

System Pendeteksi Pelanggar *Traffic Light* pada *Zebra Cross* Menggunakan *Raspberry Pi* Berbasis Pengolahan Citra Digital

Rachmat Hidayat S.
Program Studi Teknik Elektro
Universitas Merdeka Malang
Kota Malang, Indonesia
Rachmath275@gmail.com

Abdur Rabi'
Program Studi Teknik Elektro
Universitas Merdeka Malang
Kota Malang, Indonesia
arrabik@gmail.com

Elta Sonalitha
Program Studi Teknik Elektro
Universitas Merdeka Malang
Kota Malang, Indonesia
elta.sonalitha@unmer.ac.id

Abstrak— Berdasarkan permasalahan terkait banyaknya pelanggaran pada *zebra cross*, penulis bertujuan membuat penelitian tentang *system* pendeteksi pelanggar *traffic light* pada *Zebra cross* menggunakan *Raspberry Pi* berbasis pengolahan citra digital dengan menggunakan *library open CV* dan metode *contour area* dan *background subtraction* sebagai pendeteksi bentuk *zebra cross* dan mendeteksi objek pelanggar. Dalam penelitian ini berhasil mendapatkan data pendeteksian pada klasifikasi objek kendaraan dengan luas objek kendaraan motor dengan luas objek 5000-9500 *pixel* dan untuk luas objek pada kendaraan mobil memiliki lebih dari 10000 *pixel*.

Kata kunci —*Pengolahan Citra Digital; Raspberry Pi; Open CV; Contour Area, Background Substraction*

I. PENDAHULUAN

Lalu lintas di kota sangat dipengaruhi oleh pengendali lampu lalu lintas. Saat menunggu lampu lalu lintas, pengemudi kehilangan waktu dan penggunaan bahan bakar. Oleh karena itu, tidak jarang para pengemudi kendaraan nekat untuk menerobos lampu merah, yang dalam hal ini dikarenakan beberapa pengemudi kendaraan tersebut banyak yang melewati batas marka jalan yaitu marka pada tempat penyebrangan jalan atau *zebra cross* yang seharusnya digunakan oleh pejalan kaki pada saat keadaan lampu sedang dalam keadaan merah [1].

Pengguna jalan dalam keadaan berlalu-lintas memiliki hak dalam ketertiban suatu keadaan berlalu-lintas yang berlangsung secara teratur sesuai dengan kewajiban setiap pengguna jalan. Mengenai (Pasal 1 angka 32 UU Nomor 22 Tahun 2009 Tentang LLAJ).

Masalah tentang ketertiban dalam berlalu-lintas di jalan merupakan tanggung jawab bersama, bukan hanya dari pihak kepolisian. Tetapi setiap pengguna jalan wajib mentaati peraturan dalam berlalu lintas yang berlaku.

Berdasarkan peraturan hukum yang mengatur mengenai tempat penyebrangan pejalan kaki/*zebra cross* adalah tertera di dalam Undang-Undang No. 22 Tahun 2009 tentang LLAJ. peraturan tentang pejalan kaki tertera didalam undang undang menyebutkan bahwa "Pejalan kaki berhak mendapatkan prioritas pada saat menyebrang jalan di tempat penyebrangan". Tentu dengan adanya tempat

penyebrangan *zebra cross*, para pengendara kendaraan seperti mobil, sepeda motor dapat lebih memperhatikan dalam berlalu lintas, serta lebih mengutamakan keselamatan bagi pejalan kaki yang akan melewati *zebra cross*[2].

Berdasarkan *TMC* atau *Traffic management center* memiliki data yang diperoleh POLRI yaitu data evaluasi dan analisa serta pelanggaran lalu-lintas selama operasi *zebra cross* tahun 2018 yang berada pada kota malang tanggal 30 oktober sampai dengan tanggal 12 November pada tahun 2018 secara keseluruhan jumlah 4.175 pengendara yang melakukan pelanggaran lalulintas.

