



PROYEK AKHIR

SISTEM REAL TIME PENGKLASIFIKASI JENIS KENDARAAN DAN PENDETEKSI PELANGGARAN LAMPU LALU LINTAS BERBASIS INTERNET OF THINGS

Achmad Rahman Mawardi
NRP. 2210141045

Dosen Pembimbing 1:
Mochamad Mobed Bachtiar, S.ST., M.T
NIP.198802172015041002

Dosen Pembimbing 2:
Adnan Rachmat Anom Besari, S.ST., M.Sc.
NIP. 198509102012121003

PROGRAM STUDI D4 TEKNIK KOMPUTER
DEPARTEMEN TEKNIK INFORMATIKA
DAN KOMPUTER
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2018



PROYEK AKHIR

***SISTEM REAL TIME PENGKLASIFIKASI
JENIS KENDARAAN DAN PENDETEKSI PELANGGARAN
LAMPU LALU LINTAS BERBASIS INTERNET OF THINGS***

Achmad Rahman Mawardi
NRP. 2210141045

Dosen Pembimbing 1:
Mochamad Mobed Bachtiar, S.ST., M.T
NIP.198802172015041002

Dosen Pembimbing 2:
Adnan Rachmat Anom Besari, S.ST., M.Sc.
NIP. 198509102012121003

**PROGRAM STUDI D4 TEKNIK KOMPUTER
DEPARTEMEN TEKNIK INFORMATIKA
DAN KOMPUTER
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2018**

***SISTEM REAL TIME PENGKLASIFIKASI
JENIS KENDARAAN DAN PENDETEKSI PELANGGARAN
LAMPU LALU LINTAS BERBASIS INTERNET OF THINGS***

Oleh :

Achmad Rahman Mawardi
NRP. 2210141045

Proyek Akhir ini Digunakan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Sains Terapan (S.ST)

di

Politeknik Elektronika Negeri Surabaya
2018

Disetujui Oleh :

Tim Penguji Proyek Akhir : Dosen Pembimbing :

1. Mochamad Mobed Bachtiar , S.ST., M.T
NIP.198802172015041002

2. Adnan Rachmat Anom Besari, S.ST., M.Sc.
NIP. 198509102012121003

Mengetahui

Ketua Program Studi D4 Teknik Komputer
Departemen Teknik Informatika dan Teknik Komputer
Politeknik Elektronika Negeri Surabaya

Sigit Riyanto, ST., M.Kom., Ph.D
NIP. 197008111995121001

LEMBAR PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan dibawah ini dengan sebenarnya menyatakan bahwa Proyek Akhir ini saya susun tanpa tindakan plagiarisme sesuai dengan peraturan yang berlaku di Politeknik Elektronika Negeri Surabaya (PENS)

Nama : Achmad Rahman Mawardi
NRP : 2210141045
Program Studi : Teknik Komputer
Departemen : Teknik Informatika dan Komputer

Jika di kemudian hari saya terbukti melakukan tindakan plagiarisme, saya akan bertanggung jawab sepenuhnya dan menerima sanksi yang dijatuhkan oleh PENS kepada saya.

Surabaya, 21 Januari 2018

Achmad Rahman Mawardi
NRP. 2210141045

KATA PENGANTAR

Segala puji syukur kepada Tuhan Yang Maha Esa, karena atas segala rahmat dan penyertaan-Nya, penulis dapat menyelesaikan proyek akhir yang berjudul:

SISTEM REAL TIME PENGKLASIFIKASI JENIS KENDARAAN DAN PENDETEKSI PELANGGARAN LAMPU LALU LINTAS BERBASIS INTERNET OF THINGS

Buku Proyek Akhir ini dimaksudkan untuk memenuhi persyaratan guna menyelesaikan studi Diploma IV di Politeknik Elektronika Negeri Surabaya (PENS).

Terdapat beberapa literatur dan teori baik yang diperoleh dalam perkuliahan maupun dari luar perkuliahan yang digunakan dalam penyelesaian proyek akhir ini, dan juga tidak lepas dari dukungan dosen pembimbing serta pihak-pihak lain yang telah banyak memberikan semangat dan bantuan.

Penulis menyadari bahwa buku proyek akhir ini masih memiliki banyak kekurangan. Untuk itu penulis memohon maaf sebesar-besarnya atas kekurangan yang ada pada buku proyek akhir ini. Selain itu penulis mengharapkan saran dan kritik dari semua pihak demi kesempurnaan buku ini.

Besar harapan penulis agar buku proyek akhir ini dapat bermanfaat khususnya bagi penulis dan pembaca pada umumnya serta untuk menambah wawasan dan pengetahuan.

Surabaya, 21 Januari 2018
Penulis

ABSTRAK

Kurangnya kesadaran pengguna kendaraan akan ketaatan pada peraturan lalu lintas sering menjadi pemicu terjadinya pelanggaran, khususnya pada area lampu lalu lintas pelanggaran terhadap marka sering dilakukan oleh pengguna kendaraan. Oleh karena itu pada tugas akhir akan dibuat, sistem yang dapat mendeteksi dan klasifikasi kendaraan serta pelanggaran yang terjadi. Data yang diperoleh akan dikirim dan diproses melalui internet guna memberikan informasi kepada pihak berwenang untuk proses lebih lanjut. Proses-proses dari sistem yang dilakukan yaitu deteksi kendaraan, klasifikasi kendaraan, dan *violation detection* kemudian pengiriman data melalui internet. Untuk mendeteksi kendaraan metode yang digunakan adalah *frame difference* dari background, kemudian mendeteksi objek bergerak dan melakukan perbaikan gambar lalu menentukan apakah termasuk kendaraan, jika iya maka objek tersebut akan mengalami proses klasifikasi. Klasifikasi sendiri menggunakan perhitungan luas area objek. Sedangkan untuk mendeteksi pelanggaran digunakan proses *violation detection* yang berisi mengenai beberapa keadaan untuk menentukan jenis pelanggaran. Untuk perhitungan dan klasifikasi kendaraan dengan metode ini dapat akurasi hingga 93% untuk *video* yang bagus tanpa *noise*, dan akurasi 78% pada *noisy video*.

Kata kunci: kendaraan, deteksi, pelanggaran, internet

ABSTRACT

Lack of vehicle *user* awareness of obedience to traffic rules is often a trigger for violations, particularly in areas where traffic lights violate the marks are often done by vehicle *users*. Therefore in this final project will be made, the system that can detect, classification of vehicles and violations that occurred. The data obtained will be sent and processed through the internet to provide information to the authorities for further process. The step of the system are vehicle detection, vehicle classification, and *violation detection* and then data transmission over the internet. To detect the vehicle the method used is the frame difference of the background, then detect the moving objek and do *image* repairs then determine whether the included vehicle, if yes then the objek will undergo the process of classification. The classification itself uses the calculation of the area of the objek. Meanwhile, to detect violations used *violation detection* process that contains about several circumstances to determine the type of violation. For the calculation and classification of vehicles with this method can be up to 93% accuracy for good *video* without *noise*, and 78% accuracy on noisy *video*.

Keywords: *vehicle, detection, violation, internet*

DAFTAR ISI

DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan dan Manfaat.....	3
1.5 Metodologi Penelitian	3
BAB II TINJAUAN PUSTAKA	5
2.1 Lalu Lintas	5
2.2 Marka Jalan	9
2.3 <i>Video Surveillance</i>	11
2.4 Deteksi Kendaraan	13
2.5 Klasifikasi Kendaraan.....	16
2.6 <i>Violation Detection</i>	16
2.7 Internet of Things	19
2.8 Referensi Penelitian.....	20
BAB III PERANCANGAN DAN IMPLEMENTASI SISTEM	23
3.1 Perancangan Penelitian.....	23
3.2 Perancangan Deteksi Kendaraan	24
3.3 <i>Switch Mode</i>	31
3.4 Deteksi Pelanggaran saat Lampu Hijau	32
3.5 Deteksi Pelanggaran saat Lampu Merah	34
3.6 Setting Dropbox	36
3.7 Setting Faceboook	36
3.8 Setting Zapier	37
3.9 Graphical User Interface.....	41
BAB IV PENGUJIAN DAN ANALISA	43
4.1 Pengujian Blob (Deteksi kedaraan)	44
4.2 Pengujian Klasifikasi Kendaraan.....	47
4.3 Pengujian Deteksi Lampu.....	48
4.4 Pengujian Pelanggaran Lampu	48
4.5 Pengujian Pelanggaran Marka	49
4.6 Pengujian <i>Realtime</i>	50
4.7 Pengujian Report	51
4.8 Jenis <i>Error</i>	52

BAB V PENUTUP.....	55
12.1 Kesimpulan.....	55
12.2 Hal yang Dilakukan Selanjutnya	55
DAFTAR PUSTAKA	57
LAMPIRAN	59

DAFTAR GAMBAR

Gambar 1.1, Metodologi penelitian	3
Gambar 2.1, Kondisi lalu lintasa	5
Gambar 2.2, Kondisi pelanggaran di area lampu lalu lintas	7
Gambar 2.3, Titik konfilk persimpangan	8
Gambar 2.4, Marka garis membujur	9
Gambar 2.5, Marka garis melintang	10
Gambar 2.6, Marka serong	10
Gambar 2.7, Marka lambang	11
Gambar 2.8, Cara kerja <i>video surveillance</i>	12
Gambar 2.9, Operasi thresholding	14
Gambar 2.10, Operasi Convexhull	15
Gambar 2.11, Metode klasifikasi berdasar ukuran kendaraan	16
Gambar 2.12, ROI-Garis pelanggar lampu merah	17
Gambar 2.13, ROI-Garis pelanggar lampu hijau	18
Gambar 2.14, Internet report dan penyimpanan data	19
Gambar 2.15, Hasil pengujian deteksi pelanggaran lampu hijau	21
Gambar 2.16, Hasil pengujian deteksi pelanggaran lampu merah	21
Gambar 3.1, Diagram sistem alur yang akan dibuat	23
Gambar 3.2, Diagram alur proses perancangan deteksi kendaraan	24
Gambar 3.3, Hasil image subtraction	25
Gambar 3.4, Hasil threshold image	26
Gambar 3.5, Hasil contour image	26
Gambar 3.6, Hasil convexhull pada contour	27
Gambar 3.7, Hasil blob pada convexhull	28
Gambar 3.8, Pergantian merah ke hijau & hijau ke merah	31
Gambar 3.9, Diagram alur sistem pendeteksi pelanggaran lampu hijau ..	32
Gambar 3.10, Proses preprocessing pada deteksi pelanggaran hijau	32
Gambar 3.11, Proses segmentasi pada deteksi pelanggaran hijau	33
Gambar 3.12, Proses <i>violation detection</i> pada deteksi pelanggaran hijau	33
Gambar 3.13, Diagram alur sistem deteksi pelanggaran lampu merah ...	34
Gambar 3.14, Proses preprocessing pada deteksi pelanggaran merah	34
Gambar 3.15, Proses segmentasi pada deteksi pelanggaran merah	35
Gambar 3.16, Proses vilation detection pada deteksi pelanggaran merah	35
Gambar 3.17, Laman installasi Dropbox	36
Gambar 3.18, Laman pembuatan Facebook pages	37
Gambar 3.19, Laman home Zapier	37
Gambar 3.20, Proses pengaitan Dropbox-Facebook step-1	38
Gambar 3.21, Proses pengaitan Dropbox-Facebook step-2	38

Gambar 3.22, Proses pengaitan Dropbox-Facebook step-3	38
Gambar 3.23, Proses pengaitan Dropbox-Facebook step-4	39
Gambar 3.24, Proses pengaitan Dropbox-Facebook step-5	39
Gambar 3.25, Proses pengaitan Dropbox-Facebook step-6	40
Gambar 3.27, Proses pengaitan Dropbox-Facebook step-4	40
Gambar 3.28, GUI platform desktop	41
Gambar 4.1, Posisi hadap kamera CCTV yang digunakan	43
Gambar 4.2, Input <i>video</i> yang digunakan dari berbagai waktu	44
Gambar 4.3, Penggunaan garis counter pada saat lampu hijau	47
Gambar 4.4, Hasil pengujian deteksi lampu	48
Gambar 4.5, Penggunaan garis counter pada saat lampu merah	48
Gambar 4.6, Penggunaan garis counter pada saat lampu hijau	49
Gambar 4.7, Proses pengujian <i>realtime</i>	50
Gambar 4.8, Hasil pengujian report	51
Gambar 4.9, Jenis <i>Error</i> : Kamera bergetar	52
Gambar 4.10, Jenis <i>Error</i> : Pantulan cahaya berlebih	52
Gambar 4.11, Jenis <i>Error</i> : Penyebrang jalan	53
Gambar 4.12, Jenis <i>Error</i> : Double check	53

DAFTAR TABEL

Tabel 4.1	Image substract	44
Tabel 4.2	Contour image	45
Tabel 4.3	Blob image.....	45
Tabel 4.4	Hasil pengujian jumlah kendaraan	46
Tabel 4.5	Hasil pengujian jenis kendaraan	47
Tabel 4.6	Hasil pengujian pelanggaran lampu.....	49
Tabel 4.7	Hasil pengujian pelanggaran marka.....	50
Tabel 4.8	Hasil pengujian <i>realtime</i>	51

BAB I

PENDAHULUAN

1.1 Latar Belakang

Setiap tahun pengguna kendaraan bermotor selalu bertambah dan kemacetan terjadi di hampir setiap jalan di seluruh wilayah Surabaya. Maka tak jarang kita melihat banyak kendaraan-kendaraan yang terjebak macet hampir di sepanjang jalan, khususnya di lampu lalu lintas. Namun, kurangnya kesadaran akan ketaatan terhadap peraturan lalu lintas serta keegoisan para pengendara kendaraan kerap kali menjadi pemicu terhadap terjadinya pelanggaran-pelanggaran yang terjadi. Pihak berwenang telah berusaha melakukan terobosan dalam hal menertibkan para pengguna jalan tetapi hal ini masih dirasa kurang. Kondisi ini merupakan tanggung jawab semua pengguna jalan bukan hanya tanggung jawab pihak kepolisian.

