SISTEM DETEKSI PELANGGARAN DI PERSIMPANGAN LALU LINTAS PADA MOBIL DENGAN OPENCV MENGGUNAKAN RASPBERRY PI

CAR VIOLATION DETECTION SYSTEM AT TRAFFIC LIGHT WITH OPENCV USING RASPBERRY PI

Iwan Iwut Tritoasmoro,M.T[1], Ir. Rita Magdalena ,M.T[2] , Timothy Kawulusan[3]

[1,2,3]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
[1]**iwaniwut@telkomuniversity.ac.id** [2]**ritamagdalena@telkomuniversity.ac.id**
[3]**timothykawulusan@students.telkomuniversity.ac.id**

Abstrak

Dunia sekarang menuju pada sebuah perubahan yang sangat signifikan, diantaranya yaitu revolusi industri 4.0 di mana setiap alat atau perangkat keras akan terkoneksi dengan internet dan atau menjalankan sesuatu sistem secara otomatis maka dari itu kita memerlukan alat atau perangkat yang akan membantu pada sistem terpadu nantinya dan salah satu contoh untuk membantu sistem terpadu pada pemerintahan yaitu sistem deteksi pelanggaran lalu lintas khusus pada kendaraan mobil. Lebih dari itu alat ini diharapkan akan membantu mengurangi tingkat pelanggaran lalu lintas pada saat ini.

Penggunaan *image processing* dalam mengambil sebuah informasi sudah sering digunakan pada saat ini, terlebih khusus dalam hal automasi. Dengan menggunakan *image processing* penulis merancangan sebuah sistem untuk mendeteksi pelanggaran lalu lintas pada persimpangan jalan untuk kendaraan mobil menggunakan sistem arsitektur yang minimum seperti Raspberry Pi.

Kata kunci : revolusi industry 4.0, image processing, automasi, smart city, Raspberry Pi, pararel processing

Abstract

World nowadays is heading toward significantly, including the industry revolution 4.0 where every device or hardware will be connected to the internet and or running some system automatically, therefore we need tools or devices that will help us to integrated the government system later such as the system of detection for traffic violations especially for cars. Also, with this tool author will be expected to help reduce the level of traffic violations at this time.

Using image processing for retrieving an information nowdays already often used, especially for automation. By using image processing, author designed a system to detect traffic violations at crossroads for automobiles using minimum architectural systems such as the Raspberry Pi.

Keywords: ***industrial revolution 4.0, image processing, automation, smart cities, Raspberry Pi, pararel processing.***

1. Pendahuluan

Kecelakaan lalu lintas di Indonesia termasuk tertinggi di dunia [1], menurut Kepala Kepolisian Indonesia Jenderal Tito Karnavian mengatakan tingkat kecelakaan lalu lintas di Indonesia masih tinggi di antara negara-negara ASEAN [2] dan Dinas Perhubungan (DISHUB) juga mencatat serupa yaitu pada tahun 2016 saja kecelakaan lalu lintas terjadi sebanyak 25.859 kali [3].

Masalah yang terjadi pada sekarang ini, sulitnya para pihak instansi pemerintah untuk menegakkan keadilan untuk meningkatkan rasa jera pada pelanggar dengan cara mendata para pelanggar lalu lintas, dengan cara ini bias menjadi salah satu cara untuk mencegah terjadinya kecelakaan - kecelakaan lalu lintas.

Dengan kemajuan industry zaman ini dan sekarang sedang menuju ke zaman *Industry* 4.0 di mana Automasi dan Internet akan digabung menjadi satu atau sering kita sebut dengan *Internet of Things* (IoT) [5]. Sistem ini sangat cocok karena menggunakan alat yang mendukung IoT, dengan begitu sistem ini sangat baik untuk diimplementasikan, dikembangkan, serta membantu mempercepat kemajuan industry 4.0 di Indonesia

Hasil dari tugas akhir ini didapatkan bahwa sistem ini bisa berkerja dengan baik dengan hasil akurasi deteksi pelanggaran sebesar 100% dan hasil kerberhasilan keputusan jenis pelanggaran sebesar 92% menggunakan SSIM dan juga menemukan bahwa sistem layak bekerja pada sistem yang minimum dan menggunakan teknik *multiprocessing* dimana meningkatkan performa sistem dan mempercepat waktu komputasi.