Berdasarkan permasalahan tersebut, salah satu hal dapat dimanfaatkan untuk mengurangi jumlah pelanggaran pada *zebra cross* salah satunya yaitu dengan melakukan penerapan pendeteksian pelanggaran dengan tampilan yang akan terintegrasi dengan pengaturan lalu-lintas atau *Traffic management center* [3]. yaitu menggunakan deteksi pelanggaran secara *visual*, membutuhkan *system software* yang saling terintegrasi dengan baik sehingga pada saat proses deteksi pada pelanggaran tersebut dapat mendapatkan hasil yang optimal, serta tampilan dari proses pendeteksi pelanggaran dengan menerapkan sebuah *imageprocessing*[4].

Dengan munculnya permasalahan tersebut, maka penulis membuat penelitian yang berjudul "system pendeteksi pelanggar *zebra cross* pada *traffic light* menggunakan *raspberry pi* berbasis pengolahan citra digital" pada perancangan penelitian yang dilakukan terkait dengan implementasi *library open cv* serta penerapan pengolahan citra digital pada system yang dibuat serta dapat menangkap objek pengendara yang melanggar *zebra cross* pada *traffic lihgt* serta menggunakan *raspberry pi* sebagai *mikro komputer*, konsep yang digunakan adalah *input video/gambar* dengan menggunakan kamera yang mendeteksi posisi *zebra cross* kemudian mendeteksi adanya pelanggaran setelah itu sistem akan otomatis menangkap gambar pelanggar[5].

II. METODOLOGI

Metode pada penelitian ini menjelaskan mengenai proses perancangan dan variabel yang akan diteliti sehingga memudahkan pada pengerjaan tugas akhir ini.

A. Variabel penelitian

Perancangan *system* secara umum merupakan tahap awal sebagai acuan dalam perancangan sistem yang akan dibuat. Pada bagian ini akan dibahas mengenai garis besar dari masing-masing bagian dari sub-program serta apa proses yang akan terjadi dari tiap-tiap bagiannya[6]. Perancangan *program* terbagi menjadi beberapa bagian yang terdiri dari inisialisasi masukan *video*, *contour area*, *background subtraction*. Variabel penelitian meliputi :

- Fungsionalitas perangkat

Fungsionalitas perangkat sangat penting untuk diuji guna mengetahui apakah semua fungsi dari sebuah perangkat elektronik kamera dan modul-modul didalamnya dapat terintegrasi dan bekerja dengan baik. Dalam hal ini yang akan diuji adalah mengenali kendaraan yang melanggar garis *zebra cross* melalui *input* gambar perekaman kamera yang akan di proses menggunakan *raspberry pi* yang menggunakan program *library opencv* kemudian gambar objek yang berupa data akan dikirimkan ke web. Analisis yang akan dilakukan adalah mengukur efektifitas metode *transformasi hough* pada pendeteksian garis *zebra cross* melalui *input-an video*[7].

- Perancangan *contour area*

Pada perancangan *contour area* ini data masukan berasal dari *video* dari kamera yang berisi kumpulan informasi *biner* yang memuat *intensitas* piksel yang ada di dalamnya. Sedangkan *output* dari perancangan *contour area* berupa deteksi bentuk pada *zebra cross* yang nantinya akan diproses dalam proses *filter* atau penyaringan guna mendapatkan hasil bentuk *zebra cross* yang baik..

- Perancangan *background subtraction*

Pada perancangan ini *background subtraction* digunakan untuk mempermudah *programmer* dalam melakukan pendeteksian pelanggar setelah dilakukannya proses *contour area*.

- Perancangan Deteksi Pelanggaran Saat Lampu Merah

Kasus pelanggaran lalu-lintas sering terjadi pada saat lampu merah menyala adalah kendaraan berhenti di *zebra cross* serta yang melewati garis *zebra cross*. Untuk dapat mendeteksi pelanggaran yang terjadi pada saat keadaan *traffic light* berwarna merah sedang menyala, ada beberapa tahapan pemrosesan gambar yang dilakukan. Tahapan-tahapan tersebut adalah pengubah gambar menjadi *threshold* atau proses perubahan gambar menjadi hitam dan putih, proses pendeteksian *contour area* demi mendapatkan bentuk dari *zebra cross*, proses klasifikasi objek guna mendapatkan data pelanggar.