Secara kuantitatif, perkara pelanggaran lalu lintas di pengadilan sangat mendominasi jumlah perkara yang harus diperiksa, diselesaikan dan diputuskan oleh Hakim. Pada tahun 2013 tercatat dalam Laporan Tahunan Mahkamah Agung 2013, perkara pelanggaran lalu lintas merupakan jenis perkara terbesar. Total jumlah perkara pidana yang ditangani pengadilan negeri pada 2013 berjumlah 3.386.149 perkara. Sebanyak 3.214.119 atau 96,40% merupakan perkara tindak pidana ringan dan pelanggaran lalu lintas. Perkara pidana biasa pada 2013, sebesar 119.876 atau 3,60%. Sisanya merupakan perkara pidana singkat sebesar 231 perkara atau 0,01%[1]. Bahkan dalam setiap minggu pada pengadilan di kota besar dapat menerima 2000 (dua ribu) sampai dengan 4000 (empat ribu) perkara[2].

Pelanggaran kecil yang dinyatakan pada undang-undang No.22/2009 pasal 287 (1) jo. Pasal 106 (4) huruf b: “Melanggar aturan perintah atau larangan yang dinyatakan dengan rambu lalu lintas atau marka” yang mencakup pelanggaran di area lampu lalu lintas yang menjadi perhatian pada penelitian ini. Khususnya ketika pengendara berhenti pada *zebra cross* atau melewatinya ketika lampu merah dan ketika salah mengambil lajur ketika lampu hijau.

Pelanggaran-pelanggaran di jalan yang sering dianggap remeh masyarakat dapat berakibat fatal. Menurut WHO, kecelakaan lalu lintas telah menelan 1.3 juta jiwa hanya pada tahun 2015. Jumlah angka kematian yang diakibatkan kecelakaan lalu lintas ini menduduki peringkat 2 penyebab kematian manusia setelah penyakit kronis seperti HIV/AIDS dan TBC[3].

Dari uraian yang dijabarkan, pada proyek akhir ini akan dibuat sistem pengawasan yang dapat mengklasifikasi kendaraan serta mendeteksi pelanggaran lalu lintas serta menyimpan bukti pelanggarannya kemudian melakukan pengiriman data melalui internet guna proses optimalisasi dan pengambilan saran yang cocok dengan kondisi saat ini pada area lampu lalu lintas.

1.2 Rumusan Masalah

Berdasarkan uraian diatas, maka permasalahan dapat dirumuskan sebagai berikut:

1. Bagaimana mengatur agar mendapat sudut pandang kamera dapat dijadikan objek pengamatan?
2. Bagaimana cara melakukan *tracking* kendaraan pada *video*?
3. Bagaimana cara mengklasifikasi kendaraan menjadi mobil atau motor?
4. Bagaimana cara mendeteksi jumlah pelanggar disaat lampu mengisyaratkan berhenti/berjalan?
5. Bagaimana data yang didapat dikirim melalui internet?

1.3 Batasan Masalah

Berdasarkan rumusan masalah diatas, maka untuk proyek akhir ini diberikan batasan masalah sebagai berikut:

1. Kondisi jalan yang dipantau memiliki 5 lajur, dengan kondisi sebagai berikut:
 - a. Belok kiri langsung
 - b. Lurus mengikuti isyarat lampu
 - c. Lurus mengikuti isyarat lampu
 - d. Lurus atau belok kanan mengikuti isyarat lampu
 - e. Belok kanan mengikuti isyarat lampu.
2. Kamera yang digunakan harus dapat mengambil gambar kegaris tepi antara batas lampu lalu lintas dan *zebra cross*.
3. Terdapat 2 mode yang digunakan pada *violation detection* yaitu mode merah dan mode hijau.
4. Setiap kendaraan harus melewati suatu ROI/garis untuk dapat melakukan proses klasifikasi.
5. Kondisi jalan yang sangat ramai dan posisi kendaraan saling bertumpuk akan mengurangi akurasi dari program.

6. Jenis pelanggaran yang dapat dideteksi oleh sistem adalah jenis pelanggaran sebagai berikut:
 - a. Pelanggaran Lampu - Posisi tengah kendaraan berada di garis pelanggaran lampu atau melewatinya ketika mode merah.
 - b. Pelanggaran Marka - Keseluruhan bagian kendaraan telah melewati garis pelanggaran marka ketika mode hijau.

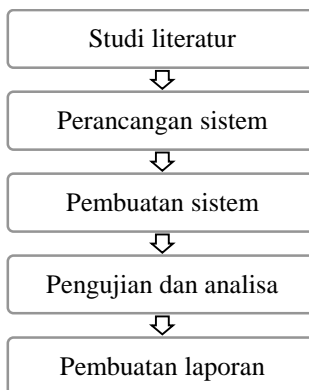
1.4 Tujuan dan Manfaat

Tujuan dari proyek akhir ini adalah membangun sistem pengawasan, dapat mendeteksi pelanggaran, dapat mendeteksi kendaraan dan melakukan laporan secara *realtime* pada lingkup area lampu lalu lintas.

Manfaat dari proyek ini adalah agar dapat melakukan pengawasan pada area lampu lalu lintas dengan memanfaatkan teknik *image processing* dan *computer vision* sehingga program dapat mendeteksi pelanggaran. Selain itu terdapat bukti yang berisi jenis kendaraan dan pelanggaran yang dilakukan dalam gambar, serta data pelanggar akan dimasukkan melalui internet untuk proses pembuatan laporan secara spesifik untuk proses lebih lanjut pihak berwenang.

1.5 Metodologi Penelitian

Sistem penelitian yang dikerjakan melalui beberapa tahap, Gambar 1.1 merupakan metodologi yang diterapkan dalam pengerjaan proyek akhir:



Gambar 1.1, Metodologi penelitian

1.5.1 Studi Literatur

Tahap meliputi studi pustaka dan literatur tentang keseluruhan sistem penelitian yang terdiri dari studi tentang *image processing* dan *computer vision* serta teori dan data lain yang mendukung penelitian.

1.5.2 Perencanaan Sistem

Tahap perencanaan sistem yang dimulai dari *preprocessing video*, pengklasifikasian kendaraan dan pembuatan *violation detection* yang akan digunakan pada *platform desktop* kemudian pengiriman semua data statistik.

1.5.3 Pembuatan Sistem

Tahap pembuatan sistem terdiri dari: proses *segmentasi* gambar, perbaikan gambar, klasifikasi kendaraan, *tracking* kendaraan dan terakhir penentuan *violation rule* sehingga dapat dideteksi. Setelah semua proses pada *video* selesai data tersebut akan dikirim ke internet.

1.5.4 Pengujian dan Analisa

Tahap dilakukan dengan pengujian pada satu tempat dengan jumlah kendaraan dan waktu yang berbeda. *Video* yang dimasukan tadi akan dimasukan kedalam proses *image subtraction*, *threshold*, *contour*, Convex hull, blob, klasifikasi kendaraan, *tracking* kendaraan dan *violation detection*. Dari setiap proses dapat diamati perubahannya sehingga dapat dilakukan klasifikasi kendaraan dan pengamatan pada pelanggaran.

1.5.5 Pembuatan Laporan

Tahap yang dilakukan adalah pembuatan laporan berdasarkan penelitian yang telah dilakukan, hasil pengujian sistem, serta analisa yang didapat.

BAB II

TINJAUAN PUSTAKA

2.1 Lalu Lintas

Lalu lintas dalam undang undang No. 22 tahun 2009 dedefinisikan sebagai gerak kendaraan dan orang di ruang lalu lintas jalan, sedang yang dimaksud dengan ruang lalu lintas jalan adalah prasarana yang diperuntukan bagi gerak pindah kendaraan, orang dan/atau barang yang berupa jalan dan fasilitas pendukung. Pemerintah mempunyai tujuan untuk mewujudkan lalu lintas dan angkutan jalan yang tertib, aman, cepat, lancar, nyaman, efisien, dan teratur melalui manajemen lalu lintas dan rekayasa lalu lintas[3].

Tata cara berlalu lintas di jalan diatur oleh undang-undang yang menyangkut arah lalu lintas, prioritas menggunakan jalan, jalur lalu lintas dan pengendalian arus di persimpangan.



Gambar 2.1, Kondisi lalu lintas

2.1.1 Komponen Lalu Lintas

Terdapat 3 komponen utama dalam terjadinya lalu lintas yaitu manusia sebagai pengguna, kendaraan dan jalan yang saling berinteraksi dalam pergerakan kendaraan yang memenuhi persyaratan kelayakan mengemudi mengikuti aturan lalu lintas yang ditetapkan berdasar kepada perundang-undangan yang menyangkut mengenai lalu lintas dan angkutan jalan.

Ketiga komponen utama dalam terjadinya lalu lintas adalah manusia, kendaraan dan jalan, berikut adalah penjelasannya:

1. Manusia

Manusia sebagai pengguna, manusia dapat berperan sebagai pengemudi atau pejalan kaki yang pada dasarnya memiliki kemampuan dan kesiagaan berbeda. Keadaan tersebut umumnya dipengaruhi keadaan fisik, psikologi, usia serta jenis kelamin serta pengaruh dari luar seperti cuaca, penerangan dan tata ruang

2. Kendaraan

Kendaraan digunakan oleh pengemudi mempunyai karakter yang berkaitan dengan kecepatan, percepatan, perlambatan, dimensi, muatan yang membutuhkan ruang lalu lintas secukupnya untuk dapat berperan dalam lalu lintas.

3. Jalan

Jalan merupakan lintasan yang dirancang untuk dilalui kendaraan bermotor ataupun tidak, termasuk juga pejalan kaki. Jalan direncanakan untuk mampu mengalirkan aliran lalu lintas dengan lancar dan mampu mendukung beban muatan kendaraan serta aman sehingga dapat meredam kecelakaan.

2.1.2 Menejemen lalu lintas

Manajemen lalu lintas meliputi kegiatan perencanaan, pengaturan, pengawasan dan pengendalian lalu lintas. Tujuan dari manajemen lalu lintas adalah menciptakan lalu lintas yang aman, tertib dan lancar. Tata cara manajemen lalu lintas dilakukan antara lain:

1. Usaha peningkatan kapasitas ruang jalan, persimpangan atau jaringan jalan.
2. Pemberian prioritas kepada pengguna jalan tertentu.
3. Penyesuaian antara permintaan perjalanan dengan tingkat pelayanan tertentu dengan mempertimbangkan keterpaduan intra dan antar moda.
4. Penetapan sirkulasi lalu lintas, larangan dan/atau anjuran pada pengguna jalan.

2.1.3 Pelanggaran Lalu Lintas

Perumusan mengenai pelanggaran lalu lintas tidak dapat ditemukan dalam buku ketiga KUHP sebab pelanggaran lalu lintas diatur suatu undang-undang tersendiri yaitu dalam Undang-undang No. 14 tahun 1992 tentang lalu lintas dan angkutan jalan. Lalu lintas dalam KBBI memiliki definisi yaitu bolak-balik, hilir-mudik perihal perjalanan di jalan,

perhubungan antar satu tempat ke tempat lainnya. Sedangkan dalam pasal 1 butir 1 UU No. 14 tahun 1992 didefinisikan sebagai lalu lintas gerak kendaraan, orang dan hewan di jalan.

Undang-undang No. 14 tahun 1992 mengatur juga mengenai semua aturan yang terkait di lalu lintas jalan. Tujuan dibentuknya undang-undang tersebut adalah:

1. Mempertinggi mutu kelancaran, keamanan dan ketertiban lalu lintas dan angkutan di jalan raya.
2. Mengatur dan menyalurkan secara tertib segala pengangkutan barang dan penumpang terutama dengan kendaraan umum.
3. Melindungi semua jalan dan jembatan agar jangan dihancurkan atau dirusak dan jangan sampai susut melewati batas diakrenakan beban muatan yang sangat berat.



Gambar 2.2, Kondisi pelanggaran di area lampu lalu lintas

Untuk mencapai tujuan tersebut dalam undang-undang No. 14 tahun 1992 dimuat berbagai aturan mengenai tindak pidana pelanggaran lalu lintas dan angkutan jalan. Peraturan mengenai ketentuan pidana lalu lintas dan angkutan jalan diatur dalam Bab XII dari pasal 54 sampai 67 UU No. 14 tahun 1992. Tindak pidana pelanggaran lalu lintas dan angkutan jalan yang diatur ada 2 jenis, yaitu:

1. Tindak pidana pelanggaran lalu lintas, yang terdiri dari:
 - a. Pelanggaran terhadap pemberi isyarat lalu lintas
 - b. Pelanggaran terhadap marka
 - c. Pelanggaran terhadap rambu-rambu lalu lintas
 - d. Pelanggaran terhadap kecepatan
 - e. Pelanggaran terhadap peringatan suara
 - f. Pelanggaran terhadap persyaratan administrative pengemudi kendaraan

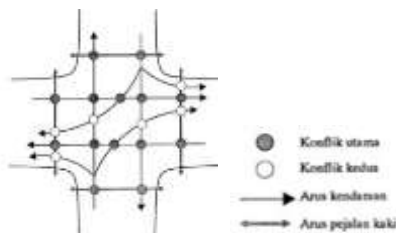
2. Tindak pidana angkutan jalan, yang terdiri dari:
 - a. Pelanggaran terhadap persyaratan teknis dan baik jalan kendaraan
 - b. Pelanggaran terhadap perizinan
 - c. Pelanggaran terhadap berat muatan kendaraan

Berdasarkan penjelasan diatas dapat disimpulkan bahwa pelanggaran lampu lalu lintas dalam perbuatan yang bertentangan dengan apa yang secara tegas dituliskan dalam undang-undang No. 14 tahun 1992 sebagai salah satu bentuk pelanggaran. Tetapi pelanggaran lalu lintas bukan hanya diatur undang-undang No. 14 tahun 1992 karena undang-undang tersebut bersifat umum dan berlaku secara nasional di Indonesia, sehingga dimungkinkan adanya UU yang bersifat khusus untuk daerah-daerah tertentu.

2.1.4 Karakteristik Lampu Lalu Lintas

Kondisi geometri dan lalu lintas akan berpengaruh pada kapasitas dan kinerja lalu lintas di persimpangan. Karena itu perencana harus merancang sedemikian rupa sehingga mampu mendistribusikan waktu kepada masing-masing kelompok pergerakan kendaraan secara proporsional dan mendapat hasil yang sebaik-baiknya.