2. Dasar Teori

2.1 Citra analog ke difital

Pengelolaan citra digital tidak terlepas dengan namanya kamera digital. Kamera adalah perangkat keras yang mempunyai sensor dan kita tahu fungsinya untuk mengubah objek yang kita lihat menjadi sebuah objek dua dimensi yang sering kita sebut gambar, sehingga kamera memerlukan alat – alat khusus dan *analog to digital converter* untuk mengubah yang sebelumnya merupakan intesitas cahaya menjadi nilai biner.

2.1.1 Sensor Cahaya

Cahaya yang kita ketahui ternyata merupakan gelombang Elektromagnetik di mana membawa elektron, sehingga ketika gelombang cahaya datang elektron dari gelombang cahaya tersebut datang dan memantul pada benda lain, hasil dari pantulan itu kita tangkap dengan sebuah sensor di mana sensor dibuat untuk mengubah intesitas cahaya menjadi besaran listrik.

2.1.2 Analog to digital Converter

Setelah cahaya masuk ke sensor, nilai dari sensor akan diubah menjadi sinyal digital menggunakan *Analog to Digital Converter*(ADC) dengan begitu nilai yang awalnya berupa nilai fisis menjadi nilai biner, pengubahan nilai ini perlu karena sinyal dari sensor photodioda bersifat analog dan komputer tidak bisa mengenal banyak nilai secara sekaligus. Perbedaan cahaya – cahaya tersebut akan disusun secara array dengan model dua dimensi, sehingga gambar digital merupakan bentuk dua dimensi [7]

2.1.3 Warna pada gambar digital dengan penggunaan *Bayer Filter*

Warna pada gambar digital adalah hasil manipulasi dari sebuah alat yang bernama *Bayer Filter*[1], seperti yang penulis jelaskan sebelumnya hal ini dikarenakan sensor menangkap intesitas cahaya, bukan warna cahaya. Maka dari itu kita perlu membuat alat dari susunan *filter* warna atau disebut *Bayer Filter*.

2.2 Pengelolaan citra digital

Pengelolaan citra gambar digital adalah Teknik atau cara untuk memanipulasi citra gambar digital dengan bantuan komputer dengan tujuan untuk mengambil informasi dari gambar, *pattern*, analisis pada kompresi gambar, dan memperbaiki gambar.

2.2.1 *Gray level image processing*

Pengelolaan citra gambar digital dengan menggunakan *Gray level image processing* sering digunakan dan menjadi dasar dari pengelolaan citra lain, dengan menggunakan *Gray level image processing* bisa mengetahui apakah gambar terlalu terang atau gelap, dengan begitu memudahkan kita untuk ke proses selanjutnya.

2.2.2 *Binary level image processing*

Gambar biner adalah gambar yang dibentuk oleh pixel yang yang terdiri oleh angka biner yaitu 1 atau 0. Penggunaan gambar biner ditunjukan agar komputer bisa mendapatkan informasi yang tepat pada sebuh objek yang akan kita tentukan, dikarenakan jika memproses gambar bewarna komputer sulit untuk membedakan antar objek yang mau kita proses.

2.2.3 *Structural Similarity Image* (SSIM)

SSIM adalah metode dimana untuk mengukur persamaaan dari kedua gambar, SSIM biasanya digunakan untuk mengukur perbedaan kualitas citra pada pertelevisian atau perfiliman, akan tetapi penulis akan menggunakan teknik ini sebagai pembanding citra sebelum dan sesudah untuk mengetahui adanya pelanggar atau tidak dan juga sebagai pendefinisi apakah objek dalam hal ini mobil tersebut melanggar dengan menerobos lampu lalu lintas atau berhenti pada marka jalan saja.

2.2.3 *Canny Edge Detection*

Edge Detection atau yang dikenal dengan deteksi tepi adalah operasi yang dijalankan untuk mendeteksi garis tepi atau *edges* yang membatasi dua wilayah citra homogen yang memiliki tingkat kecerahan yang berbeda (Gambar 2.4). Tujuan awal dari adanya *Edge Detection* ini untuk membuat citra dua dimensi berbentuk kurva sehingga ketika sebuah

citra telah berhasil dibuah melalui *Edge Detection* maka akan menghasilkan kurva dan ketika di tunjukkan akan berbentu garis putih.