B. parameter

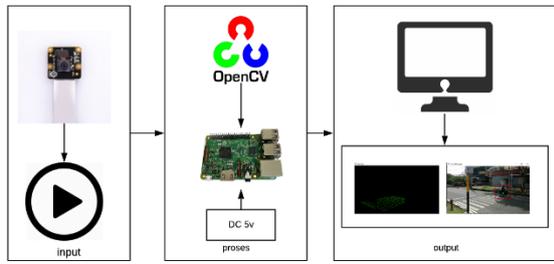
Dalam perancangan serta pembuatan alat terdapat parameter yang digunakan. Dalam pengujian yang bertujuan mengetahui kualitas dari perancangan yang kemudian akan diteliti serta dianalisis dengan harapan diperoleh data-data mengenai kinerja alat yang bermanfaat kedepannya.

Sudut pengambilan kamera merupakan besaran yang mengacu pada berapa panjang lintasan telah melewati selama gerakannya[8]. Penggunaan variasi *threshlod* sebagai parameter pada penelitian tugas akhir ini adalah bertujuan untuk melakukan pengujian pada saat *user* mengatur sudut pengambilan video untuk memicu langkah awal sebagai proses pendeteksian objek *zebra cross*, kemudian pengujian terhadap *contour area* bertujuan untuk pendeteksian pada objek *zebra cross*, serta untuk pengujian terhadap objek kendaraan menggunakan metode *background subtraction*[9]. Dan yang terakhir yaitu klasifikasi objek yang melakukan pelanggaran pada *zebra cross* demi mengetahui berapa luas objek kendaraan untuk melakukan klasifikasi jenis kendaraan serta mengetahui pada waktu ke berapa pelanggar melakukan pelanggaran terhadap *zebra cross*.

C. Model Arsitektur Skema

Sebagaimana telah dijelaskan pada pendahuluan , pada penelitian ini akan dibangun “*system* pendeteksi pelanggar *zebra cross* pada *traffic light* menggunakan *raspberry pi* berbasis pengolahan citra digital” sebagai alat yang mendeteksi bahwa adanya pelanggaran pada pengendara kendaraan bermotor khususnya pada *zebra cross*[10]. Alat ini mendeteksi adanya pelanggaran yang dilakukan oleh pengendara kendaraan bermotor dengan bantuan dari sensor kamera Pi NoIR versi 2 sebagai pengambil video pada *traffic light* pelanggar kemudian gambar akan di proses melalui *library opencv* serta akan menampilkan data pelanggar sesuai dengan jenis kendaraan. Alur kinerja alat adalah sebagai berikut :

1. *Raspberry pi 3B* bertugas sebagai pengendali utama atau sebagai perangkat minikomputer guna melakukan semua proses pengolahan gambar/video sampai mendapatkan hasil dari pendeteksian pelanggar *zebra cross*.
2. Agar sistem dapat berfungsi, perlu dilakukan pengaturan pada kualitas video yang dihasilkan yaitu 360p dikarenakan *system* pada *raspberry pi* bekerja dengan baik dengan kualitas video 360p apabila lebih dari 360p maka pemrosesan *system* pada *raspberry pi* akan sangat berat dijalankan bahkan banyak mengalami error dalam pendeteksian objek pelanggar *zebra cross* .