Perlu diketahui dengan adanya peraturan lalu lintas baru (PP 42 dan PP 43) untuk kendaraan yang belok kiri selama tidak diatur secara khusus maka kendaraan boleh jalan terus. Pemberian isyarat dengan nyala lampu diterapkan untuk memisahkan pergerakan lalu lintas berdasarkan waktu. Pemisahan ini diperlukan khususnya untuk jenis konflik primer, yang dalam keadaan tertentu juga diterapkan dalam konflik sekunder.



Gambar 2.3, Titik konflik persimpangan

Konflik primer adalah pertemuan aliran kelompok pergerakan kendaraan dari persimpangan jalan (*crossing*). Konflik sekunder adalah pertemuan yang berasal dari aliran pergerakan kendaraan dari persilangan jalan, konflik sekunder dapat berupa pertemuan lalu lintas yang berlawanan arus dengan belok ke kiri (*opposing straight – through traffic*) dan pertemuan dengan pejalan kaki (*crossing pedestrian*).

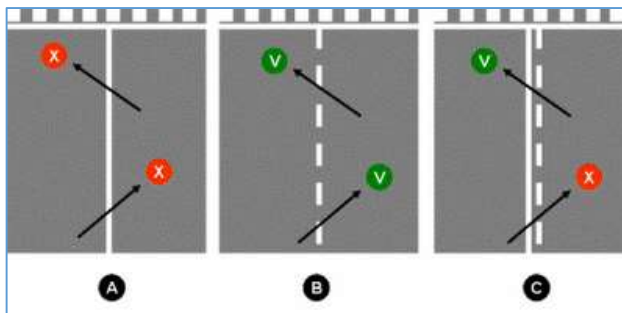
2.2 Marka Jalan

Marka jalan adalah suatu tanda yang berada di permukaan jalan atau di atas permukaan jalan yang meliputi peralatan atau tanda yang membentuk garis membujur, garis melintang, garis serong serta lambang lainnya yang berfungsi untuk mengarahkan arus lalu lintas dan membatasi daerah kepentingan lalu lintas. Berikut jenis-jenis marka jalan.

2.2.1 Marka Garis Membujur

Marka garis membujur, marka yang sejajar dengan sumbu jalan. Ada tiga jenis marka garis membujur, yaitu:

- Marka garis membujur penuh (tidak terputus). Pengemudi dilarang melintasi marka ini. Biasanya dipasang di tempat-tempat yang mengandung bahaya, misalnya tikungan, tanjakan, turunan, atau tempat yang ramai.
- Marka garis membujur putus-putus. Pengemudi boleh melintasi marka ini, misalnya untuk pindah jalur atau mendahului kendaraan lain.
- Marka garis membujur kombinasi penuh dan putus-putus. Pengemudi yang berada di sisi jalan yang lebih dekat dengan marka putus-putus, boleh melintasi marka kombinasi. Sebaliknya, pengemudi yang berada di sisi jalan yang lebih dekat dengan marka penuh, dilarang melintasi marka kombinasi.

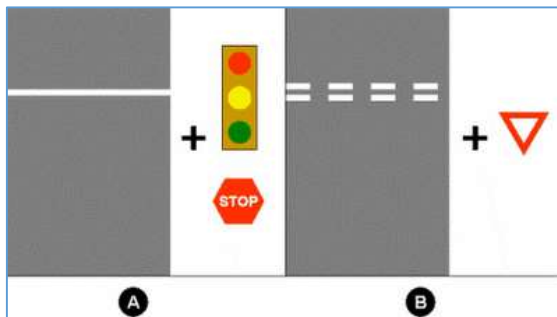


Gambar 2.4, Marka garis membujur

2.2.2 Marka Garis Melintang

Marka garis melintang, marka yang tegak lurus terhadap sumbu jalan. Marka ini digunakan untuk mengingatkan pengendara untuk berhenti atau mengurangi kecepatan. Marka ini juga berfungsi untuk menguatkan rambu dan traffic light. Marka garis melintang antara lain:

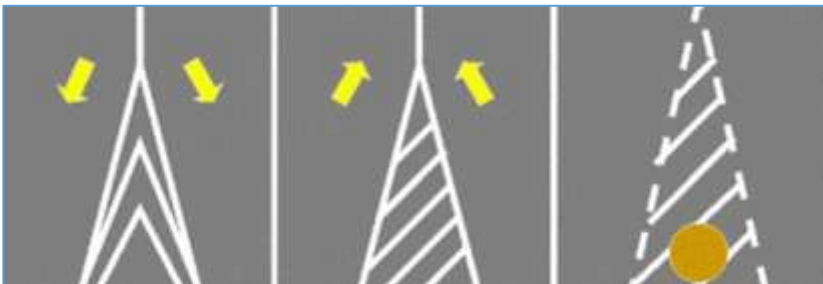
- Marka garis melintang utuh. Marka ini menguatkan rambu stop dan traffic light sebagai tanda berhenti kendaraan. Marka ini juga ada di perlintasan kereta api.
- Marka garis melintang putus-putus. Marka ini menguatkan rambu hati-hati sebagai tanda batas berhenti untuk memberikan kesempatan mendahului kendaraan lain yang telah ditetapkan oleh rambu.



Gambar 2.5, Marka garis melintang

2.2.3 Marka Serong

Marka serong, marka yang membentuk garis utuh yang tidak termasuk dalam pengertian marka membujur atau marka melintang. Gunanya untuk menyatakan suatu daerah di permukaan jalan yang bukan merupakan jalur lalu lintas kendaraan.



Gambar 2.6, Marka serong

2.2.4 Marka Lambang

Marka lambang, tanda yang mengandung arti tertentu untuk menyatakan peringatan, perintah dan larangan untuk melengkapi atau menegaskan maksud yang telah disampaikan oleh rambu lalu lintas atau tanda lalu lintas lainnya. Bentuknya berupa panah, segitiga, atau tulisan dipergunakan untuk mengulangi maksud rambu-rambu lalu lintas atau untuk memberitahu pemakai jalan yang tidak dinyatakan dengan rambu.



Gambar 2.7, Marka lambang

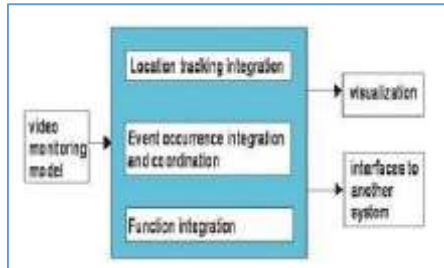
2.3 Video Surveillance

Video surveillance Sistem atau di sebut *Closed Circuit Television* Sistem berfungsi mengontrol semua kegiatan secara *visual (audio visual)* pada area tertentu yang dipasang suatu alat berupa kamera. Fungsinya secara langsung dapat mengawasi, dan mengamati serta merekam kejadian di suatu tempat, ruangan atau area tertentu, alat ini terdiri dari: kamera, *digital video recorder*, *monitor* yang terintegrasi dalam suatu sistem jaringan secara *online* atau biasa juga di implementasikan secara manual.

Tujuan dari setiap orang menggunakan CCTV adalah untuk memantau daerah yang luas dan mungkin jauh dari suatu lokasi yang sulit dikontrol dan dijangkau pada saat waktu yang bersamaan.

2.3.1 Cara Kerja Video Surveillance

Video surveillance adalah suatu kamera yang berguna untuk mengawasi kegiatan yang berada di lokasi tertentu. Cara kerja dan arsitektur *video surveillance* dapat dilihat pada gambar 2.8:



Gambar 2.8, Cara kerja *video surveillance*

Keterangan mengenai skema di atas:

1. *Location tracking integration*: gerakan pendeteksian yang mengikuti jalan pekerjaan suatu objek.
2. *Event occurrence integration and coordination*: korelasi aktivitas untuk keputusan akhir.
3. *Function integration*: deteksi untuk IDS, VoIP, dan menggabungkan dengan WLAN yang ada untuk sistem keamanan.

Dari gambar dapat disimpulkan bahwa cara kerja *video surveillance*, yaitu:

1. *Video* dari masing-masing kamera berhasil diperoleh karena kamera bergerak mengikuti langkah-langkah, yakni: mengisyaratkan pendeteksian, mengikuti jalan objek, penggolongan objek, dan pendeteksian kegiatan.
2. Hubungkan database yang mungkin digunakan dalam pendeteksian dan penggolongan.
3. Modul visual digunakan menyajikan berbagai data secara efisien dan mungkin dapat menunjang hasil deteksi

2.3.2 Fungsi Video Surveillance

Implementasi *video surveillance* di Indonesia diterapkan pada tempat umum seperti kampus, hotel, jalan, bandara dan lain sebagainya. Banyak tujuan dari pemasangan *video surveillance* ini antara lain:

1. Merekam 24 jam dan *online* jarak jauh melalui berbagai media.
2. Memonitor aktivitas di suatu lokasi dan fleksibel (dapat digerakan).
3. Jika terjadi tindak kriminal, dapat dijadikan bukti yang kuat.

2.4 Deteksi Kendaraan

Pada deteksi kendaraan diperlukan beberapa proses yang harus dilakukan terhadap *video surveillance* yang didapatkan, yaitu: *preprocessing*, *segmentasi* dan *noise reduction*. Berikut adalah metode yang digunakan oleh peneliti untuk proses-proses tersebut:

2.4.1 Grayscale

Metode ini termasuk dalam proses *preprocessing*, proses ini paling sering digunakan untuk *image processing* adalah pengubahan warna gambar ke bentuk *grayscale*, hal ini dikarenakan untuk menyederhanakan model gambar. Untuk mengubah gambar berwarna yang mempunyai nilai r, g dan b ke dalam *grayscale* dengan nilai s, maka konversi dapat dilakukan dengan mengambil nilai rata-rata dari r, g dan b.

$$\text{Grayscale } s = \frac{r+g+b}{3} \quad \dots (1)$$

2.4.2 Image Segmentation

Metode ini termasuk dalam proses *preprocessing*, *image segmentasi* (*Segmentation* gambar) adalah pemisahan objek satu dengan objek yang lain dalam suatu gambar. Ada 2 macam *segmentasi*, yaitu *full segmentasi* dan *partial segmentasi*. *Full segmentasi* adalah pemisahan objek secara *individu* dari *background* dan diberikan ID pada tiap-tiap *segmentasi*. *Partial Segmentation* adalah pemisahan sejumlah data dari *background* dimana data yang disimpan hanya data yang dipisahkan saja untuk mempercepat proses selanjutnya. Ada 3 tipe pada *segmentasi* yaitu:

- a. *Classification – based*: *segmentasi* berdasarkan dari kesamaan suatu ukuran *pixel* gambar.
- b. *Edge – based*: mencari garis yang ada pada gambar dan garis tersebut digunakan untuk pembatas tiap segment.
- c. *Region – based*: *segmentasi* berdasarkan kumpulan kesamaan *pixel* (warna, texture atau tingkat *graycale*) dari titik yang telah ditentukan.

2.4.3 Frame Difference

Metode ini termasuk dalam proses *segmentasi*, *frame difference* atau yang lebih dikenal *image subtraction* ini digunakan untuk mendeteksi dan *segmentasi* pada penelitian ini adalah dengan menghitung perbedaan setiap *frame*. Metode ini didesain untuk situasi yang stabil. Dimana kamera yang digunakan tidak bergerak, tidak ada perubahan cahaya yang tiba-tiba, serta tidak ada bayangan yang menaggu pengambilan gambar.

Metode ini bekerja berdasarkan pada perubahan waktu sekarang (t) dengan waktu sebelumnya (t-1), untuk bentuk matematisnya menjadi:

$$FD(x,y,t) = |I(x,y,t) - I(x,y,t-1)| \quad \dots (2)$$

Dimana:

FD = nilai *pixel* (x,y,t)= titik ke (x,y) dalam waktu saat ini

I = data dari *frame* (x,y,t-1)= titik ke (x,y) dalam waktu sebelumnya

2.4.4 *Thresholding*

Metode ini termasuk dalam proses *segmentasi*, *thresholding* adalah proses mengubah gambar berderajat keabuan menjadi gambar biner atau hitam putih sehingga dapat diketahui daerah mana yang termasuk obyek dan *background* dari gambar secara jelas. Gambar hasil *thresholding* biasanya digunakan lebih lanjut untuk proses pengenalan obyek serta ekstraksi *feature*.

Metode *thresholding* secara umum dibagi menjadi dua, yaitu:

1. *Thresholding global*, *Thresholding* dilakukan dengan mempartisi histogram dengan menggunakan sebuah *threshold* (batas ambang) *global* T, yang berlaku untuk seluruh bagian pada gambar.
2. *Thresholding adaptif*, *Thresholding* dilakukan dengan membagi gambar menggunakan beberapa sub gambar. Lalu pada setiap sub gambar, *segmentasi* dilakukan dengan menggunakan *threshold* yang berbeda.

Didalam *thresholding* terdapat berbagai macam metode yang digunakan, seperti *binary threshold*, *binary inverted threshold*, *threshold truncated*, *threshold to zero* dan *threshold to zero inverted*. Untuk penelitian ini nantinya akan dipilih metode *threshold* yang paling memadai untuk *segmentasinya*.



Gambar 2.9, Operasi *thresholding*

2.4.5 Dilatasi

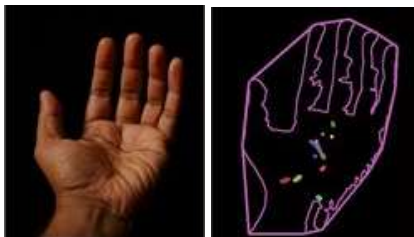
Metode ini termasuk dalam proses *noise Reduction*, Dilatasi adalah operasi *morphologi* yang akan menambahkan *pixel* pada batas antar objek dalam suatu gambar digital. Operasi ini menggunakan aturan sebagai berikut: “Untuk gambar *grayscale* maka nilai hasil operasi (*output pixel*) adalah nilai maksimal yang diperoleh dari himpunan *pixel* tetangganya. Dalam *binary image*, jika ada *pixel* tetangga yang bernilai 1 maka *output pixel* akan berubah menjadi 1”.

2.4.6 Erosi

Metode ini termasuk dalam proses *noise reduction*, Erosi adalah operasi *morphologi* yang akan mengurangi *pixel* pada batas antar objek dalam suatu gambar digital. Operasi ini menggunakan aturan sebagai berikut: “Untuk gambar *grayscale* maka nilai hasil operasi (*output pixel*) adalah nilai minimal yang diperoleh dari himpunan *pixel* tetangganya. Dalam *binary image*, jika ada *pixel* tetangga yang bernilai 0 maka *output pixel* akan berubah menjadi 0”.