Penulis menggunakan *Edge Detection* guna untuk membuat *Region of Interest* (ROI) atau daerah gambar yang dimana hanya marka jalan *zebra-cross* berwarna putih, diperlukan ROI agar sistem hanya akan fokus mengamati yang terjadi pada daerah tersebut dan membuat waktu pengelolaan gambar lebih cepat. Ketika citra diambil, citra tersebut diubah dalam bentuk *edges* dan membentuk garis yang hanya pada daerah marka jalan, setelah itu dikarenakan *edges* berbentuk dalam kurva penulis mengambil nilai koordinat terendah dan tertinggi dan memotong citra awal dengan nilai koordinat tadi agar citra akhir menunjukkan ROI yang spesifik yaitu marka jalan tersebut.

2.3 Hardware dan Software

2.3.1 OpenCV

OpenCV adalah library computer vision yang *open-source* di mana ditulis dalam Bahasa C dan C++ dan bisa dijalankan pada Linux, Mac, dan atau Windows diaman OpenCV di desain untuk memudahkan komputasi pengelolaan citra gambar pada *real-time*.

2.3.2 Python

Python adalah Bahasa pemograman yang mudah dimengerti dikarenakan syntax yang mudah dibaca dan dipahami sehingga Python sangat baik untuk dipakai pada proyek yang besar dan berlangsung lama agar pada pengembang kode yang akan mendatang nanti bisa mudah memahami isi program.

Pada pemograman perlunya optimasi pada program yang ditulis, fungsi dari optimasi tersebut yaitu untuk meningkatkan kecepatan proses pada saat program berjalan, program mudah untuk diubah dalam skala besar, dan mudah untuk dipahami, terlebih khusus pada sistem yang dibuat penulis dijalankan pada sistem yang memiliki spesifikasi yang minimum.

2.3.2 Kamera dan Raspberry Pi

Kamera digital adalah alat yang berfungsi untuk memproyeksikan tampilan atau gambar dari objek sesuatu yang dibiaskan melalui lensa dan ditangkap oleh sensor, kemudian menyimpannya ke dalam media digital.

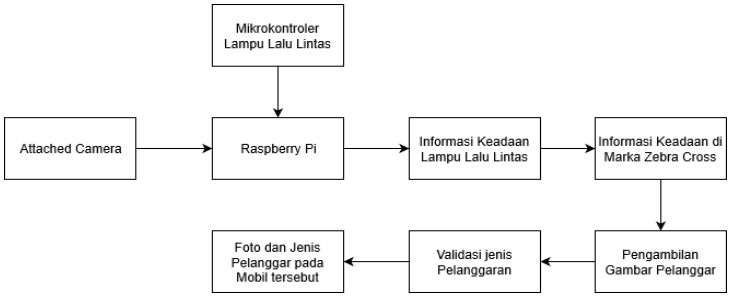
Kamera sudah memiliki alat – alat yang penting untuk mengubah sinyal cahaya menjadi sinyal digital seperti sensor CMOS, *Analog to Digital Converter*, *Bayer Filter*, dan penyimpanan media digital di mana bisa membuat Kamera digital ini berfungsi selayaknya dan menyimpannya dalam bentuk digital.

Raspberry Pi merupakan komputer sehingga memiliki kemampuan perhitungan selayaknya komputer pada umumnya yang mempunyai *Processor*, RAM, ROM, GPU, I/O. Sehingga digunakan untuk mengelola semua proses yang ada termasuk mengambil gambar pada saat pengecekan pelanggaran atau mendefinisikan jenis pelanggaran yang dilaakukan pelanggar.