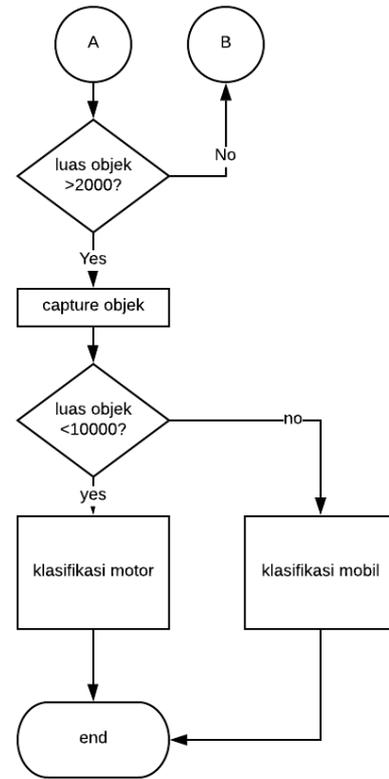
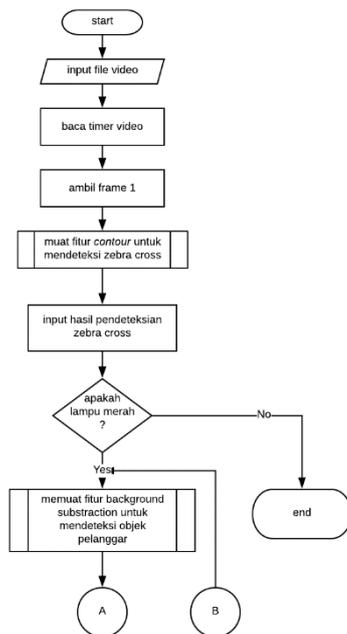


Gambar 1. Bagan kinerja perangkat

Pada gambar 1 dijelaskan bahwa *camera pi Noir* digunakan sebagai pengambil data *video* kemudian hasil dari pengambilan video akan diteruskan ke *library open cv* yang telah *terinstall* pada perangkat *raspberry pi*, file video yang telah di *input* ke *open cv* maka video akan diproses menggunakan metode *contour area* untuk pendeteksian citra bentuk *zebra cross* kemudian setelah proses pendeteksian bentuk pada *zebra cross* yaitu proses pendeteksian objek pelanggar menggunakan metode *background subtraction*[11]. Setelah hasil dari citra bentuk dari *zebra cross* didapatkan serta objek pelanggar telah terdeteksi maka output hasilnya akan ditampilkan pada monitor.

D. Flowchart system

Tampilan dari *flowchart* dapat dilihat pada gambar 2. *Start* adalah langkah awal yang harus dilakukan *user* lalu masuk pada tahap *input* file *video* yang bertujuan untuk melakukan perhitungan data dari hasil *capture* gambar pelanggar *zebra cross* pada tahap ini pembuatan diagram alir atau *flowchart* sistem deteksi pelanggaran *zebra cross* pada *traffic light*[12]. Secara keseluruhan sistem dapat dilihat pada gambar 2.



Gambar 2. Flowchart system

E. metode yang digunakan

berdasarkan hasil dari penggambaran yang dihasilkan berdasarkan *video* ditentukan pada model *background*. Selanjutnya dilakukan proses pemodelan citra *background* secara *manual* dengan memilih atau mengambil citra digital dimana sebelum adanya *object* di dalamnya. Pemodelan berdasarkan *background* ini dilakukan secara *manual* dengan batasan bahwa lingkungan bersifat statis sehingga citra latar belakang tanpa objek juga bersifat statis pada lingkungan statis dikarenakan teknik ini lebih efektif dan cepat jika dibandingkan penerapan pemodelan citra latar belakang adaptif[13].

III. HASIL DAN ANALISA

Pengujian pada penelitian ini terdapat 4 macam pengujian yang dilakukan. Pada pengujian pertama dilakukan dengan tujuan untuk mengetahui pengaruh proses penentuan pada nilai *threshold*, kemudian untuk pengujian kedua dilakukan untuk mengetahui parameter *contour area* dari *zebra cross* demi memperbaiki citra dari *images*, lalu pengujian *background subtraction* dilakukan untuk melakukan pendeteksian pada objek kendaraan, dan terakhir adalah pengujian klasifikasi objek pada pelanggaran yaitu demi mengetahui berapa pixel yang

dihasilkan serta pada waktu ke berapa pelanggar melakukan pelanggaran dalam video[14].