2.4.7 Convex Hull

Metode ini termasuk dalam proses *noise reduction*, metode Convex hull merupakan persoalan klasik dalam geometri komputasional, persoalan tersebut digambarkan secara sederhana dalam ruang dimensi dua (bidang) sebagai mencari *subset* dari himpunan titik pada bidang tersebut sedemikian rupa sehingga jika titik-titik tersebut dijadikan poligon maka akan membentuk poligon yang konveks. Suatu poligon dikatakan konveks jika digambarkan garis yang menghubungkan antar titik maka tidak ada garis yang memotong garis yang menjadi batas luar poligon. Definisi lain dari Convex hull adalah poligon yang disusun dari *subset* titik sedemikian sehingga tidak ada titik dari himpunan awal yang berada di luar poligon tersebut. Proses ini digunakan agar hasil *threshold* yang berlebihan tidak dianggap sebagai objek dalam hal ini kendaraan.



Gambar 2.10, Operasi Convex hull

2.4.8 BLOB

BLOB (*Binnary Large Object*) adalah sebuah kumpulan dari *pixel* yang saling terhubung didalam sebuah gambar yang memiliki kesamaan *property* (warna, *texture* atau tingkat *grayscale*). Kegunaan utama dari blob ini adalah untuk mendeteksi objek yang telah diproses oleh Convex hull sehingga dianggap sebagai kendaraan.

2.5 Klasifikasi Kendaraan

Untuk dapat melakukan klasifikasi pada kendaraan yang bergerak melewati jalur tertentu diberikan data set dari masing-masing kelas. Pada penelitian ini klasifikasi kendaraan akan dibagi menjadi 2 jenis yaitu kendaraan motor atau mobil. Beberapa metode digunakan untuk klasifikasi kendaraan ada bermacam macam dari yang paling simple menghitung area kendaraan hingga *learning*.



Gambar 2.11, Metode klasifikasi berdasarkan ukuran kendaraan

Pada gambar 2.11 menggunakan metode berdasarkan ukuran. Namun, metode ini hanya dapat digunakan per-lajur dan membutuhkan ROI yang memadai sehingga panjang dan lebar kendaraan dapat diketahui.

2.6 Violation Detection

Pada proses pelanggaran yang terjadi pada area lampu lalu lintas, ada beberapa proses yang harus dilakukan terhadap *video surveillance* yang didapatkan diantaranya:

2.6.1 Switch Mode

Untuk mendapatkan mode lampu yang sesuai maka dibutuhkan 2 *video* berbeda dan berjalan bersamaan. Untuk penghitungan dan memutuskan lampu sedang merah atau hijau dapat dilakukan dengan cara mengambil *sample pixel* pada *video* lampu dan mengkalkulasi warna yang didapatkan kemudian dilakukan perbandingan.

$$\text{Mode Merah} = \text{RedValue}(x1,y1) > \text{GreenValue}(x2,y2) \quad \dots (3)$$

$$\text{Mode Hijau} = \text{GreenValue}(x2,y2) > \text{RedValue}(x1,y1) \quad \dots (4)$$

Keterangan :

Redvalue = tingkat merah *pixel* $(x1,y1)$ = posisi *pixel* merah yang dipilih

Greenvalue = tingkat hijau *pixel* $(x2,y2)$ = posisi *pixel* hijau yang dipilih

Metode yang dapat digunakan selain pedeteksian *pixel* adalah dengan cara memilih warna *dominant*, namun karena *resource* yang kurang karena proses yang lama dan berulang.

2.6.2 ROI - Garis Pelanggar Merah

Proses ini akan bekerja pada saat lampu sedang merah, ROI adalah *region of interest* area yang akan di jadikan sebagai media observasi guna mengetahui keadaan apakah terjadi pelanggaran atau tidak. ROI yang akan diberikan pada adalah pada *video surveillance* yang dipakai adalah diarea *zebra cross*. Ketika terdapat bagian tengah kendaraan yang berada diatas garis atau melewatinya maka akan dapat dianggap sebagai pelanggaran.



Gambar 2.12, ROI – Garis pelanggar untuk lampu merah

2.6.3 ROI - Garis Pelanggar Hijau

Proses ini akan bekerja pada saat lampu sedang hijau, blob tracking akan memberikan data perpindahan kendaraan dari tiap-tiap frame. Sehingga dapat diketahui arah perpindahan dan posisi kendaraan yang terdeteksi oleh blob. Pemakaian garis pelanggar ini dapat digunakan diantara tiap-tiap lajur. Untuk deteksi pelanggaran untuk marka akan dianggap pelanggaran apabila keseluruhan bagian kendaraan melewati ROI garis pelanggar.

Proses *optical flow* yang biasa digunakan dapat dihilangkan dengan begitu akan mengurangi proses program sehingga lebih efisien. Penghilangan proses diantaranya *feature extraction* dan *optical flow* sendiri. Kedua proses tersebut telah digantikan dengan blob detection dan blob tracking, dimana posisi fitur yang didapat untuk tiap kendaraan dapat dipilih.



Gambar 2.13, ROI – Garis pelanggar untuk lampu hijau

Pada pemberian ROI – garis pelanggar baik pada lampu hijau dan lampu merah sendiri terdapat perbedaan posisi yang jauh, oleh karena itu penghitungan dalam klasifikasi mobil atau motor akan diketahui jika blob melewati garis saja. Hal tersebut dilakukan karena klasifikasi berdasarkan area tidak dianjurkan pada *camera* dengan sudut pandang *bird view*.

Untuk proses klasifikasi terdapat nilai minimal area untuk mendeteksi mobil atau ketika selain itu maka akan dianggap sebagai motor. Karena itu terdapat proses kalibrasi sebelum melakukan proses *realtime automation* pada program.

2.8 Referensi Penelitian

2.8.1 Meanshift Blob Tracking with Target Model adaptive Update - ZHAO Yunji, ZHANG Bin, ZHANG Xinliang - 2014

Sebuah mekanisme pembaruan model adaptif untuk pelacakan pergeseran rata-rata. Gaussian Model selalu telah digunakan dalam estimasi model latar belakang untuk merealisasikan pelacakan objek. Ini adalah novel yang masing-masing sampah dari histogram kernel dimodelkan sebagai campuran Gaussian dan pendekatan on-line yang digunakan untuk memperbarui model. Distribusi Gaussian diperintahkan berdasarkan nilai kebugaran berat dan kovarian. Model objek ditentukan oleh masing-masing distribusi dan berat gaussian. Oleh karena itu pergeseran rata-rata yang ditingkatkan tidak hanya dapat memperbarui model objek dalam waktu tetapi juga berurusan dengan perubahan penampilan objek dan kemacetan. Eksperimen menunjukkan bahwa metode yang ditingkatkan dapat melacak objek di bawah perubahan penampilan dan oklusi dengan hasil yang memuaskan.

Pada implementasinya pada penelitian penggunaan meanshift mendukung untuk multi objek tracking. Sehingga digunakan dan menghilangkan proses extraction feature guna mencari keypoint dan optical flow guna mengamati tracking.

2.8.2 Btraffwatch - Hario Baskoro Basoeki – 2013

Kurangnya kesadaran akan kepatuhan terhadap peraturan lalu lintas sering menjadi penyebab pelanggaran lalu lintas, area lampu lalu lintas merupakan pelanggaran umum dari tanda yang disengaja oleh pengguna jalan. Studi ini menciptakan sebuah sistem surveilans yang bisa mendeteksi pelanggaran yang terjadi di lalu lintas. Pelanggaran dibagi menjadi dua, yaitu pelanggaran lampu merah dan saat lampu hijau menyala. Pelanggaran itu terjadi saat lampu merah dan lampu hijau bisa terdeteksi. Setelah melakukan percobaan menggunakan 10 *video* yang menggambarkan kondisi lampu merah dan 10 *video* yang menggambarkan kondisi lampu hijau, sistem dapat mencapai keakuratan 85,83% untuk mendeteksi benda yang melanggar lampu merah saat menyala dan 96,67% untuk pelanggaran pada saat lampu hijau menyala [9].

No	Data uji	Jumlah Kendaraan		Tingkat Keberhasilan
		Pelanggar	Terdeteksi Melanggar	
1	Video 1	3 Kendaraan	3 Kendaraan	100%
2	Video 2	3 Kendaraan	2 Kendaraan	66,66%
3	Video 3	4 Kendaraan	2 Kendaraan	50%
4	Video 4	2 Kendaraan	1 Kendaraan	50%
5	Video 5	2 Kendaraan	2 Kendaraan	100%
6	Video 6	3 Kendaraan	3 Kendaraan	100%
7	Video 7	4 Kendaraan	4 Kendaraan	100%
8	Video 8	0 Kendaraan	0 Kendaraan	100%
9	Video 9	3 Kendaraan	3 Kendaraan	100%
10	Video 10	1 Kendaraan	1 Kendaraan	100%
Rata - rata keberhasilan				86,67 %

Gambar 2.15

Gambar 2.15, Hasil Pengujian Deteksi Pelanggaran Lampu Hijau di Lajur 1

No	Data uji	Jumlah Kendaraan		Tingkat Keberhasilan
		Pelanggar	Terdeteksi Melanggar	
1	Video 1	4 kendaraan	4 kendaraan	100%
2	Video 2	4 kendaraan	4 kendaraan	100%
3	Video 3	5 Kendaraan	5 Kendaraan	100%
4	Video 4	3 Kendaraan	2 Kendaraan	66,67%
5	Video 5	1 Kendaraan	1 Kendaraan	100%
6	Video 6	0 Kendaraan	0 Kendaraan	100%
7	Video 7	1 Kendaraan	1 Kendaraan	100%
8	Video 8	0 Kendaraan	0 Kendaraan	100%
9	Video 9	0 Kendaraan	0 Kendaraan	100%
10	Video 10	4 Kendaraan	4 Kendaraan	100%
Rata - rata keberhasilan				96,67%

Gambar 2.16

Gambar 2.16, Hasil Pengujian Deteksi Pelanggaran Lampu Merah di Lajur 1

Pada penelitian ini *violation detection* pada lampu merah digunakan *contour changing observation*, yang berarti perubahan ROI menandakan adanya pelanggaran oleh *contour* yang overlap/bertabrakan dengannya. Sedangkan pada lampu hijau menggunakan *feature extraction* dengan metode *haris-conner detection* dan Lucas-Kanade *optical flow*, untuk *tracking* gerak kendaraan apakah sesuai jalur atau tidak. Perubahan metode pada *optical flow* dan rule pada *violation detection* diharapkan dapat meningkatkan akurasi baik dari klasifikasi dan tingkat kebenaran data pelanggaran.

Pada Btraffwatch sendiri terdapat masih menggunakan metode basic seperti pedeteksi contour dan optical flow, dengan efisiensi yang lebih tinggi blob dapat juga dapat digunakan untuk mengganti 2 fungsi itu. Paper ini ditunjukan sebagai pembandingan hasil akhir sehingga dapat diketahui metode mana yang memiliki akurasi lebih baik.

2.8.3 Shadow elimination and vehicles classification approaches in traffic video surveillance context – H. Asaidi – 2014

Video surveillance di jalan raya adalah topik hangat dan tantangan besar dalam *Intelligent Transportation Systems*. Dalam aplikasi seperti yang membutuhkan ekstraksi objek, bayangan bayangan menyebabkan distorsi bentuk dan fusi objek mengganggu kinerja algoritma. Penghapusan bayangan memungkinkan untuk meningkatkan kinerja ekstraksi objek *video*, klasifikasi dan *tracking*. Di sisi lain, sangat penting untuk mengenali jenis objek yang terdeteksi agar dapat melacak andal dan memperkirakan parameter lalu lintas dengan benar. Makalah ini menyajikan dua pendekatan untuk meningkatkan sistem pengawasan lalu lintas otomatis. Pertama berkaitan dengan penghapusan bayangan dan yang kedua menyangkut klasifikasi kendaraan, berdasarkan penglihatan dan pengolahan gambar yang kuat. Untuk

memindahkan bayangan eliminasi, model kontras diusulkan untuk menggambarkan dan menghilangkan bayangan dinamis berdasarkan gagasan bahwa transformasi bayangan adalah perubahan yang kontras. Untuk klasifikasi kendaraan, momen Hue dihitung dengan cara mengurangi efek perspektif dan digunakan untuk menggambarkan kendaraan dalam basis pengetahuan. Hasil percobaan pada berbagai rangkaian *video* yang menantang menunjukkan bahwa pendekatan yang diusulkan mengungguli metode klasifikasi dari pekerjaan terkait (dengan akurasi klasifikasi 96,96%), dan bahwa pendekatan penghapusan bayangan melakukan pekerjaan yang lebih baik daripada dibandingkan (dengan tingkat deteksi 95-99% dan Tingkat diskriminasi 85,7-89%)[10].

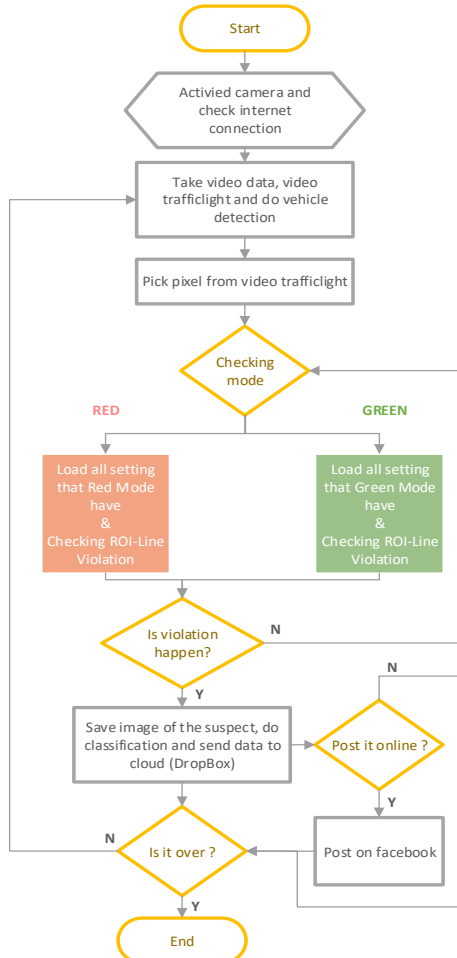
Pada penelitian ini meningkatkan akurasi klasifikasi dengan cara *shadow elimination* sehingga *feature extraction* yang dilakukan untuk mendapat *point track* dapat lebih efektif oleh karena itu penelitian sehingga data yang diperoleh dapat memiliki tingkat akurasi yang lebih tinggi dari pada penelitian sebelumnya.

