendaraan secara bersamaan antara lajur kendaraan masuk dengan lajur kendaraan keluar.





Gambar 4.11 *Output ROI*

Pada Gambar 4.11 merupakan hasil dari *foreground mask*, *dilation* dan ROI pada sistem penghitung kendaraan menggunakan ROI dua arah.

4.2.5 Tampilan *Capture*



Gambar 4.12 *Output Capture File*

Pada Gambar 4.12 merupakan hasil dari *capture* objek yang berekstensi file .jpg dengan resolusi sesuai dengan *bounding box* saat melakukan pendeteksian dan terdapat 3 folder *capture* dimana masing-masing folder berisikan tentang hasil *capture* sesuai dengan arah objek yang dideteksi.

4.2.6 Tampilan Website



Gambar 4.13 Tampilan Website

Pada Gambar 4.13 merupakan tampilan *website* sistem penghitung jumlah kendaraan dua arah dengan *background subtraction*. Dimana tampilan awal *website* berisikan judul, *datepicker*, dan dashboard yang berisikan tentang *cardbox* mobil keluar, mobil masuk dan total mobil dimana nilai masing-masing *cardbox* ditentukan oleh range tanggal yang diinputkan.

4.3 Analisa Akurasi Sistem

Selanjutnya tahap Analisa Akurasi, tujuan dilakukan tahap ini adalah untuk dapat mengetahui seberapa tingkat akurasi sistem yang telah dibangun. Pada penelitian ini dilakukan percobaan dengan menggunakan *dataset* video berdurasi 1 menit dan diambil secara langsung menggunakan kamera *handphone* dan dengan bantuan *tripod* supaya gambar/video yang dihasilkan stabil.

4.3.1 Pengujian dan Analisa Pengaruh *Object Filter*

Dalam pengujian dan analisa pengaruh *object filter* bahan uji yang digunakan adalah area ROI kiri / lajur kendaraan keluar dan area ROI kanan / lajur kendaraan masuk, serta dua parameter uji yaitu *kernel* dan iterasi. Parameter uji pertama *kernel* dengan menggunakan *matriks kernel 3x3* dan *matriks kernel 5x5*, parameter uji kedua iterasi dengan menggunakan iterasi 1,2 dan 3. Nilai *threshold* yang digunakan pengujian ini adalah 50. Pengujian dan analisa pengaruh *object filter* bertujuan untuk mengetahui *matriks kernel* yang cocok agar dapat diimplementasikan oleh sistem dan iterasi yang sesuai dengan *dataset* yang sudah tersedia. Pada pengujian ini dilakukan perhitungan *error* dengan menggunakan pengujian *Mean Square Error* seperti pada Persamaan 4. Untuk nilai hasil evaluasi pengujian pengaruh *object filter* dapat dilihat pada Tabel 4.2 dan Tabel 4.3.

Tabel 4.1 Tabel Analisa Pengaruh *Object Filter* pada ROI Kiri

ROI Kiri / Mobil Keluar					
Percobaan ke	Kernel	Dilation	Jumlah yang dihasilkan sistem	Jumlah yang sebenarnya	Mean Square Error
1	3x3	1	41	30	20,16
2	3x3	2	31	30	0,16
3	3x3	3	34	30	2,66
4	5x5	1	34	30	2,66
5	5x5	2	33	30	1,5
6	5x5	3	30	30	0