A. Pengujian variasi Threshold

Pada pengujian *threshold* yang digunakan pada pengujian ini yaitu menggunakan nilai *threshold* $t = 100$, $t = 150$, $t = 200$. Pada pengujian ini menggunakan jumlah data uji yang terdapat *zebra cross* pada kualitas/resolusi video yang digunakan yaitu 360p, dari pengujian variasi nilai *threshold* dapat menghasilkan hasil gambar *zebra cross*. dari pengujian nilai *threshold* hasil terbaik ini akan digunakan untuk melakukan pengujian kontur area.

Pada pengujian *threshold* dengan nilai 100 didapatkan hasil seperti ditunjukkan pada gambar



Gambar 3. pengujian threshold 100

Pada gambar 3 yaitu menunjukkan pengujian *threshold* dengan menggunakan nilai yaitu 100 dari gambar 4.1 dapat dilihat bahwa pada daerah *zebra cross* tidak terlihat dengan jelas dikarenakan *threshold* terlalu rendah sehingga warna putih lebih dominan sehingga informasi citra bentuk dari *zebra cross* hilang, sehingga citra yang dihasilkan dapat membuat kesalahan dalam pendeteksian sebuah *zebra cross*.

Setelah itu pengujian *threshold* dengan nilai 150 dengan hasil ditunjukkan pada gambar 4.



Gambar 4. Pengujian Threshold 150

Pada gambar 4 yaitu menunjukkan pengujian *threshold* dengan menggunakan nilai yaitu 150 dari gambar 4 dapat dilihat bahwa pada daerah *zebra cross* sudah terlihat dengan jelas tetapi nilai *threshold* masih terlalu kecil sehingga membuat gambar pada daerah jalan membuat citra gambar masih terlalu putih sehingga pada saat akan melakukan proses pendeteksian *zebra cross* dapat membuat kesalahan pada saat pendeteksian *zebra cross*[15].

Setelah itu pengujian *threshold* dengan nilai 200 dengan hasil ditunjukkan pada gambar 5.



Gambar 5. pengujian threshold 200

Pada gambar 5 yaitu menunjukkan pengujian *threshold* dengan menggunakan nilai yaitu 200 dari gambar 4.4 dapat dilihat bahwa pada daerah *zebra cross* sudah terlihat dengan jelas tetapi dikarenakan nilai *threshold* terlalu tinggi sehingga pada citra gambar yang dihasilkan pada pengujian nilai *threshold* 200 membuat gambar terlalu gelap sehingga daerah pada *zebra cross* banyak yang hilang[16].

Setelah melakukan pengujian variasi nilai *threshold* didapatkan nilai *threshold* terbaik yaitu 180 untuk hasil gambar *threshold* terbaik ditunjukkan pada gambar 6.

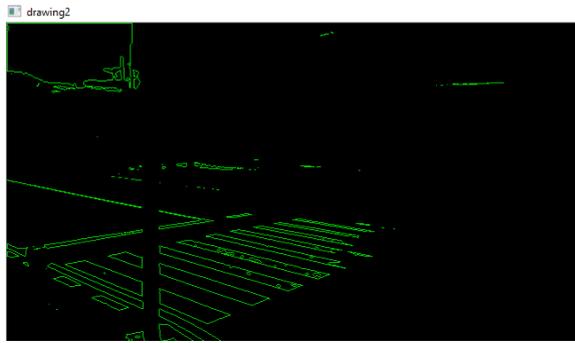


Gambar 6. threshold terbaik 180

Pada gambar 4.5 yaitu menunjukkan pengujian *threshold* terbaik dengan menggunakan nilai yaitu 180 dari gambar 4.5 dapat dilihat bahwa pada daerah *zebra cross* sudah terlihat dengan jelas serta hasil pada area *zebra cross* yang ditampilkan sudah hampir menyeluruh sehingga dapat melakukan pendeteksian *zebra cross* dengan menggunakan *countur area* [18].