Perubahan atau pre-proses pada gambar sangat menguntungkan dalam penentuan ekstraksi fitur, pada program ini yang dipakai juga sangat membantu oleh karena itu pada ke program pada bagian *preprocessing* ditambahkan hue proses untuk penghilangan bayangan.

BAB III PERANCANGAN DAN IMPLEMENTASI SISTEM

3.1 Perancangan Penelitian

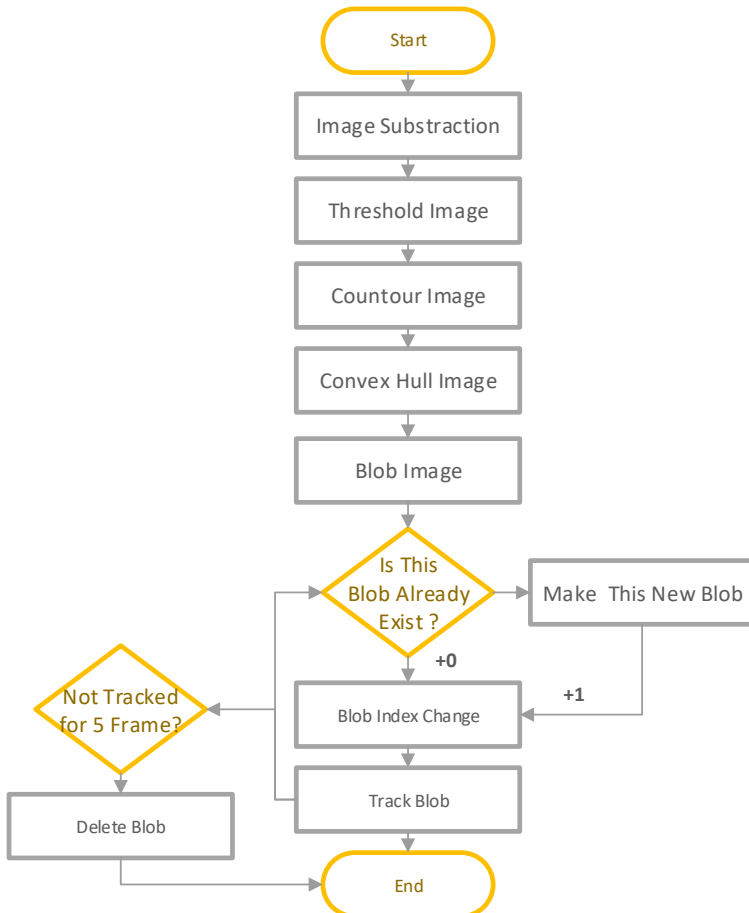
Pada penelitian ini dibuat sistem yang dapat mengklasifikasi dan mendeteksi pelanggaran pada lampu lalu lintas. Berikut adalah diagram alur yang dari sistem yang akan dibuat:



Gambar 3.1, Diagram alur sistem yang akan dibuat

3.2 Perancangan Deteksi Kendaraan

Berikut adalah diagram alur dari proses yang akan dibuat:



Gambar 3. 2, Diagram alur proses perancangan deteksi kendaraan

Sebelum masuk pada pendeteksian pelanggaran, proses yang akan dilakukan program mengikuti beberapa hal. Oleh karena itu tahap pertama yang dilakukan pada pembuatan program adalah dengan membuat klasifikasi dimana kendaraan dapat diketahui posisi dan jenisnya. Sehingga lebih mudah dalam mendeteksi pelanggarannya

3.2.1 Image Substraction

Pada *image subtraction* ini dilakukan proses pengambilan *foreground* dari *background* gambar, dengan cara mengurangi nilai *pixel frame* sekarang dengan nilai *pixel frame* sebelumnya. Berikut kode program yang digunakan pada proses ini.

```
cvtColor(imgFrame1Copy, imgFrame1Copy, CV_BGR2GRAY);  
cvtColor(imgFrame2Copy, imgFrame2Copy, CV_BGR2GRAY);  
  
GaussianBlur(imgFrame1Copy, imgFrame1Copy, cv::Size(5, 5), 0);  
GaussianBlur(imgFrame2Copy, imgFrame2Copy, cv::Size(5, 5), 0);  
  
absdiff(imgFrame1Copy, imgFrame2Copy, imgDifference);
```

Berikut adalah hasil yang didapatkan dari proses tersebut.



Gambar 3. 3, Hasil *image subtraction*

3.2.2 Threshold

Pada proses ini gambar akan diperjelas dengan memberikan *threshold* yang dapat disesuaikan melalui *trackbar*. Berikut kode program yang digunakan pada proses ini.

```
threshold(imgDifference,      imgThresh,      slider1,      slider2,  
CV_THRESH_BINARY);
```

Berikut adalah hasil yang didapatkan dari proses tersebut.



Gambar 3.4, Hasil *threshold image*

3.2.3 *Contour*

Pada proses ini gambar dari *threshold* akan dilakukan pembentukan lagi agar lebih padat dengan cara dilatasi dan erosi. Untuk proses ini nilai yang diberikan dapat bervariasi dapat menggunakan 3x3, 5x5 atau 7x7 sesuai dengan kebutuhan, sehingga gambar yang dibentuk terlihat jelas. Berikut kode program yang digunakan pada proses ini.

```
for (unsigned int i = 0; i < 2; i++) {  
    dilate(imgThresh, imgThresh, structuringElement3x3);  
    erode(imgThresh, imgThresh, structuringElement3x3);  
}  
Mat imgThreshCopy = imgThresh.clone();  
vector<vector<Point> > contours;  
Contours(imgThreshCopy, contours, RETR_EXTERNAL, CHAIN_APPROX_SIMPLE)  
;
```

Berikut adalah hasil yang didapatkan dari proses tersebut.



Gambar 3.5, Hasil *contour image*

3.2.4 Convex Hull

Pada tahap ini dilakukan pengurangan *noise* dengan proses Convex hull. Penggunaan Convex hull disini adalah membentuk gambar sehingga tidak ada bagian yang lepas dari satu *contour*. Berikut kode program yang digunakan pada proses ini. Berikut kode program yang digunakan pada proses ini.

```
vector< vector< Point> > convexHulls(contours.size());  
  
for (unsigned int i = 0; i < contours.size(); i++)  
{ convexHull(contours[i], convexHulls[i]); }
```

Berikut adalah hasil yang didapatkan dari proses tersebut



Gambar 3.6, Hasil Convex hull pada *contour*

3.2.5 Blob

Blob (*Binnary large objek*) adalah proses yang digunakan untuk membentuk objek dimana memiliki nilai yang sama, karena blob hanya memproses *contour* maka objek yang memiliki *contour* yang sama yang akan dijadikan satu. Kemudian untuk menghilangkan *noise* maka akan ada *decision* yang akan menentukan apakah ukuran tersebut termasuk kendaraan atau hanya *noise*.

Untuk *contour* yang termasuk ukuran kendaraan maka akan diproses kalau tidak termasuk maka akan langsung tidak dianggap. Untuk ukuran kendaraan dapat ditentukan oleh *user* menggunakan slider dengan menaikan batasan terendah untuk motor dan batasan terendah untuk mobil, sehingga penggunaan program dapat dilakukan pada perempatan manapun yang diawasi CCTV pada jalan dan lampu lalu lintasnya. Pada blob ini akan terdapat proses dimana penggambaran kotak sehingga akan diketahui apakah objek tersebut adalah kendaraan.

Berikut kode program yang digunakan pada proses ini.

```
for (auto &convexHull : convexHulls) {  
    Blob possibleBlob(convexHull);  
    if (possibleBlob.currentBoundingRect.area() > 450 &&  
        possibleBlob.currentBoundingRect.area() < 25000 &&  
        possibleBlob.dblCurrentAspectRatio > 0.2 &&  
        possibleBlob.dblCurrentAspectRatio < 4.0 &&  
        possibleBlob.currentBoundingRect.width > 30 &&  
        possibleBlob.currentBoundingRect.height > 30 &&  
        possibleBlob.dblCurrentDiagonalSize > 60.0 &&  
        (cv::contourArea(possibleBlob.currentContour) /  
         (double)possibleBlob.currentBoundingRect.area()) > 0.50)  
    {  
        currentFrameBlobs.push_back(possibleBlob);  
    }  
}
```

Berikut adalah hasil yang didapatkan dari proses tersebut.



Gambar 3.7, Hasil Blob pada Convex hull

3.2.6 New Blob or Existing Blob

Pada umumnya blob hanya digunakan untuk mendeteksi sesuatu *frame*, bukan sesuatu pada *frame to frame*, sehingga dibutuhkan beberapa tambahan fungsi program untuk mendefinisikan apakah ini blob baru atau blob ini sudah ada.

Blob akan dimasukan kedalam fungsi untuk mengecek apakah blob baru atau tidak. Pada fungsi tersebut terdapat varibel pendukung seperti jarak minimal prediksi dan pengambilan fitur pada blob (*centerpostion*)

```
int indexOfLeastDistance = 0;
double dblLeastDistance = 100000.0;
for (unsigned int i = 0; i < existingBlobs.size(); i++) {
    if (existingBlobs[i].blnStillBeingTracked == true) {
        double dblDistance = distanceBetweenPoints(currentFrameBlob.
            centerPositions.back(),existingBlobs[i].predictedNextPosition);
        if (dblDistance < dblLeastDistance) {
            dblLeastDistance = dblDistance;
            indexOfLeastDistance = i;
        }
    }
}
```

Syarat untuk menjadi blob baru adalah jarak yang terlalu jauh (prediksi salah) atau memang tidak ada prediksinya sejak awal.

```
if (dblLeastDistance < currentFrameBlob.dblCurrentDiagonalSize*0.5)
{
    addBlobToExistingBlobs(currentFrameBlob, existingBlobs,
        indexOfLeastDistance);
} else {
    addNewBlob(currentFrameBlob, existingBlobs);
}
```

Blob akan dilepas dari feature track ketika tidak terlihat pada frame untuk 5 frame beruntun. 5 frame dipilih karena semakin banyak blob terdeteksi dan hilang maka masih akan ada 5 frame proses lagi untuk mendeteksi blob tersebut, cukup memakan resource dan dapat membuat program menjadi lambat.

```
if (existingBlob.blnCurrentMatchFoundOrNewBlob == false) {
    existingBlob.intNumOfConsecutiveFramesWithoutAMatch++;
}
if (existingBlob.intNumOfConsecutiveFramesWithoutAMatch >= 5) {
    existingBlob.blnStillBeingTracked = false;
}
```

3.2.7 Blob Index Change

Blob *index* ini berkaitan pada pendeteksian kendaraan karena setiap kendaraan yang terdeteksi akan diberi label menurut *index*. Karena itu maka blob *index* akan dianggap jumlah kendaraan terdeteksi.

3.2.8 Track Blob

Pada tahap ini blob yang terdeteksi harus dilakukan *tracking* sehingga ketika terjadi *collision* agar tidak mengganggu jalannya program. *Collision* sendiri ada 2 yaitu *collision* pada saat blob dibentuk dan *collision* saat blob sudah dibentuk. Untuk *collision* saat dibentuknya blob tidak dapat dihindari lagi sedangkan untuk *collision* saat blob sudah dibentuk maka dapat dilakukan *tracking* untuk setiap blob. Dengan menggunakan prediksi ini *collision* akan dapat dihindari meskipun hanya *collision* ketika blob sudah terbentuk.

Tahap ini akan memprediksi posisi *center* blob, dengan cara melihat nilai *center* blob sampai 5 *frame* sebelumnya kemudian mencari nilai total perubahan pada X dan Y. Lalu mencari nilai deltaX dan deltaY dengan cara membulatkan nilai total perubahan pada X(untuk deltaX) dan Y(untuk deltaY) yang dibagi dengan nilai total *frame* (jika 5 maka 5!=10). kemudian posisi center blob sekarang dapat diprediksi dengan rumus berikut :

```
predictedPosition.x = centerPositionsX + deltaX;
predictedPosition.y = centerPositionsY + deltaY;    ... (3)
```

```
...
else if (numPositions >= 5) {
int sumOfXChanges =
((centerPositions[numPositions - 1].x -
centerPositions[numPositions - 2].x) * 4)
+((centerPositions[numPositions - 2].x -
centerPositions[numPositions - 3].x) * 3)
+((centerPositions[numPositions - 3].x -
centerPositions[numPositions - 4].x) * 2)
+((centerPositions[numPositions - 4].x -
centerPositions[numPositions - 5].x) * 1);
int deltaX = (int)std::round((float)sumOfXChanges / 10.0);

int sumOfYChanges =
((centerPositions[numPositions - 1].y -
centerPositions[numPositions - 2].y) * 4) +
((centerPositions[numPositions - 2].y -
centerPositions[numPositions - 3].y) * 3) +
((centerPositions[numPositions - 3].y -
centerPositions[numPositions - 4].y) * 2) +
((centerPositions[numPositions - 4].y -
centerPositions[numPositions - 5].y) * 1);
int deltaY = (int)std::round((float)sumOfYChanges / 10.0);
predictedNextPosition.x = centerPositions.back().x + deltaX;
predictedNextPosition.y = centerPositions.back().y + deltaY;
}
```

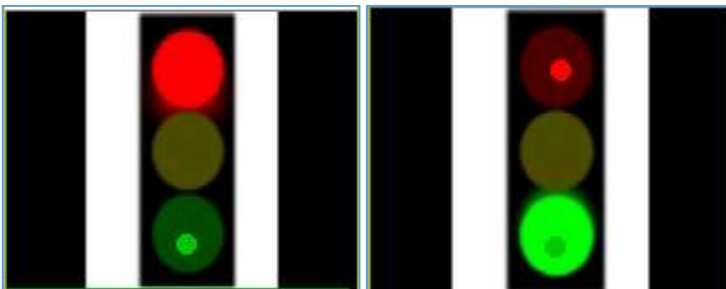
3.3 Switch Mode

Pergantian mode dari hijau ke merah dan sebaliknya, diperlukan *input video* lain yang memperlihatkan lampu lalu lintas. Jika tidak ada *video input* tersebut maka fungsi ini tidak bisa berjalan.