3. Perancangan Sistem

3.1. Desain Sistem

Desain sistem yang akan dijabarkan untuk perancangan dan implementasi sistem deteksi pelanggaran di persimpangan lalu lintas pada mobil dengan OpenCV menggunakan Raspberry Pi menggunakan *prototype* alat yang menyerupai persimpangan lampu lalu lintas, *prototype* alat diharapkan bisa mensimulasikan keadaan di persimpangan lalu lintas dengan pemodelan sistem sesuai dengan berikut :



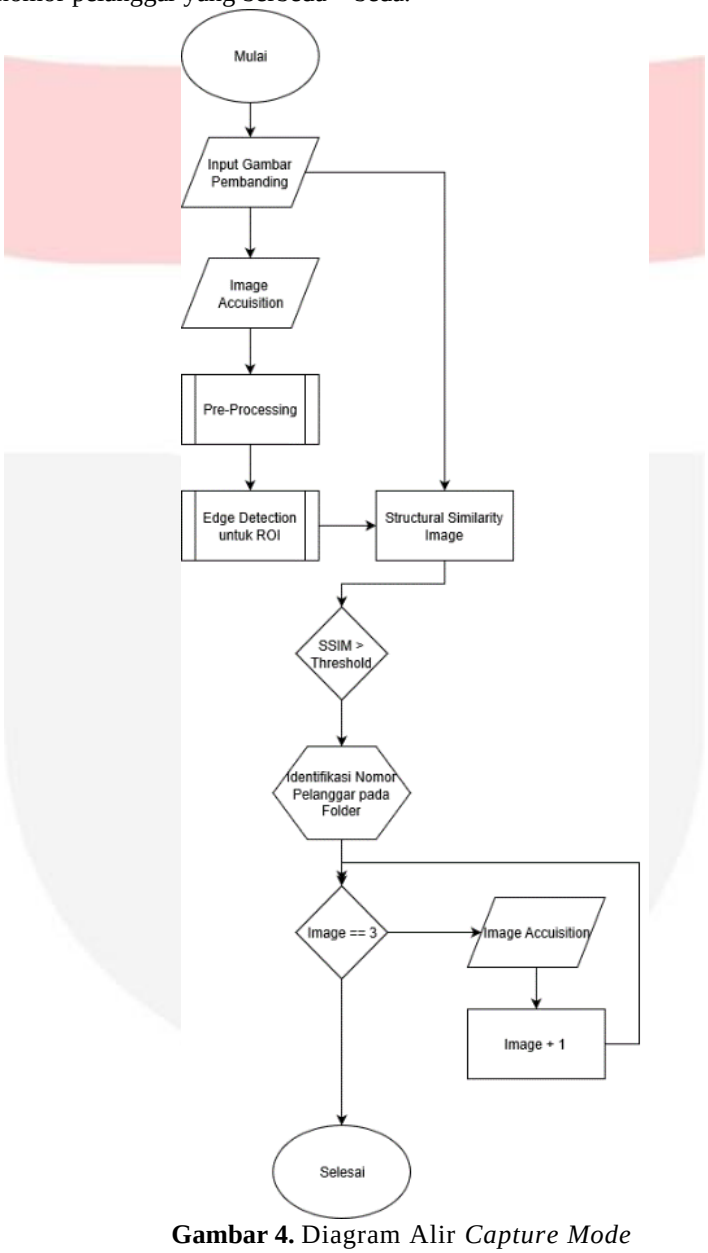
Gambar 3. Diagram Blok Sistem

3.2. Desain Alir Pemograman

3.2.1 Desain Alir *Capture Mode*

Pada *Capture Mode* seperti pada Gambar 4, Raspberry Pi akan membuka kamera terlebih dahulu kemudian akan mengambil gambar tiap frame dan memeriksa apakah adanya perbedaan antara gambar sebelum adanya objek diatas marka *zebra cross* atau tidakk, jika terjadi perbedaan maka kita membuat iterasi baru dan kemudian membuat folder baru, ketika folder baru telah berhasil dibuat diambil tiga gambar sekaligus.

Hal ini akan berlangsung terus – menerus jika keadaan lampu lalu lintas bewarna merah, jika ditemukan pelanggar baru maka akan membuat iterasi baru sehingga tiap pelanggar mempunyai nomor pelanggar yang berbeda – beda.