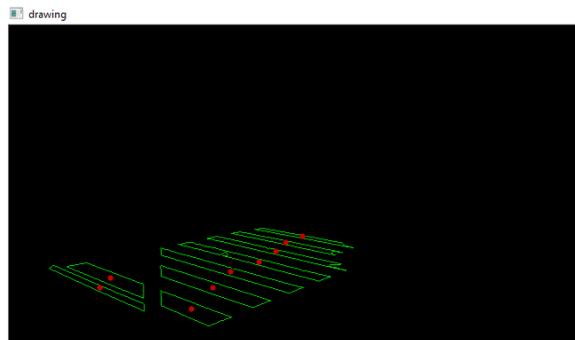
B. Pengujian Countur Area

Setelah mendapatkan hasil terbaik dari proses pengujian nilai *threshold* maka proses selanjutnya adalah melakukan perbaikan citra agar mendapatkan bentuk objek dari *zebra cross*. Proses perbaikan citra dengan menggunakan *contour area* sebelum dilakukan proses penyaringan pada daerah *zebra cross* menghasilkan tampilan keseluruhan *contour* seperti ditunjukkan pada gambar



Gambar 7. tampilan keseluruhan *contour*

Dapat dilihat dari gambar pada tampilan keseluruhan *contour* setelah dilakukannya proses penentuan *threshold* maka langkah yang dilakukan adalah memberikan perintah, `cv.findContours(thresh, cv.RETR_TREE, cv.CHAIN_APPROX_SIMPLE)` maka program akan mendeteksi *contour* dari keseluruhan hasil citra yang telah diperoleh dari proses penentuan nilai *threshold* maka menghasilkan gambar seperti yang ditunjukkan pada gambar. Setelah melakukan pencarian *contour*, maka hasil dari pencarian *contour* dari seluruh tampilan gambar proses selanjutnya adalah melakukan filter dari bentuk *contour* agar mendapatkan bentuk dari *zebra cross*, untuk hasil gambar dari proses filter dari *contour* ditunjukkan pada gambar 8.



Gambar 8. *Contour* Dari *Zebra Cross*

Pada gambar 8 menunjukkan bentuk dari *countur* yang telah dilakukan proses filter menggunakan perintah `if Area >186 and arc >200 and mc[i][0]>100:` proses penyaringan atau filter bertujuan untuk memilih luas, keliling, serta *centroid* pada *contour* agar dapat mendapatkan hasil dari objek *zebra cross*. dari pendeteksian objek *zebra cross* didapatkan hasil data dari bentuk *zebra cross* seperti ditunjukkan pada tabel 1.

Tabel 1. Data *Contour* *Zebra Cross*

<i>Contour</i> (number)	Luas (pixel)	Keliling (pixel)	Koordinat Centroid (x,y)
8	1220.5	202.8	205, 320
11	1512.0	292.8	229, 296

12	727.5	260.4	102, 296
14	1099.0	214.5	114, 285
21	1385.0	371.4	249, 278
27	1053.5	392.7	281, 267
37	757.0	370.3	300, 255
39	470.5	281.0	311, 245
46	301.0	240.2	330, 238
53	187.0	232.6	197, 224

Berdasarkan tabel 1. Gambar pendeteksian *zebra cross* menunjukkan data *contour*, dikarenakan setiap *contour* memiliki label atau penomoran, serta setiap *contour* memiliki luas serta bentuk masing-masing *contour*, dan keliling *contour* dalam bentuk pixel, dan koordinat *centroid* dari masing-masing *contour*.

C. pengujian pendeteksian objek

Pada penelitian ini pengujian pendeteksian objek menggunakan metode yaitu *background subtraction* ditujuk untuk hasil pengujian pendeteksian objek dilakukan pada saat kondisi jalan masih kosong serta setelah adanya objek kendaraan yang melewati jalan. Untuk hasil pada pengujian pendeteksian objek sebelum adanya objek dapat dilihat pada gambar 9 [17].