Switch mode pada dasarnya adalah pengecekan warna pada suatu gambar (setelah proses *crop*) apabila warna merah yang dominan maka menunjukkan lampu lalu lintas sedang berwarna merah, apabila warna hijau yang dominan maka menunjukkan lampu lalu lintas sedang berwarna hijau. Karena proses warna dominan terlalu panjang dan membutuhkan looping yang lama, maka untuk proses *switch mode* ini digunakan pengecekan warna pada 2 *pixel* yang dipilih, *pixel1* ditempatkan di area lampu dengan cahaya merah tinggi dan *pixel2* ditempatkan di area lampu dengan cahaya hijau tinggi. Berikut kode program yang digunakan pada proses ini.

```
MouseCallBack (){
    If(event == shift + klik kiri){
        Pixel1.X=x; Pixel1.Y=y;
    }
    If(event == shift + klik kanan){
        Pixel2.X=x; Pixel2.Y=y;
    }
}

...
RedValue = Mat Img.at<cv::Vec3b>(Pixel1)[2];
GreenValue = Mat Img.at<cv::Vec3b>(Pixel2)[1];
...
If(RedValue < GreenValue )
    { Mode : Hijau }
Else
    { Mode : Merah }
```

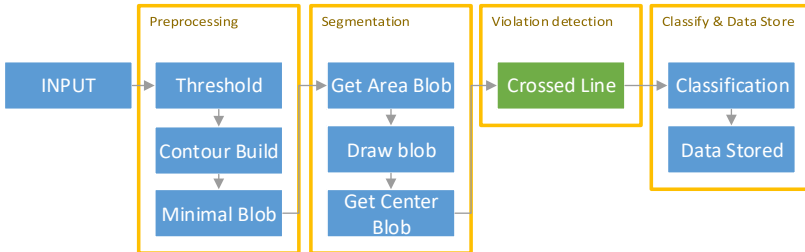


Gambar 3.8, Pergantian merah ke hijau & hijau ke merah

3.4 Deteksi Pelanggaran saat Lampu Hijau

3.3.1 Pembuatan Program

Berikut adalah *flow* dari bagaimana deteksi pelanggaran saat lampu hijau dapat terdeteksi.

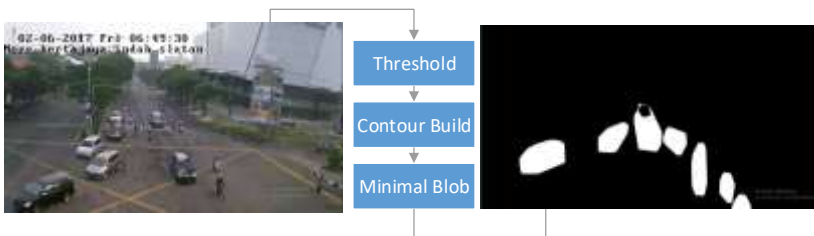


Gambar 3.9, Diagram alur sistem pendeteksi pelanggaran lampu hijau

Ada 4 tahapan yang harus dilewati untuk mendapatkan output yang benar, tahap pertama *preprocessing*, tahapan kedua segmentasi, tahapan ketiga *checking line ROI* dan tahap terakhir klasifikasi dan penyimpanan data.

3.3.2 Preporcessing

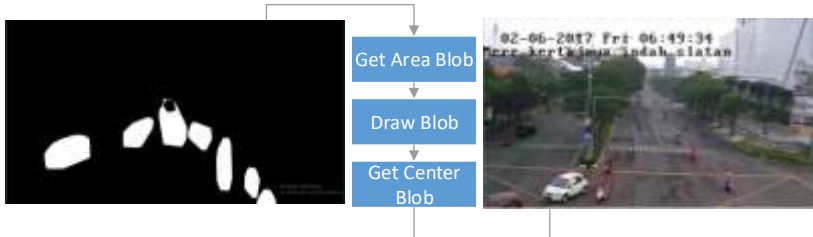
Setelah proses *image subtraction*. Tahapan pertama untuk menentukan baik buruknya output adalah threshold yang digunakan untuk mendeteksi objek layak atau tidak dianggap sebagai kendaraan. Setelah itu proses selanjutnya adalah contour build disini hasil threshold akan direkontruksi ulang menggunakan erosi dan dilatasi sehingga objek terlihat lebih jelas, *draw blob* dilakukan dari hasil *contour* yang telah direkontruksi. Berikut contoh hasil proses ini:



Gambar 3.10, Proses preprocessing pada deteksi pelanggaran lampu hijau

3.3.3 Segmentation

Tahap ini menggunakan blob sebagai dasar, dimulai dari mengambil area blob kemudian membandingkannya dengan *minimal* blob yang diperbolehkan lalu kemudian melakukan *feature extraction* proses dengan cara mengambil 1 titik *pixel* pada tengah blob.



Gambar 3.11, Proses segmentasi pada deteksi pelanggaran lampu hijau

3.3.4 Violation Detection

Tahap ini akan mengidentifikasi setiap adanya perpindahan jalur, namun karena observasi yang dilakukan agar mendapatkan data paling sesuai maka untuk pelanggaran marka akan dihitung ketika seluruh bagian kendaraan telah melewati garis.



Gambar 3.12, Proses *violation detection* pada deteksi pelanggaran lampu hijau

3.3.5 Classification & Data Store

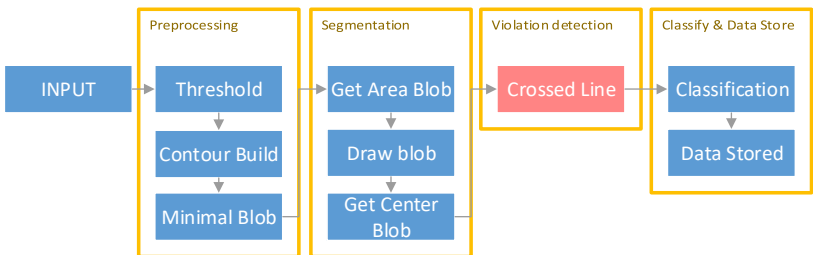
Untuk klasifikasi terdapat nilai minimal untuk mengatakan bahwa blob tersebut adalah motor atau mobil, dengan menggunakan panjang lebar dari suatu blob yang sesuai maka dapat melakukan klasifikasi dengan tepat.

Untuk penyimpanan data digunakan local ataupun *cloud* dapat menggunakan Dropbox, sehingga *file* yang disimpan itu juga dapat melakukan post pada social media Facebook.

3.5 Deteksi Pelanggaran saat Lampu Merah

3.4.1 Pembuatan Program

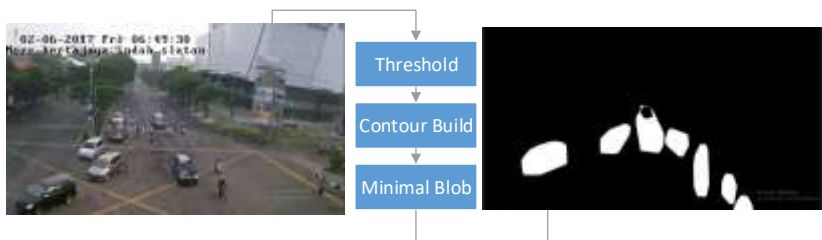
Berikut adalah *flow* dari bagaimana deteksi pelanggaran saat lampu merah dapat terdeteksi.



Gambar 3.13, Diagram alur sistem pendeteksi pelanggaran lampu merah

3.4.2 Preporcessing

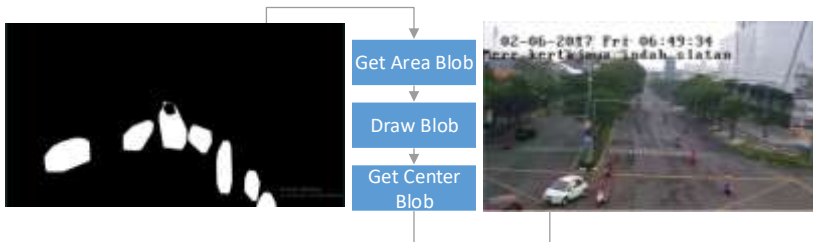
Setelah proses *image subtraction*. Tahapan pertama untuk menentukan baik buruknya *output* adalah *threshold* yang digunakan untuk mendeteksi objek layak atau tidak dianggap sebagai kendaraan. Setelah itu proses selanjutnya adalah *contour build* disini hasil *threshold* akan direkontruksi ulang menggunakan erosi dan dilatasi sehingga objek terlihat lebih jelas, *draw blob* dilakukan dari hasil *contour* yang telah direkontruksi.



Gambar 3.14, Proses preprocessing pada deteksi pelanggaran lampu merah

3.4.3 Segmentation

Tahap ini menggunakan blob sebagai dasar, dimulai dari mengambil area blob kemudian membandingkannya dengan *minimal* blob yang diperbolehkan lalu kemudian melakukan *feature extraction* proses dengan cara mengambil 1 titik *pixel* pada tengah blob.



Gambar 3.15, Proses segmentasi pada deteksi pelanggaran lampu merah

3.4.4 Violation Detection

Tahap ini akan mengidentifikasi setiap adanya perpindahan jalur, namun karena observasi yang dilakukan agar mendapatkan data paling sesuai maka untuk pelanggaran lampu akan dihitung ketika separuh bagian kendaraan telah melewati garis atau melewatinya.



Gambar 3.16, Proses *violation detection* pada deteksi pelanggaran lampu merah

3.4.5 Classification & Data Store

Untuk klasifikasi terdapat nilai minimal untuk mengatakan bahwa blob tersebut adalah motor atau mobil, dengan menggunakan panjang lebar dari suatu blob yang sesuai maka dapat melakukan klasifikasi dengan tepat.

Untuk penyimpanan data digunakan local ataupun *cloud* dapat menggunakan Dropbox, sehingga *file* yang disimpan itu juga dapat melakukan *post* pada *social media* Facebook.

3.6 Setting Dropbox

Dikarenakan program aplikasi ini menggunakan *cloud* komersil dan gratis maka diharapkan dapat melakukan instalasi pada PC dan mengikuti prosedur berikut :

1. Unduh aplikasi desktop.



Gambar 3.17, Laman instalasi Dropbox

2. Setelah terpasang, Anda akan diminta untuk masuk atau membuat akun baru.
3. Jika Anda menghubungkan akun Dropbox Business, pastikan Anda juga masuk ke akun ini juga.
4. Setelah Dropbox dipasang di komputer, Anda akan melihat *folder* Dropbox baru di hard drive. Anda juga akan melihat ikon Dropbox di baki sistem atau bilah menu.
5. Pastikan instalasi tersebut berada di “C://\$user\$/Dropbox”

Setelah itu program aplikasi akan dengan sendirinya membuat *folder* dan menata *file* menurut jenisnya menjadi 3 yaitu pelanggaran, report dan view.

3.7 Setting Faceboook

Untuk *social media* sendiri sangat disarankan menggunakan Facebook *pages* karena dengan adanya *pages* tersebut maka tidak hanya 1 orang yang dapat mengurusnya namun bisa beberapa orang sekaligus. Untuk membuat Facebook *pages* ikuti prosedur berikut :

1. Buka Facebook pada *browser*
2. Klik pada *profile* dan pilih buat halaman
3. Pilih Bisnis dan isi identitas dari halaman yang anda inginkan

-
- The screenshot shows the Facebook mobile app interface. At the bottom, there is a navigation bar with five icons. The third icon from the left, which represents a page, is highlighted with a blue bar. Below this bar, the text 'Buat Halaman' (Create Page) is visible. Other icons in the bar include a home icon, a search icon, a message icon, and a profile icon. The main content area of the app is partially visible, showing a post from 'Mila Sariyanti' and a post from 'Komunitas atau Teman Masyarakat'.

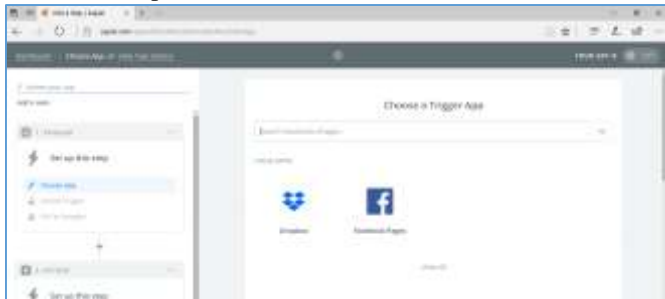
Setelah Pemasangan Facebook *pages* maka dapat dilakukan proses IFTT dimana ketika ada suatu *file* atau foto baru masuk kedalam Dropbox maka otomatis akan melakukan *posting*.

Zapier adalah *website* yang berguna untuk menjalankan automator atau IFTT selama pihak pengguna menyetujui syarat dan ketentuannya. Zapier sendiri bukanlah program gratis namun untuk pemakaian baru Zapier dapat digunakan (demo) selama 14 hari. Untuk membuat *post* otomatis dari suatu *file* baru yang masuk kedalam Dropbox diperlukan prosedur seperti berikut :

-
- The screenshot shows the Zapier website's interface for selecting apps to create workflows. At the top, there's a navigation bar with 'Zapier' logo, 'Home', 'Blog', 'Integrations', 'Tools & Apps', 'Contact Us', and a 'Sign Up' button. Below the navigation bar, there's a section titled 'Pick Apps to Explore Workflow Ideas'. This section includes a search bar with the text 'Search 1000+ apps' and a 'Filter' button. Below the search bar, there's a grid of app icons, each with a label underneath. The apps shown are: Google Sheets, Gmail, Slack, WhatsApp, Google Calendar, Facebook, Google Drive, YouTube, Microsoft Word, Facebook, Google Photos, Instagram, Amazon SageMaker, Google Contacts, and WhatsApp. The 'WhatsApp' app is highlighted with a red border.