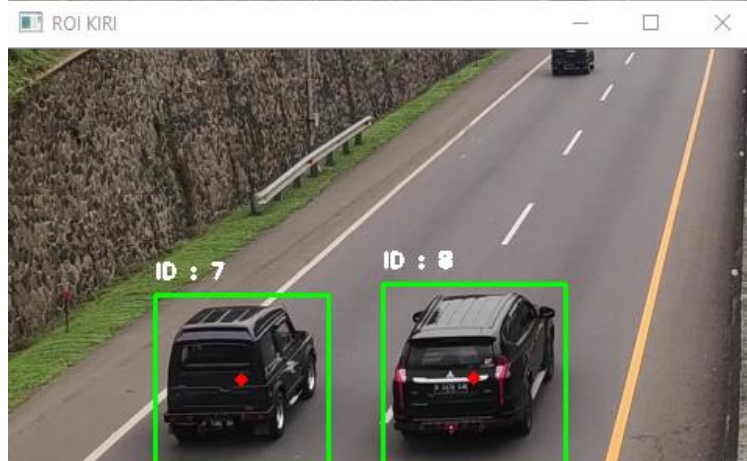
Pada Tabel 4.2 merupakan hasil analisa pengaruh *object filter* pada ROI Kiri mendapatkan nilai error terendah pada pengujian ke-6 dengan nilai MSE=0.

Tabel 4.2 Tabel Analisa Pengaruh *Object Filter* pada ROI Kanan

ROI Kanan / Mobil Masuk					
Percobaan ke	Kernel	Dilation	Jumlah yang dihasilkan sistem	Jumlah yang sebenarnya	Mean Square Error
1	3x3	1	43	29	32,66
2	3x3	2	31	29	0,66
3	3x3	3	35	29	6
4	5x5	1	33	29	2,66
5	5x5	2	29	29	0
6	5x5	3	28	29	-0,16

Pada Tabel 4.3 merupakan hasil analisa pengaruh *object filter* pada ROI Kanan mendapatkan nilai error terendah pada pengujian ke-6 dengan nilai MSE=-0,16.

Hasil pengujian pengaruh *object filter* ini diambil dari dua data uji yaitu ROI Kiri dan ROI Kanan, hasil *error* terendah 0 dan -0,16 dari kedua data uji yang digunakan seperti pada baris tabel berwarna hijau, kemudian hasil *error* terbesar



Gambar 4.14 *Output* Pendeteksian Benar

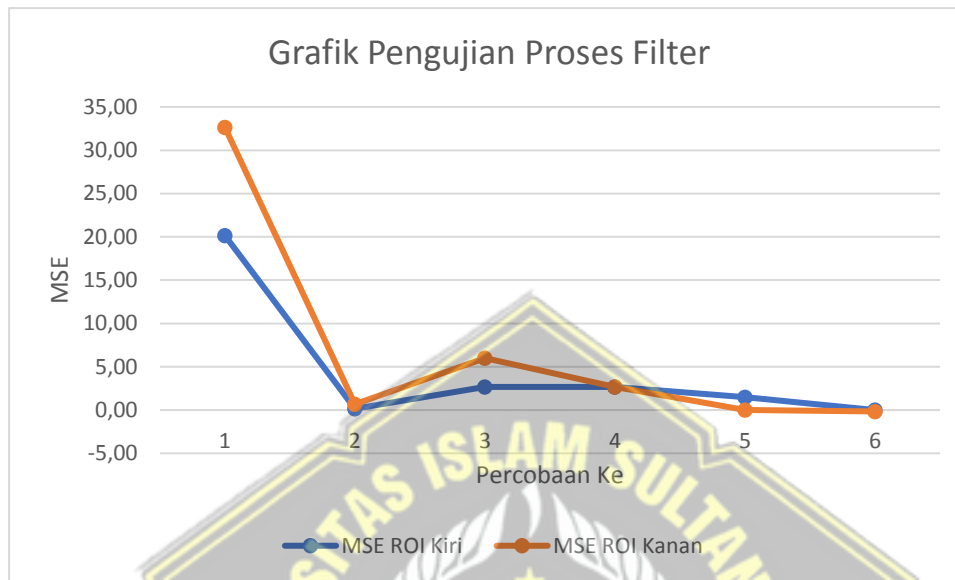
20,16 dan 32,66 dari kedua data uji yang digunakan seperti pada baris tabel berwarna merah, dengan nilai *threshold* adalah 50.