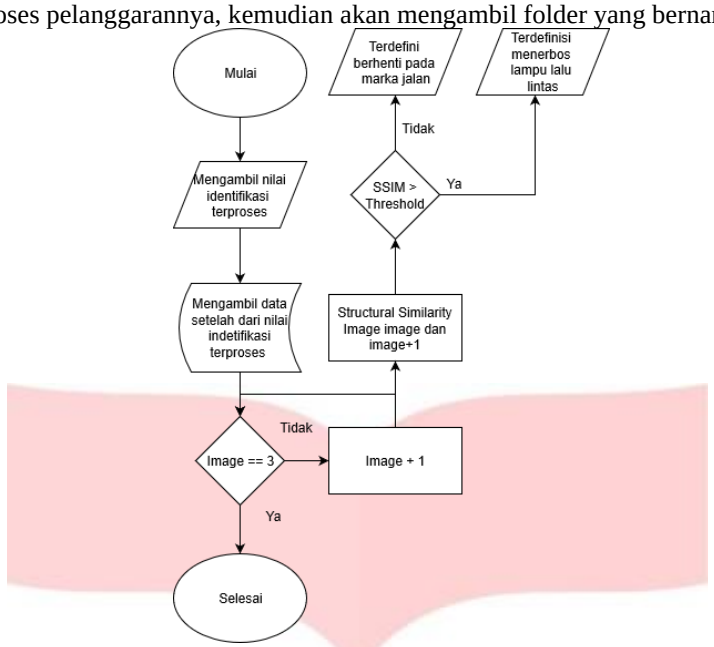
Gambar 4. Diagram Alir *Capture Mode*

3.2.2. Diagram alir *Processing Mode*

Diagram Alir *Processing Mode* seperti pada Gambar 5 terjadi pada saat lampu lalu lintas berwarna hijau, ini dilakukan agar Raspberry Pi tidak terbebani saat mengambil gambar dan memeriksa pelanggaran yang terjadi sehingga hal yang krusial bisa mempunyai tingkat deteksi yang lumayan tinggi.

Hal pertama, Raspberry Pi akan mengambil nilai terakhir dari jumlah pelanggar yang

telah di proses pelanggarannya, kemudian akan mengambil folder yang bernamakan nilai



Gambar 5. Diagram Alir *Processing Mode*.

tersebut. Raspberry Pi akan melakukan perulangan dimana membaca semua citra didalam folder dan diisikan kedalam sebuah array baru.

Hal kedua, Raspberry Pi akan melakukan perulangan untuk membaca citra dalam array dan kemudian membandingkan dengan *Structural Similarity Image*(SSIM) citra pertama dan citra selanjutnya sampai citra selanjutnya habis. Jika SSIM melebih batas *threshold* maka akan mengembalikan nilai *False* dan sebaliknya nilai *True*, nilai tersebut dimasukkan kedalam sebuah array data lagi. Hasil dalam array data akan dilakukan *logic or*, jika salah satu bernilai *True* maka bisa diambil kesimpulan pelanggar tersebut melakukan pelanggaran menerobos lampu lalu lintas pada saat lampu bewarna merah, dan jika *False* berarti pelanggar tersebut melakukan pelanggaran lalu lintas berhanti di marka *zebra cross* pada saat lampu berwarna merah.

4. Hasil dan Analisis

4.1 Pencarian Nilai ambang batas(threshold) SSIM

Pencarian nilai ambang batas Structural Similarity Image menggunakan statistika dasar dalam mencari nilai rataan, nilai terkecil, dan terbesar. Pencarian nilai SSIM

Tabel 1 Nilai ambang batas(threshold) SSIM

Pengujian Ke -	Nilai SSIM saat citra sama	Nilai SSIM saat citra beda
1	0,971	0,589
2	0,837	0,588
3	0,837	0,59
4	0,9819	0,578
5	0,97	0,297
6	0,98	0,264
7	0,83	0,61
8	0,937	0,603
9	0,981	0,548
10	0,971	0,61
11	0,972	0,598
12	0,945	0,71
13	0,873	0,49
14	0,837	0,74
15	0,981	0,741
16	0,838	0,702
17	0,9705	0,501
18	0,937	0,286
19	0,982	0,696
20	0,972	0,681
21	0,982	0,775
22	0,973	0,709
23	0,98	0,47
24	0,972	0,284
25	0,961	0,301
Rata - Rata	0,938856	0,55844
Nilai Tertinggi	0,982	0,775
Nilai Terendah	0,83	0,264
SSIM Threshold	0,8	