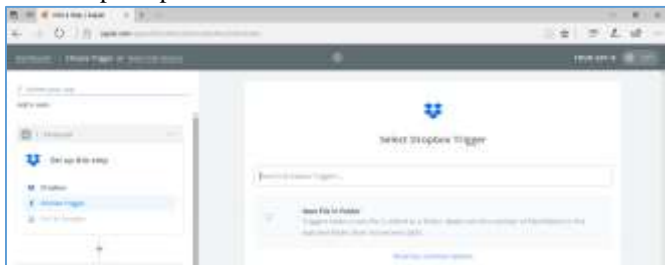
37

4. Pilih API Dropbox dan Facebook



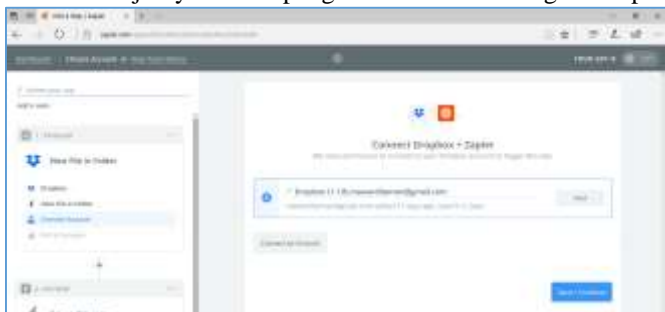
Gambar 3.20, Proses pengaitan Dropbox-Facebook step-1

5. Pada deopbox pilih New File in Folder



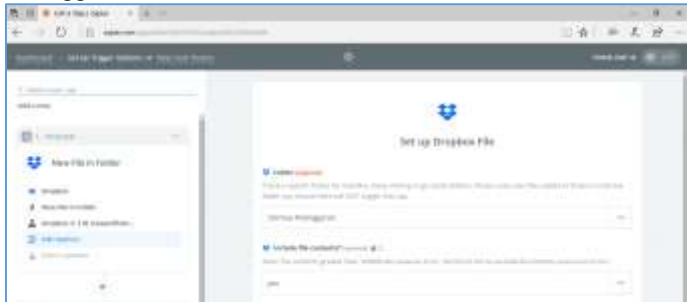
Gambar 3.21, Proses pengaitan Dropbox-Facebook step-2

6. Proses selanjutnya adalah pengecekan koneksi dengan Dropbox



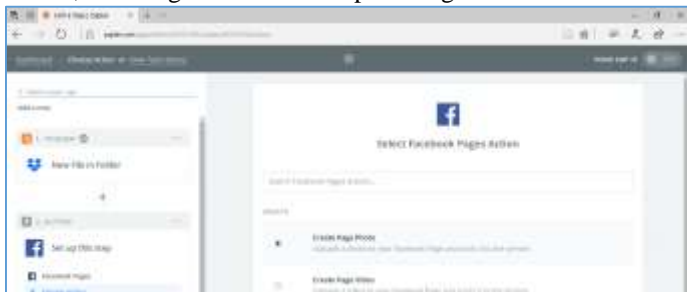
Gambar 3.22, Proses pengaitan Dropbox-Facebook step-3

7. Pilih Folder yang sesuai, pada contoh dipilih *folder* Semua Pelanggaran



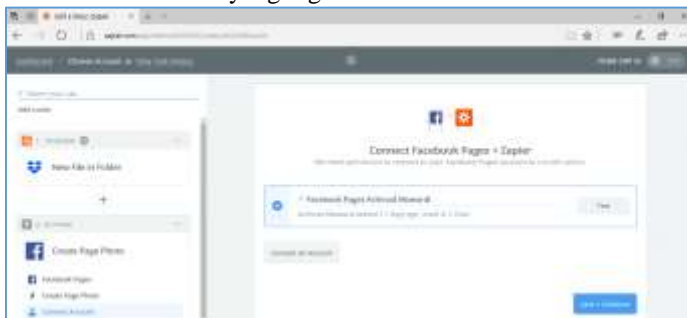
Gambar 3.23, Proses pengaitan Dropbox-Facebook step-4

8. Setelah itu klik continue dan koneksi Zapier dan Dropbox sudah selesai, sekarang atur koneksi Zapier dengan Facebook



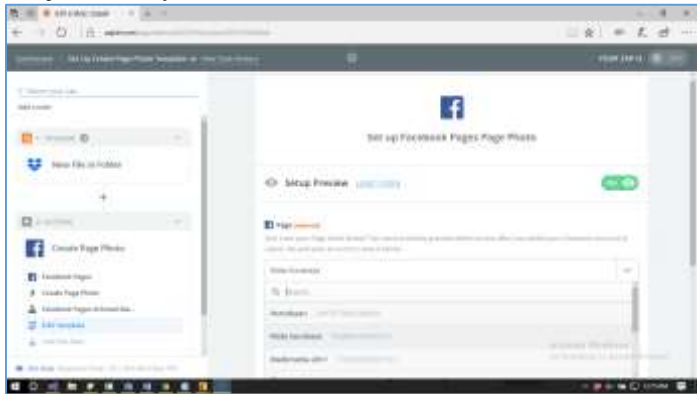
Gambar 3.24, Proses pengaitan Dropbox-Facebook step-5

9. Pilih akun Facebook yang ingin dikaitkan



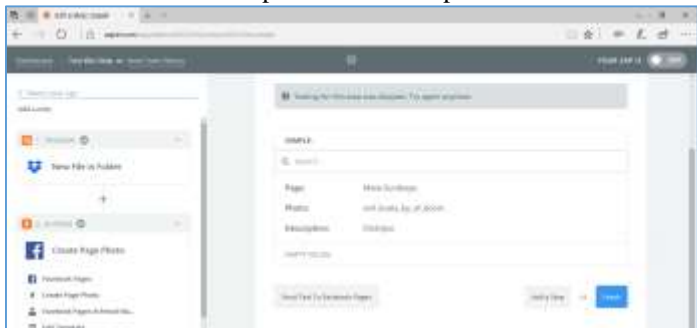
Gambar 3.25, Proses pengaitan Dropbox-Facebook step-6

10. Setelah itu koneksikan dengan pages dimana akun terkait menjadi adminnya.



Gambar 3.26, Proses pengaitan Dropbox-Facebook step-7

11. Setelah itu masukan diskripsi yang akan otomatis dilakukan kemudian lakukan test upload melalui Zapier.



Gambar 3.27, Proses pengaitan Dropbox-Facebook step-8

12. Dengan demikian proses pengaitan Zapier, Dropbox dan Facebook telah selesai dan setiap ada *file* baru masuk ke *folder* yang dipilih tadi maka akan ada *post* di Facebook mengai *file* tersebut.

3.9 Graphical User Interface

Platform yang digunakan pada program ini adalah *desktop* dengan *operating system* windows. Mengingat penggunaannya pada sistem pengawasan SITS (Surabaya *Intellegent Transport System*) juga demikian, namun untuk sistem report dapat digunakan *optional*.



Gambar 3.28, GUI platform Desktop

Untuk desain sendiri digunakan pendekatan fungsional yang berarti setiap tombol diletakan berdekatan (gruping) berdasarkan fungsinya. Pada form dialog sendiri terdapat 6 grup yaitu :

1. Loader (*highlight* hijau)

Pada bagian ini digunakan untuk memilih *file* dan *mode*. File ada 2 yaitu *file video* dan *file lampu*, sedangkan mode juga ada 2 yaitu simulasi untuk melakukan konfigurasi pengaturan dan automasi berjalan otomatis sesuai dengan perubahan dari *file* lampu.

2. Setting (*highlight* kuning)

Pada bagian ini digunakan untuk melakukan pengaturan pada program, semua pengaturan dapat secara langsung mengubah proses pada *video (realtime setting)* oleh karena itu sebelum melakukan mode automasi lebih baik mengaturnya terlebih dahulu pada mode simulasi.

3. Detail (*highlight cyan*)

Pengambilan detail ini digunakan untuk mengetahui komponen dari *video* sehingga *user* lebih nyaman menggunakannya. Pada detail sendiri terdapat tombol bantuan untuk menunjukkan *user* bagaimana program berjalan dan bagaimana penggunaan yang paling tepat.

4. All Data (*highlight pink*)

Semua data yang diambil dari *video* yang diputar setelah melalui berbagai proses.

5. Recent Data (*highlight orange*)

Penyimpanan data gambar pada *folder* untuk setiap pelanggaran yang terjadi, dan menampilkan gambar dan data pada GUI. Untuk menginformasikan pada *user* bahwa yang digunakan mendapatkan hasil yang seperti ini.

6. Video

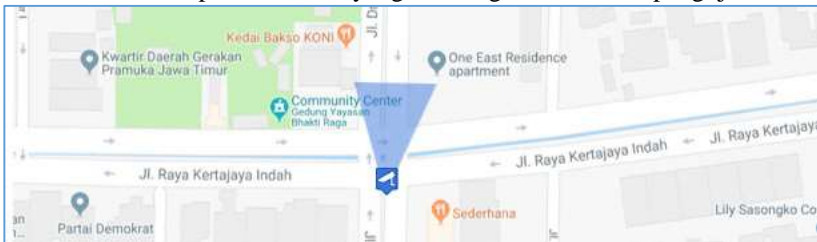
Pada bagian ini digunakan untuk memonitor semua aktivitas program mulai dari menjalankan *video*, penggambaran ROI-garis hingga deteksi pelanggaran

BAB IV PENGUJIAN DAN ANALISA

Pada bab ini akan menjelaskan tentang pengujian sistem yang telah selesai dikerjakan. Tahapan ini merupakan tahap penting dari pengerjaan proyek akhir. Pengujian ini dilakukan untuk memastikan bahwa program yang telah dibuat telah sesuai dengan yang telah direncanakan sebelumnya. Pengujian ini dilakukan secara bertahap dan berkesinambungan. Hasil dari setiap pengujian akan dianalisis untuk mendapatkan kebenaran pada tiap langkah yang dikerjakan. Hal ini diharapkan dapat mengurangi resiko kesalahan dan meningkatkan akurasi dari program.

Data yang digunakan pada pengujian ini diambil dari kamera yang digunakan oleh pihak Dishub Kota Surabaya. Peneliti telah mengirimkan surat kerjasama untuk memperoleh izin dalam pengambilan data untuk penelitian ini.

Berikut ini adalah posisi kamera yang akan digunakan untuk pengujian.



Gambar 4.1, Posisi hadap kamera CCTV yang digunakan

Kamera berada disebelang jalan atau berada pada depan WR. Sederahana yang menghadap ke Jl. Dr. Ir. H. Soekarno dari arah utara ke selatan. Pemasangan beradadi ketinggian 5 meter sehingga kondisi 5 lajur pada jalan terlihat jelas.

Pengujian akan dilakukan pada proses-proses berikut:

- | | |
|--|---|
| 1. Pengujian Blob (Deteksi kendaraan) | 4. Pengujian pelanggaran pada lampu (lampu merah) |
| 2. Pengujian Klasifikasi Kendaran | 5. Pengujian pelanggaran pada marka (lampu hijau) |
| 3. Pengujian Deteksi lampu-switch mode | 6. Pengujian <i>realtime</i> |
| | 7. Pengujian report |

4.1 Pengujian Blob (Deteksi kendaraan)

Input *video* pada area lampu lalu lintas berupa *file* mp4 yang didapatkan dari Dinas Perhubungan Surabaya, kemudian dilakukan pemotongan secara manual guna *sample* percobaan. Properties dari *video* tersebut masing-masing berukuran 720 *pixel* dengan 25 *frame per second*. Berikut contoh hasil yang didapatkan.



Gambar 4.2, Input video dari berbagai waktu

4.1.1 Proses Image Subtraction

Image Substraction atau *Image Difference* digunakan untuk mendeteksi objek bergerak. Pada penelitian ini adalah dengan melihat perubahan dari frame dikurang frame yang telah ditambah gaussianblur. Sehingga didapatkan *foreground* objek yang bergerak saja. Hasil yang didapatkan adalah *grayscale* karena kedua *frame operand* tersebut telah melalui proses *grayscale* sebelumnya.







Tabel 4.1: Image Substract

No.	Nama File	Input	Hasil Iamge Substract
1	DemoPagi1		
2	DemoSiang1		
3	DemoMalam1		

4.1.2 Proses *Contour Image*

Proses pencarian *contour image* lebih memperjelas *pixel* gambar foreground, pemakaian dilatasi dan erosi menjadi solusi yang terbaik oleh karena itu hasil yang didapatkan sesuai dan tidak saling menutupi. Berikut contoh hasil yang didapatkan.





Tabel 4.2: Contour Image



No.	Nama File	Input	Hasil Contour
1	DemoPagi1		
2	DemoSiang1		
3	DemoMalam1		

4.1.3 Proses BLOB

Object yang didapatkan dari proses *noise reduction* tersebut akan dianggap satu jika nilai objek tersebut terpenuhi, jika tidak maka akan dibuang.

Tabel 4.3: Blob Image

No.	Nama File	Input	Hasil Blob
1	DemoPagi1		
2	DemoSiang1		

No.	Nama File	Input	Hasil Blob
3	DemoMalam1		

4.1.4 Pengujian jumlah kendaraan

Pengujian ini dilakukan dengan cara menggunakan ROI-counter digunakan khusus ketika mode hijau untuk menghitung kendaraan yang melewatinya. Berikut adalah hasil yang diperoleh dari proses pengujian jumlah kendaraan:

Tabel 4.4: Hasil Pengujian jumlah kendaraan

No	Data	Mulai	Durasi	Real Counter	Program Counter	%
1	Malam-1	03:10:25	17	4	4	100%
2	Malam-2	03:15:34	19	5	6	93%
3	Malam-3	03:35:52	32	8	14	57%
4	Pagi-1	06:49:30	21	38	30	78%
5	Pagi-2	06:52:43	23	29	27	93%
6	Pagi-3	06:55:23	36	43	37	86%
7	Siang-1	11:57:48	21	102	65	63%
8	Siang-2	12:01:27	17	63	53	83%
9	Siang-3	12:05:44	19	42	48	87%
RATA-RATA						83%

Pada tabel 4.4, hasil akhir rata-rata untuk pendeteksian jumlah kendaraan sudah mencapai angka 83%. Dengan nilai tertinggi 100% dan nilai terendah 57%. Perbedaan yang cukup tinggi, namun hal itu terjadi karena beberapa faktor seperti keadaan cahaya, kepadatan kendaraan dan keadaan kamera.