Pada Gambar 4.15 merupakan *output* pendeteksian benar dimana sistem dapat membaca object berdekatan dengan jumlah obek yang sebenarnya.



Gambar 4.15 *Output* Pendeteksian Salah

Pada Gambar 4.15 merupakan *output* pendeteksian salah dimana sistem dapat mampu membaca object berdekatan tetapi tidak sesuai dengan jumlah obek yang sebenarnya.



Gambar 4.16 Grafik Pengujian Proses *Filter*

Berdasarkan grafik pada Gambar 4.16 menjelaskan hasil pada pengujian dengan nilai kombinasi *error* terendah ketika percobaan ke 6 setiap data uji dilakukan yaitu kombinasi *matriks kernel* 5x5 dan *dilation* sebanyak 3 iterasi.

4.3.2 Pengujian dan Analisa pengaruh *Distance*

Pengujian dan analisa pengaruh *Distance* berguna untuk mengetahui pengaruh *distance* terhadap objek *tracking* yang terdeteksi sistem. Pada pengujian ini dilakukan evaluasi dengan jumlah *distance* 120, 130, 140, 150 dan 160 piksel. Nilai dari pengujian ini digunakan mendapatkan nilai evaluasi terkecil dan nilai evaluasi tertinggi. Dengan memanfaatkan parameter uji yang digunakan sebelumnya yang mendapatkan *error* terendah. Pengujian dan analisa pengaruh *distance* didapatkan nilai evaluasi *error* dan akurasi yang berbeda-beda. Untuk nilai hasil evaluasi pengujian pengaruh *distance* dapat dilihat pada Tabel 4.4 dan Tabel 4.5.

Tabel 4.3 Tabel Analisa *Distance* pada ROI Kiri

Pengujian dan Analisa <i>Distance</i> Terhadap ROI Kiri				
Percobaan ke	Setting <i>Distance</i>	Jumlah yang dihasilkan sistem	Jumlah yang sebenarnya	MSE
1	120piksel	30	30	0
2	130piksel	30	30	0
3	140piksel	32	30	0,8
4	150piksel	30	30	0
5	160piksel	30	30	0

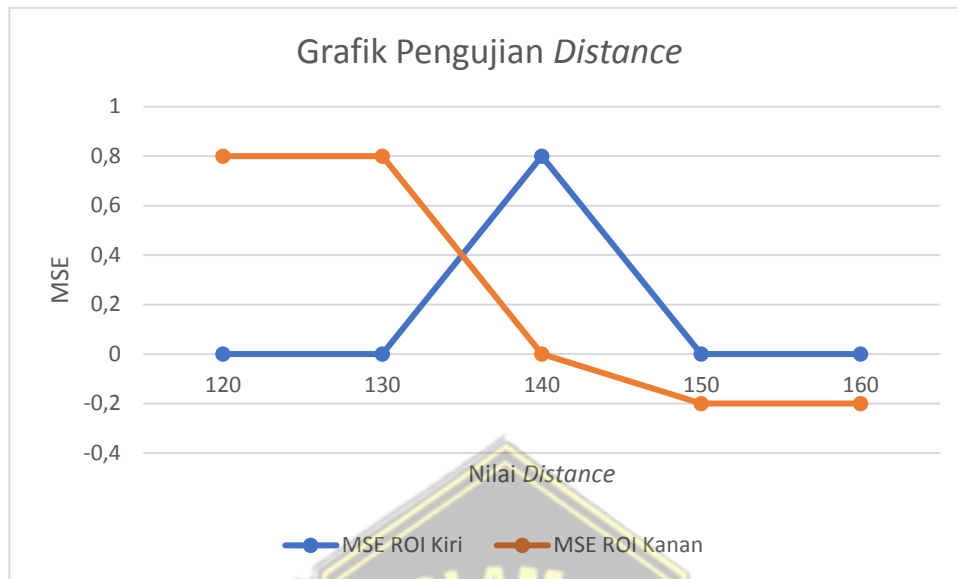
Pada Tabel 4.4 merupakan hasil Analisa *distance* pada ROI Kiri mendapatkan nilai error terendah pada pengujian ke 4 dengan nilai MSE=0.