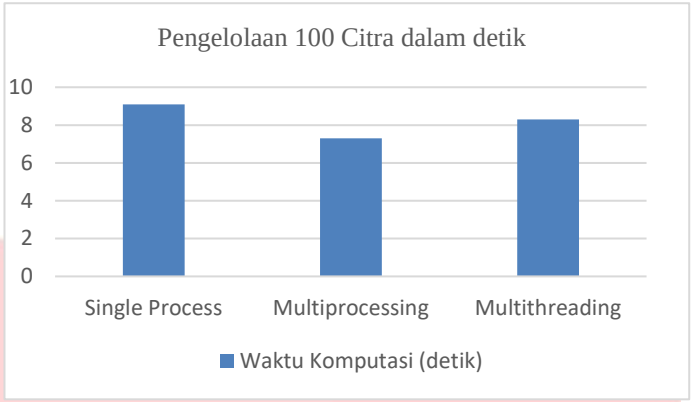
dilakukan membandingkan dua buah citra yang sama dan beda dimana pada saat lampu merah yang sama menunjukkan ada tidak ada pelanggaran dan yang sama sebaliknya, kemudian pada saat lampu hijau yang citra sama menunjukkan bahwa melakukan pelanggaran berhenti di marka jalan dan citra beda menerbos. Diambil 25 data dan menemukan seperti pada Tabel 1

Ketika telah diuji, SSIM pada citra yang sama mendapatkan nilai tertinggi sebesar 0,982 dan terendah sebesar 0,83 sedangkan pada saat ada objek mendapatkan nilai tertinggi sebesar 0,775 dan terendah mendapatkan 0,47. Dikarenakan nilai terendah pada saat tidak ada objek adalah 0,83 dan nilai tertinggi pada saat ada objek adalah 0,775 maka sebaiknya mengambil nilai terendah pada saat tidak ada objek yaitu 0,83 dan dibulatkan menjadi 0,8.

4.2 Pengujian kecepatan proses dan optimasi program

Pengujian kedua dilakukan untuk mencari tahu teknik mana agar mempercepat pengelolaan citra pada saat pendefinisian pelanggaran lalu lintas. Ini sangat diperlukan dikarenakan dibutuhkan agar sistem tidak terlalu banyak menunda pekerjaan pendefinisian pada saat lampu lalu lintas berwarna hijau jika tanpa menggunakan pemograman pararel, lebih dari itu kita tahu bahwa saat lampu berwarna hijau tidak memakan waktu yang lama untuk berpindah ke merah lagi apalagi pada Indonesia.

Pengujian dilakukan dengan cara mencoba 100 citra untuk di pre-processing citra pada sistem ini seperti di grayscale, binarization, reduce noise, dan edge detection. Hasil yang diamati yaitu Teknik pemograman mana yang berhasil menyelesaikan *pre-processing* 100 gambar dengan waktu tercepat.



Gambar 6 Hasil Pengujian pararel program

4.3 Pengujian dan Analisa tingkat keberhasilan deteksi pelanggar.

Tingkat keberhasilan mendeteksi pelanggar sangat penting karena tujuan awal dan pemrosesan awal dari sistem ini yaitu pada saat mendeteksi pelanggar – pelanggar, untuk itu penulis pertama mencari tahu kecepatan pemrosesan pengambilan gambar pada saat lampu merah, dengan menggunakan pengukuran kinerja kecepatan pengambilan dan pengelolaan gambar dalam 10 detik lampu lalu lintas berwarna merah mendapatkan sekitar 198 citra yang berhasil di *pre-processing* dan di compare dengan SSIM sehingga mendapatkan pemrosesan dalam citra per detik sebesar :

$$\text{Citra per detik} = \frac{198}{10 \text{ detik}} = 19{,}8 \approx 20$$

Kemudian penulis mengujicobakan pada sistem secara terus menerus untuk mendapatkan seberapa akurat yang bisa dideteksi oleh sistem ini, penulis melakukan 10 kali percobaan pada saat lampu lalu lintas dengan waktu yang lampu merah yang berbeda – beda dan mendapatkan data berikut:

Tabel 2 Hasil Pengujian deteksi pelanggar

Percobaan ke -	Durasi Lampu	Mobil Lewat	Terdeteksi
1	30	8	8
2	30	9	9
3	10	3	3
4	10	4	4
5	15	6	6
6	20	7	7
7	25	5	5
8	30	10	10
9	20	6	6
10	5	2	2
	Jumlah	60	60
Keberhasilan Deteksi			100

4.4 Pengujian dan Analisa keberhasilan keputusan jenis pelanggaran.

Keberhasilan dalam memberikan jenis pelanggaran pada pelanggar sangat penting karena seperti yang telah dijelaskan sebelumnya bahwa setiap pelanggaran mempunyai undang –

undang yang berbeda, maka dari itu sistem harus mampu untuk bisa memberikan jenis pelanggaran dengan tepat. Penulis melaksanakan uji coba sebanyak 25 data untuk mengetahui seberapa besar keberhasilan memberikan jenis pelanggaran dengan tepat kepada pelaku dengan data sebagai berikut :

Tabel 3 Hasil Pengujian keberhasilan keputusan

Percobaan	SSIM > Threshold			Hasil	
	Gambar 1 dan 2	Gambar 2 dan 3	Gambar 3 dan 4	Dilihat dari pengamatan	Keputusan Sistem
1	FALSE	TRUE	TRUE	MENEROBOS	MENEROBOS
2	FALSE	FALSE	FALSE	MARKA JALAN	MARKA JALAN
3	TRUE	TRUE	TRUE	MENEROBOS	MENEROBOS
4	FALSE	FALSE	FALSE	MARKA JALAN	MARKA JALAN
5	TRUE	TRUE	TRUE	MENEROBOS	MENEROBOS
6	FALSE	FALSE	FALSE	MARKA JALAN	MARKA JALAN
7	TRUE	TRUE	FALSE	MENEROBOS	MENEROBOS
8	TRUE	TRUE	TRUE	MENEROBOS	MENEROBOS
9	FALSE	FALSE	FALSE	MARKA JALAN	MARKA JALAN
10	FALSE	FALSE	FALSE	MARKA JALAN	MARKA JALAN
11	TRUE	TRUE	TRUE	MENEROBOS	MENEROBOS
12	FALSE	FALSE	FALSE	MARKA JALAN	MARKA JALAN
13	FALSE	FALSE	FALSE	MARKA JALAN	MARKA JALAN
14	FALSE	FALSE	FALSE	MARKA JALAN	MARKA JALAN
15	TRUE	TRUE	FALSE	MARKA JALAN	MENEROBOS
16	TRUE	TRUE	TRUE	MENEROBOS	MENEROBOS
17	FALSE	FALSE	FALSE	MARKA JALAN	MARKA JALAN
18	TRUE	TRUE	TRUE	MENEROBOS	MENEROBOS
19	TRUE	TRUE	TRUE	MENEROBOS	MENEROBOS
20	FALSE	FALSE	FALSE	MARKA JALAN	MARKA JALAN
21	TRUE	TRUE	TRUE	MENEROBOS	MENEROBOS
22	TRUE	FALSE	FALSE	MENEROBOS	MENEROBOS
23	TRUE	TRUE	TRUE	MENEROBOS	MENEROBOS
24	TRUE	TRUE	FALSE	MARKA JALAN	MENEROBOS
25	TRUE	FALSE	FALSE	MENEROBOS	MENEROBOS
				Total Benar	**23**
				Keberhasilan Proses	**92%**

Pada tabel tersebut penulis menemukan bahwa tingkat keberhasilan sebesar 92%, *error* terjadi pada pengamatan ke 15 dan 24, kedua gambar tersebut menujukkan sebenarnya terjadi pelanggaran marka jalan tetapi sistem mendeteksi menerobos lampu lalu lintas, mobil pada kedua pelanggar tersebut mundur ketika berhenti di marka jalan.