Untuk keadaan cahaya akan menyebabkan *error* cahaya tersebut memantul ke aspal jalan, sehingga dianggap sebagai kendaraan. Untuk kepadatan kendaraan akan menyebabkan *error* ketika saling menutupi dan keadaan kamera akan menyebabkan *error* ketika terjadi gerakan tiba-tiba (bergerak) karena guncangan dari kendaraan besar yang melewatinya.

4.2 Pengujian Klasifikasi Kendaraan

Pengujian ini dilakukan dengan cara menggunakan ROI-counter digunakan khusus ketika mode hijau untuk menghitung kendaraan yang melewatinya. Klasifikasi dapat ditentukan oleh pengguna dengan mengubah nilai minimal dari motor atau mobil, untuk pengujian ini ukuran akan berada pada default untuk setiap waktu pengambilan gambar. Berikut adalah hasil yang diperoleh dari proses pengujian jumlah kendaraan:



Gambar 4.3, Penggunaan garis counter pada saat lampu hijau

Tabel 4.4: Hasil Pengujian jumlah kendaraan

No	Data	Real Motor	Program Motor	%	Real Mobil	Program Mobil	%
1	Malam-1	2	2	100%	2	7	28%
2	Malam-2	4	3	75%	1	2	50%
3	Malam-3	4	8	50%	4	6	60%
4	Pagi-1	33	24	72%	5	6	83%
5	Pagi-2	19	15	78%	10	12	83%
6	Pagi-3	17	8	47%	26	29	89%
7	Siang-1	42	24	57%	60	41	68%
8	Siang-2	15	9	60%	53	44	83%
9	Siang-3	15	12	80%	37	36	97%
RATA-RATA				69%			72%

Pada tabel 4.4 klasifikasi motor didapatkan akurasi rata-rata 69% untuk akurasi terendah adalah 47% dan tertinggi adalah 100%. Sedangkan untuk mobil akurasi rata-rata adalah 72% untuk akurasi terendah adalah 28% dan untuk akurasi tertinggi adalah 97%.

4.3 Pengujian Deteksi Lampu

Pada pengujian ini dimasukan *file* lampu dan pengecekan pada mode automasi guna melakukan pemilihan titik *pixel* posisi dimana lampu merah dan lampu hijau berada. Karena menggunakan animasi maka data yang dipakai menghasilkan data 100% sesuai dengan yang diinginkan. Kemungkinan akurasi turun akan terjadi apabila terdapat faktor luar seperti lampu mati, lampu tertutup, hujan deras sehingga warna terhalang atau lampu tertutup oleh body mobil.



Gambar 4.4, Hasil pengujian deteksi lampu

4.4 Pengujian Pelanggaran Lampu

Pada pengujian ini akan dilakukan pada mode merah saja dimana kendaraan yang melewati garis akan dianggap sebagai pelanggaran.



Gambar 4.5, Penggunaan garis pelanggaran pada saat lampu merah

Tabel 4.6: Hasil pengujian pelanggaran lampu

No	Data	Durasi	Real Pelanggaran	Program Pelanggaran	Akurasi
1	MerahPagi1	02:51	2	2	100%
2	MerahPagi2	02:33	4	1	25%
3	MerahPagi3	01:46	6	5	83%
4	MerahPagi4	01:26	3	15	20%
5	MerahPagi5	02:03	9	33	28%
6	MerahSiang1	00:56	8	8	100%
7	MerahSiang2	02:31	9	9	100%
8	MerahSiang3	00:41	8	35	25%
9	MerahSiang4	01:15	11	29	37%
10	MerahSiang5	01:15	16	34	47%
RATA-RATA					57%

Pada tabel 4.6 data rata-rata yang didapatkan dari 10 data adalah 57% dengan nilai paling tinggi 100% dan paling rendah 20%. Kebanyakan kesalahan terjadi karena “double-check *error*” dimana suatu objek akan dihitung berkali-kali selama objek tersebut berada diatas ROI-garis pelanggaran.

4.5 Pengujian Pelanggaran Marka

Pada pengujian ini akan dilakukan pada mode hijau saja dimana kendaraan yang melewati garis akan dianggap sebagai pelanggaran.



Gambar 4.6, Penggunaan garis pelanggaran pada saat lampu hijau

Tabel 4.7: Hasil pengujian pelanggaran marka

No	Data	Durasi	Real Pelanggaran	Program Pelanggaran	%
1	HijauPagi1			30	
2	HijauPagi2			13	
3	HijauPagi3			18	
4	HijauPagi4			6	
5	HijauPagi5			5	
6	HijauSiang1				
7	HijauSiang2			3	
8	HijauSiang3				
9	HijauSiang4				
10	HijauSiang5				
RATA-RATA					

4.6 Pengujian *Realtime*



Gambar 4.7, *Input video* dari berbagai waktu

Tabel 4.8: Hasil Pengujian jumlah kendaraan

No	Data	Durasi	Real Pelanggar	Program Pelanggar	%
1	Pagi				
2	Siang				
RATA-RATA					

Pada tabel 4.8 data rata-rata yang didapatkan dari 10 data adalah 57% dengan nilai paling tinggi 100% dan paling rendah 20%. Kebanyakan kesalahan terjadi karena “*double-check error*” dimana suatu objek akan dihitung berkali-kali selama objek tersebut berada diatas ROI-garis pelanggaran.

4.7 Pengujian Report

Pada pengujian ini dilakukan penambahan *file* baru untuk *folder* yang digunakan untuk *upload* otomatis ke Facebook. Pengujian ini berjalan 100% sesuai dengan perencanaan, sehingga ketika ada *file* baru maka akan dilakukan post baru ke Facebook pages yang ditunjuk.



Gambar 4.8, *Input video* dari berbagai waktu

4.8 Jenis Error

Masih ada beberapa kesalahan yang belum bisa ditangani oleh program, sedangkan untuk jenisnya pun bermacam-macam berikut adalah penjelasan dan mengapa bisa terjadi:

1. Kamera bergetar



Gambar 4.9, Jenis Error: Kamera bergetar

Pada *error* ini penyebabnya adalah kendaraan besar yang lewat dan menyebabkan pergerakan pada tiang tempat kamera terpasang, sehingga proses image subtraction terganggu dan menyebabkan banyak blob yang terdeteksi.

2. Cahaya memantul



Gambar 4.10, Jenis Error: Cahaya memantul

Pada *error* ini penyebabnya adalah kendaraan yang memiliki lampu yang jauh dan terang, mobil dan kendaraan besar yang bisa menimbulkan *error*. Deteksi terganggu karena cahaya dari mobil atau kendaraan besar tersebut dapat dianggap sebagai kendaraan.

3. Penyebrang jalan



Gambar 4.11, Jenis *Error*: Penyebrang jalan

Pada *error* ini penyebabnya adalah pejalan kaki yang berada tepat pada ROI-garis pelanggaran khususnya lampu merah sehingga proses pendeteksian terganggu.

4. Double check (ROI-Garis selalu berada dibagian tengah kendaraan)



Gambar 4.12, Jenis *Error*: Penyebrang jalan

Pada *error* ini penyebabnya adalah pelanggar lampu merah yang melaju diatas ROI-Garis sehingga akan selalu melakukan counter, meskipun sudah ditanggulangi dengan pemberian flag pada blob kendaraan tersebut, jika terdapat blob baru pada objek kendaraan maka akan melakukan counter secara terus menerus.

HALAMAN SENGAJA DIKOSONGKAN

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisa yang telah dilakukan maka dapat diambil kesimpulan sebagai berikut:

1. Proses pengklasifikasi kendaraan dapat dilakukan meskipun untuk malam hari(78%) dan siang hari(78%) memiliki tingkat persentase akurasi lebih kecil dibandingkan dengan pagi hari(93%).
2. *Tracking* yang digunakan pada tahap pengklasifikasi kendaraan digunakan hanya sebagai anti-*collision* hanya dapat mencakup selama 5 *frame*, oleh karena itu dapat terjadi penggabungan kendaraan satu-dengan yang lain ketika *collision* terjadi secara terus dan berurutan tiap *frame*.
3. Penggunaan *corner detection* masih belum mencakup semua kendaraan yang ada di jalan. Oleh karena itu dibutuhkan tambahan metode untuk menambah *feature* untuk meningkatkan akurasi

5.2 Hal yang Dilakukan Selanjutnya

Selanjutnya, penulis memiliki perencanaan mengenai hal-hal yang akan dilakukan selanjutnya untuk menyelesaikan penelitian dan proyek akhir. Berikut perencanaan kedepannya:

1. Meningkatkan kualitas dari *segmentasi* dengan mengeliminasi *shadow* ketika siang hari atau mengeliminasi cahaya lampu pada malam hari
2. Melakukan pengujian berdasarkan salah satu jenis kendaraannya baik pada klasifikasi ataupun pelanggaran
3. Melakukan pencarian nilai *threshold* dan *contour* pada *violation detection* saat lampu merah
4. Melakukan *tracking feature* menggunakan metode *optical flow* pada *violation* saat lampu hijau
5. Memberikan *acceptance condition* pada *violation* saat lampu merah dan lampu hijau

HALAMAN SENGAJA DIKOSONGKAN

DAFTAR PUSTAKA



- [1] Tim Peneliti Puslitbang Kumdil dan PSHK, Alternatif Pengelolaan Perkara Pelanggaran Lalu Lintas di Pengadilan, Laporan Penelitian, Jakarta, Puslitbang Kumdil dan PSHK, 2014, Hlm.4
- [2] Sebagaimana disampaikan oleh Yahya Syam Ketua Pengadilan Negeri Jakarta Timur dalam pada seminar penelitian “*Alternatif Pengelolaan Perkara Lalu Lintas di Pengadilan*”. Acara yang merupakan rangkaian kegiatan Penelitian tentang Alternatif Pengelolaan Perkara Tilang di Pengadilan ini diselenggarakan di Jakarta, 17 Juni 2014
- [3] Shunsuke Kamijo, YasuyukiMatshushita, Katsushi Ikeuchi, Fellow, IEEE, and Masao Sakauchi. 2000.*Traffic Monitoring and Accident Detection at Intersection*. Eindhoven Univ. of Technol
- [4] en.wikipedia.org/wiki/Optical_flow
- [5] Gibson, J.J., *The Perception of the Visual World* (Riverside Press, Cambridge, 1950).
- [6] Gibson, J.J., *The Senses Considered as Perceptual Sistem*s (Houghton-Mifflin, Boston, MA.1966).
- [7] Gibson, J.J.On the analysis of change in the optic array, *Scandinavian J. PsychoL* 18 (1977)161-163.
- [8] Janssen, Cory. Internet of Things: IoT. Diakses dari situs techopedia pada 9 November 2013
- [9] Hario Baskoro Basoeki, DKK.2013.‘*BTRAFFWATCH*’ *SOLUSI UNTUK PIHAK KEPOLISIAN DALAM MELAKUKAN PENGAWASAN DAN PENDETEKSIAN PELANGGARAN PADA LAMPU LALU LINTAS*.Politeknik Elektronika Negeri Surabaya
- [10] H. Asaidim, DKK .2014.*Shadow elimination and vehicles classification approaches in traffic video surveillance context*.Nador, Morocco
- [11] Elie Nasr, Elie Kfoury, David Khoury.2016.*An IoT Approach to Vehicle Accident Detection, Reporting, and Navigation*.American University of Science and Technology




HALAMAN SENGAJA DIKOSONGKAN

LAMPIRAN

- Pengujian Jumlah Kendaraan**




No	Nama	Bukti Frame Terakhir
1	Malam-1	
2	Malam-2	
3	Malam-3	



No	Nama	Bukti Frame Terakhir
4	Pagi-1	
5	Pagi-2	
6	Pagi-3	

No	Nama	Bukti Frame Terakhir
7	Siang-1	
8	Siang-2	
9	Siang-3	




- **Pengujian Klasifikasi Kendaraan**




No	Nama	Bukti Frame Terakhir
1	Malam-1	
2	Malam-2	
3	Malam-3	


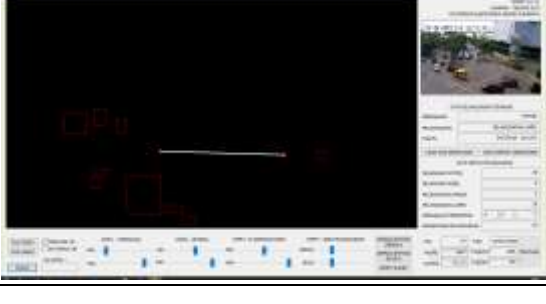
No	Nama	Bukti Frame Terakhir
4	Pagi-1	
5	Pagi-2	
6	Pagi-3	

No	Nama	Bukti Frame Terakhir
7	Siang-1	
8	Siang-2	
9	Siang-3	

- **Pelanggaran Lampu Merah**



No	Nama	Bukti Frame Terakhir
1	MerahPagi1	
2	MerahPagi2	
3	MerahPagi3	




No	Nama	Bukti Frame Terakhir
4	MerahPagi4	
5	MerahPagi5	
6	MerahSiang1	


No	Nama	Bukti Frame Terakhir
7	MerahSiang2	
8	MerahSiang3	
9	MerahSiang4	

No	Nama	Bukti Frame Terakhir
10	MerahSiang5	

- Pelanggaran Lampu Hijau**

No	Nama	Bukti Frame Terakhir
1	HijauPagi1	
2	HijauPagi2	

No	Nama	Bukti Frame Terakhir
3	HijauPagi3	
4	HijauPagi4	
5	HijauPagi5	

No	Nama	Bukti Frame Terakhir
6	HijauSiang1	
7	HijauSiang2	
8	HijauSiang3	
9	HijauSiang4	

No	Nama	Bukti Frame Terakhir
10	HijauSiang5	

HALAMAN SENGAJA DIKOSONGKAN

BIDODATA PENULIS

Nama : Achmad Rahman Mawardi
TTL : Lamongan, 17 Oktober 1996
Alamat : Lamongan
No. Hp : 0818-0379-3442
Email : achmadrahmanm@gmail.com
Web : -

Pendidikan :

Jenjang	Institusi	Tahun Masuk	Tahun Lulus	Jurusan
SD	SDN Tambakrigadung 2	2002	2008	-
SMP	SMPN 2 Lamongan	2008	2011	-
SMK	SMKN 1 Lamongan	2011	2014	Multimedia