Tabel 4.4 Tabel Analisa *Distance* pada ROI Kanan

Pengujian dan Analisa <i>Distance</i> Terhadap ROI Kanan				
Percobaan ke	Setting <i>Distance</i>	Jumlah yang dihasilkan sistem	Jumlah yang sebenarnya	MSE
1	120piksel	31	29	0,8
2	130piksel	31	29	0,8
3	140piksel	29	29	0
4	150piksel	28	29	-0,2
5	160piksel	28	29	-0,2

Pada Tabel 4.5 merupakan hasil Analisa *distance* pada ROI Kanan mendapatkan nilai error terendah pada pengujian ke 4 dengan nilai MSE=-0,2.

Hasil pengujian *distance* ini diambil dari dua data uji yaitu ROI Kiri dan ROI Kanan, hasil dari *error* terendah 0 dan -0,2 dari kedua data uji yang digunakan seperti pada baris tabel berwarna hijau dan hasil *error* terbesar 0,8 dari kedua data uji yang digunakan seperti pada baris tabel berwarna merah.



Gambar 4.17 Grafik Pengujian *Distance*

Berdasarkan grafik pada Gambar 4.17 menjelaskan hasil pada pengujian ini mendapatkan nilai *error* terendah dari kedua data uji yang digunakan adalah 150 piksel.

4.3.3 Analisa *Precision*

Perhitungan *precision* sistem penghitung jumlah kendaraan dua arah menggunakan metode *background subtraction*:

a. *Precision* ROI Kiri

$$Precision \text{ ROI Kiri} = \frac{30}{30+0} \times 100\%$$

$$Precision \text{ ROI Kiri} = 100\%$$

Dari hasil perhitungan *Precision* ROI Kiri menggunakan *dataset* yang ada didapatkan hasil 100%.

b. *Precision* ROI Kanan

$$Precision \text{ ROI Kanan} = \frac{23}{23+5} \times 100\%$$

$$Precision \text{ ROI Kanan} = 82,14\%$$

Dari hasil perhitungan *Precision* ROI Kanan menggunakan *dataset* yang ada didapatkan hasil 82,14%.

4.3.4 Analisa *Recall*

Perhitungan *recall* sistem penghitung jumlah kendaraan dua arah menggunakan metode *background subtraction*:

- a. *Recall* ROI Kiri

$$\text{Recall ROI Kiri} = \frac{30}{30+0} \times 100\%$$

$$\text{Recall ROI Kiri} = 100\%$$

Dari hasil perhitungan *Recall* ROI Kiri menggunakan *dataset* yang ada didapatkan hasil 100%.

- b. *Recall* ROI Kanan

$$\text{Recall ROI Kanan} = \frac{23}{23+8} \times 100\%$$

$$\text{Recall ROI Kanan} = 74,19\%$$

Dari hasil perhitungan *Recall* ROI Kiri menggunakan *dataset* yang ada didapatkan hasil 74,19%.

4.3.5 Analisa Akurasi

Perhitungan Akurasi sistem penghitung jumlah kendaraan dua arah menggunakan metode *background subtraction*:

- a. Akurasi ROI Kiri

$$\text{Akurasi ROI Kiri} = \frac{30+0}{30+0+0+0} \times 100\%$$

$$\text{Akurasi ROI Kiri} = 100\%$$

Dari hasil perhitungan Akurasi ROI Kiri menggunakan *dataset* yang ada didapatkan hasil 100%.

- b. Akurasi ROI Kanan

$$\text{Akurasi ROI Kanan} = \frac{23+6}{23+6+8+5} \times 100\%$$

$$\text{Akurasi ROI Kanan} = 69,05\%$$

Dari hasil perhitungan *Recall* ROI Kiri menggunakan *dataset* yang ada didapatkan hasil 69,05%.