5. Kesimpulan

Berdasarkan hasil pengamatan, pengujian, dan analisis yang telah digunakan pada sistem pendeteksi pelanggaran lalu lintas pada mobil ditemukan bahwa :

1. Deteksi pelanggaran lalu lintas menggunakan image processing bisa diterapkan pada dunia nyata dengan menggunakan sistem minimum dan tingkat deteksi yang lumayan baik.
2. Penggunaan Edge Detection untuk mendapatkan Region of Interest bisa dilakukan dan nyatanya bisa meningkatkan performa dalam mendeteksi awal para pelanggar lalu lintas.
3. Pengujian dengan menggunakan teknik pemograman berbeda agar program berjalan pararel bisa dilakukan, Teknik ini sangat baik untuk sistem minum yang dimana jika untuk melakukan proses yang hanya menggunakan *single-core* mendapatkan waktu komputasi yang kurang. Pengujian dilakukan pada saat pendefinisian jenis pelanggaran waktu sistem berada di posisi lampu berwarna hijau, seperti yang telah penulis tuliskan sebelumnya dimana pengelolaan harus cepat agar tidak menunda banyak pekerjaan dan juga seperti yang diketahui lampu warna hijau di Indonesia termasuk cepat. Hasil yang ditemukan bahwa penggunaan *multiprocessing* mendapatkan lebih cepat dalam waktu pengelolaan dan waktu komputasi daripada *single-process*.
4. Menggunakan metode SSIM untuk menentukan perbedaan gambar bisa digunakan dan mendapatkan hasil yang baik, semua gambar yang mau diproses oleh SSIM harus digrayscale terlebih dahulu baru kemudian diproses, hasil tingkat kerbehasilan deteksi

terhadap SSIM Threshold pada saat lampu lalu lintas berwarna merah mendapatkan hasil 100% pada siang hari dan 100% pada malam hari sehingga dapat disimpulkan sistem ini mencapai 100% dalam pendeteksian pelaggaran, kecepatan pengambilan serta pembandingan gambar sebelum dan sesudah menggunakan SSIM juga mendapatkan hasil sebesar 20 gambar per detik dimana dinilai sudah baik.

5. Hasil tingkat keberhasilan definisi pelanggar pada saat lampu hijau mendapatkan nilai ketepatan deteksi sebesar 92%.

Daftar Pustaka:

[1] "Celaka," Kompas, 12 Maret 2017. [Online]. Available: https://otomotif.kompas.com/read/2017/12/04/100400715/kematian-akibat-kecelakaan-di-indonesia-tertinggi-di-dunia. [Accessed 18 Februari 2019].

[2] "ASEAN," Tempo, 05 November 2017. [Online]. Available: https://nasional.tempo.co/read/1033993/angka-kecelakaan-lalu-lintas-indonesia-termasuk-tinggi-di-asean/full&view=ok. [Accessed 18 Februari 2019].

[3] "Dishub," Dishub, 29 Agustus 2018. [Online]. Available: https://dishub.acehprov.go.id/informasi/angka-kematian-akibat-kecelakaan-lalu-lintas-di-indonesia/. [Accessed 18 Februari 2019].

[4] "CCTV," Bandung, [Online]. Available: http://atcs-dishub.bandung.go.id/. [Accessed 18 Februari 2019].

[5] "IoT," Kompas, 03 Oktober 2018. [Online]. Available: https://edukasi.kompas.com/read/2018/10/03/17521731/milenial-siap-siap-sambut-revolusi-industri-40. [Accessed 18 Februari 2019].

[6] A. C. Bovik, "Types of Images," in *Handbook of Image and Video Processing*, USA, Elseveir Academic Press, 2005, pp. 4-5.

[7] A. C. Bovik, "Dimension of Images," in *Handbook of Image and Video Processing*, USA, Elseveir Academic Press, 2005, p. 5.

[8] G. Bradski and A. Kaehler, "Chapeter 1 : Overview," in *Learning OpenCV: Computer Vision with the OpenCV Library*, USA, O'Rielly Media Inc, 2008, p. 1.

[9] D. K. S. BRHANU M. GEBREGEORGIS, "SMART TRAFFIC LIGHT CONTROLLING AND VIOLATION DETECTION SYSTEM USING DIGITAL IMAGE PROCESSING," *International Journal of Engineering Research-Online* , vol. 4, no. 2, p. 522, 2016.