Gambar 9. *frame* sebelum adanya objek kendaraan

Pada gambar 9 menunjukkan pengambilan gambar pada saat kondisi jalan masih kosong atau masih belum adanya objek kendaraan yang melewati jalan. untuk hasil pendeteksian objek pada saat objek kendaraan telah melewati jalan dapat dilihat pada gambar.



Gambar 10. *frame* setelah adanya objek kendaraan

Pada gambar menunjukkan pengambilan gambar pada saat kondisi setelah adanya objek kendaraan yang melewati *zebra cross*.

D. pengujian klasifikasi objek kendaraan

Pada proses pengujian klasifikasi objek bertujuan agar mengetahui berapa banyak kendaraan yang melanggar, serta data pelanggaran dari jenis kendaraan dan pada detik keberapa kendaraan melakukan pelanggaran. Untuk data klasifikasi objek kendaraan dapat dilihat pada tabel 2.

Tabel 2. klasifikasi objek kendaraan

no	Luas objek (pixel)	Waktu pelanggaran (second)	Jenis kendaraan
1.	8395.0	2	Motor
2.	17145.5	12	Mobil
3.	<i>error</i>	-	Motor
4.	5286.5	28	Motor
5.	<i>error</i>	-	Mobil
6.	5844.5	72	Motor
7.	9576.5	91	Motor
8.	10317.0	100	Mobil

Berdasarkan data pengujian yang di tampilkan pada tabel 2 dari klasifikasi objek kendaraan dapat dilihat pada yaitu setiap kendaraan memiliki luas objek yang berbeda seperti luas objek kendaraan pada roda dua atau yang biasa disebut dengan motor memiliki luas objek 5000 pixel sampai dengan 9500 *pixel*, sedangkan pada kendaraan roda empat atau yang biasa disebut dengan mobil memiliki pixel lebih dari 10000 *pixel*. Pada tabel 2 juga terdapat data dari waktu pelanggaran yaitu proses berdasarkan dari waktu objek melakukan pelanggaran,serta pada tabel data klasifikasi objek kendaraan juga terdapat *error* dimana sistem tidak membaca objek pada pelanggar sehingga berdasarkan data dari luas objek dan waktu pelanggaran tidak dapat terbaca.

IV. KESIMPULAN

Berdasarkan pengujian dan analisis maka ditarik beberapa kesimpulan antara lain yaitu, Perancangan sistem pendeteksi pelanggar pada *zebra cross traffic light* menggunakan metode *background subtraction* sebagai pendeteksi objek kemudian menggunakan metode *contour* sebagai pendeteksi bentuk dari objek tersebut setelah pendeteksian bentuk dari objek maka dicari *centroid* dari luasan objek yang terdeteksi, setelah itu pada pendeteksian pelanggar dilakukan dengan mentracking *centroid* apakah melewati garis deteksi yang telah ditentukan, jika *centroid* objek melewati garis maka objek akan *ter-capture* setelah objek *ter-capture* maka data akan tersimpan otomatis pada perangkat *raspberry pi*

. perangkat *raspberry pi* merupakan mini komputer yang digunakan sebagai pemrosesan segala sistem.

Penerapan citra digital pada sistem pendeteksi pelanggar pada *zebra cross* menggunakan *opencv* sebagai *library* yang digunakan serta menggunakan bahasa pemrograman *python* yang di olah di dalam mini komputer yaitu *raspberry pi*.

Untuk pengembangan perangkat keras dan lunak lebih lanjut agar hasilnya lebih baik lagi, maka ada beberapa hal yang dapat ditambahkan antara lain, Berdasarkan pengujian perangkat secara *real* dengan melihat kondisi dari pewartuan pada lampu merah dapat berubah sewaktu waktu dikarenakan sistem pada lampu merah dapat mendeteksi keadaan banyaknya kendaraan. Jadi untuk pengembangan selanjutnya diharapkan dapat menggunakan 2 kamera dengan salah satunya berfungsi sebagai pendeteski apabila keadaan

UCAPAN TERIMAKASIH

Ucapan terimakasih kepada pembimbing satu dan dua yang selalu memberikan pengarahan sehingga penelitian ini dapat terlaksana dan tercapai sesuai dengan tujuan dan ucapan terimakasih kepada kawan-kawan yang selalu memberi dukungan hingga sampai detik ini..