4.3.6 Evaluasi Kinerja Sistem

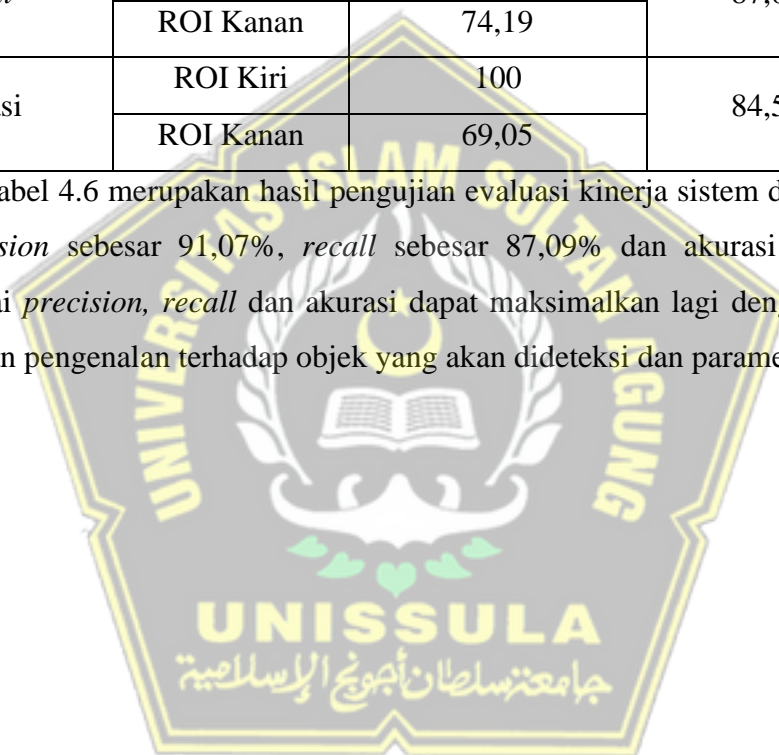
Pengujian evaluasi kinerja sistem digunakan untuk mengevaluasi kinerja *precision*, *recall* dan akurasi pada sistem penghitung jumlah kendaraan dua arah menggunakan *Background Subtraction* yang dihitung berdasarkan empat nilai yaitu *True Positive (TP)*, *False Positive (FP)*, *True Negative (TP)* dan *False Negative (FN)*.

Dari hasil pengujian dengan dataset video berdurasi 1 menit diperoleh tingkat probabilitas seperti pada Tabel 4.6.

Tabel 4.5 Tabel Evaluasi Kinerja Sistem

Jenis Kinerja	ROI	Presentase (%)	Presentase Total (%)
<i>Precision</i>	ROI Kiri	100	91,07
	ROI Kanan	82,14	
<i>Recall</i>	ROI Kiri	100	87,09
	ROI Kanan	74,19	
Akurasi	ROI Kiri	100	84,52
	ROI Kanan	69,05	

Pada Tabel 4.6 merupakan hasil pengujian evaluasi kinerja sistem diketahui bahwa *precision* sebesar 91,07%, *recall* sebesar 87,09% dan akurasi sebesar 84,52%. Nilai *precision*, *recall* dan akurasi dapat maksimalkan lagi dengan cara menambahkan pengenalan terhadap objek yang akan dideteksi dan parameter uji.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan perancangan sistem penghitung jumlah kendaraan dua arah dengan metode *background subtraction* di jalan tol Jatingaleh Semarang, penulis dapat menarik kesimpulan sebagai berikut:

1. Pendeteksi dan perhitungan jumlah kendaraan dapat dilakukan menggunakan metode algoritma *Background Subtraction* dengan memanfaatkan objek bergerak yang terdapat pada citra dengan beberapa tahapan untuk penyempurnaan sebuah citra dengan hasil yang maksimal. Kinerja sistem dengan menggunakan metode *Background Subtraction* pada kondisi lengang lebih baik daripada kondisi padat atau ramai sebesar 84,52%. Kesalahan pada saat kondisi padat banyak disebabkan karena posisi kendaraan yang berdekatan, sehingga sistem membaca banyak kendaraan sebagai satu objek kendaraan dengan dimensi besar.
2. Sistem dapat mengirimkan data berupa id, arah kendaraan dan tanggal ke *database MySQL* serta dapat memonitoring jumlah data kendaraan yang melewati jalan tol di Jatingaleh Semarang dengan statistik jumlah kendaraan yang dapat difilter menggunakan tanggal berbasis *webapp*.

5.2 Saran

Dalam implementasi metode *Background Subtraction* pada sistem penghitung jumlah kendaraan, prototipe ini masih banyak memiliki kekurangan. Karena keterbatasan penulis dalam perancangan sistem. Sehingga ditemukan beberapa saran yang bermanfaat untuk mengembangkan prototipe ini agar lebih baik.

1. Untuk penelitian selanjutnya ditambahkan proses pendekatan atau *data training* terhadap objek yang akan dideteksi supaya sistem mampu

mengetahui objek apa saja yang akan dideteksi dan menambahkan parameter supaya mendapatkan akurasi yang maksimal.

2. Sistem dapat diimplementasikan secara *real-time* dengan menambahkan *webcam* pada titik ruas jalan tol tertentu serta statistik jumlah kendaraan berbasis *webapp real-time*.



DAFTAR PUSTAKA

Adinugroho, S. dan Sari, Y.A. (2018) *Implementasi Data Mining Menggunakan Weka*. Universitas Brawijaya Press.

Ardhianto, E. *dkk.* (2013) “Implementasi Metode Image Subtracting dan Metode Regionprops untuk Mendeteksi Jumlah Objek Berwarna RGB pada File Video,” *Jurnal Teknologi Informasi DINAMIK*, 18(2), hal. 91–100.

Fajar Mit Cahyana *dkk.* (2014) “Perancangan Program Penghitung Jumlah Kendaraan Satu Arah Menggunakan Bahasa Pemrograman C++ dengan Pustaka OpenCV,” *Universitas Brawijaya* [Preprint].

Harwendhani, I.C. (2016) “Sistem Pendeteksi Jumlah Mobil Dalam Intelligent Transport System (ITS) Menggunakan Metode Viola-Jones,” *semantik*, 2.

Kurniawan, W.R. (2015) *Purwarupa sistem klasifikasi dan penghitung jumlah kendaraan bermotor menggunakan kamera webcam berbasis citra digital*. Tersedia pada: http://etd.repository.ugm.ac.id/home/detail_pencarian/86084.

M. Piccardi (2004) “Background subtraction techniques: a review,” 3099-3104, 4.

Menegaz, G. (2017) “Morphological Image Processing,” *Verona University* [Preprint].

Pratomo, A.H. *dkk.* (2020) “Implementasi Algoritma Region of Interest (ROI) Untuk Meningkatkan Performa Algoritma Deteksi Dan Klasifikasi Kendaraan,” *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 7(1), hal. 155–162. Tersedia pada: <https://doi.org/10.25126/jtiik.202071718>.

Purwanto, P. *dkk.* (2015) “Implementasi Face Identification Dan Face Recognition Pada Kamera Pengawas Sebagai Pendeteksi Bahaya,” *eProceedings of Engineering 2.1* [Preprint].

Sayidul, S. *dkk.* (2019) “Penentuan Jumlah Kendaraan Menggunakan Blob Detection dan Background Subtraction,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIC) Universitas Brawijaya*, 3(1), hal. 1029–1037.

Sutoyo, T.D. (2009) *Teori Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi.

Wang, Z. dan Bovik, A.C. (2009) “Mean squared error: Lot it or leave it? A new look at signal fidelity measures,” *IEEE Signal Processing Magazine*, 26(1), hal. 98–117. Tersedia pada: <https://doi.org/10.1109/MSP.2008.930649>.