DAFTAR PUSTAKA

- [1] T. A. S, A. Rabi', D. Minggu, and I. Mujahidin, "Frequency Hopping Video Real Time Untuk Pengamanan Data Pengintaian Operasi Inteligence TNI," *JASIEK (Jurnal Apl. Sains, Informasi, Elektron. dan Komputer)*, 2019.
- [2] D. A. Prasetya, A. Sanusi, G. Chandrarin, E. Roikhah, I. Mujahidin, and R. Arifuddin, "Community Culture Improvisation Regarding Waste Management Systems and Per Capita Income Increase," *J. Southwest Jiaotong Univ.*, vol. 54, no. 6, 2019.
- [3] F. Pakaja and A. Naba, "Jaringan Syaraf Tiruan dan Certainty Factor," *Neural Networks*, 2015.
- [4] S. Santika, N. Nachrowie, D. A. Prasetya, and B. F. Hidayatulail, "Mini Plant Sistem Pengendali Berat Limestone Pada Pltu Tanjung Jati B Unit #3&4 Berbasis Plc Dan Arduino Mega 2560," *JASIEK (Jurnal Apl. Sains, Informasi, Elektron. dan Komputer)*, 2019.
- [5] Y. Prayudi and I. Aprizal, "Pemodelan Wajah 3D Berbasis Foto Diri Menggunakan Maya Embedded Language (MEL) Script," *Media Inform.*, 2017.
- [6] D. A. Prasetya, A. Sanusi, G. Chandrarin, E. Roikhah, I. Mujahidin, and R. Arifuddin, "Small and Medium Enterprises Problem and Potential Solutions for Waste Management," *J. Southwest Jiaotong Univ.*, vol. 54, no. 6, 2019.
- [7] S. K. Sugiarto, I. Mujahidin, and A. B. Setiawan, "2, 5 GHz Antena Mikrostrip Polarisasi Circular Model

- Patch Yin Yang untuk Wireless Sensor,” *JEECAE (Journal Electr. Electron. Control. Automot. Eng.*, vol. 4, no. 2, pp. 297–300, 2019.
- [8] I. Mujahidin and B. F. Hidayatulail, “2.4 GHz Square Ring Patch With Ring Slot Antenna For Self Injection Locked Radar,” *JEEMECS (Journal Electr. Eng. Mechatron. Comput. Sci.*, vol. 2, no. 2, 2019.
- [9] B. F. Hidayatulail and I. Mujahidin, “Potential OF 77, 78 mW Red Diode Laser For Photodynamic,” *JEEMECS (Journal Electr. Eng. Mechatron. Comput. Sci.*, vol. 2, no. 2, 2019.
- [10] J. Florez, F. Calderon, and C. Parra, “Video stabilization taken with a snake robot,” in *Symposium of Signals, Images and Artificial Vision - 2013, STSIVA 2013*, 2013.
- [11] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*. 2015.
- [12] I. Mujahidin, S. H. Pramono, and A. Muslim, “5.5 Ghz Directional Antenna with 90 Degree Phase Difference Output,” 2018.
- [13] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, “Deep learning for visual understanding: A review,” *Neurocomputing*, 2016.
- [14] B. Matheus and H. Schiabel, “A CADx Scheme in Mammography: Considerations on a Novel Approach,” in *ADVCOMP 2013: The Seventh International Conference on Advanced Engineering Computing and Applications in Sciences*, 2013.
- [15] S. Rege, R. Memane, M. Phatak, and P. Agarwal, “2D Geometric Shape And Colorrecognition Usingdigital Image Processing,” *Int. J. Adv. Res. Electr. Electron. Instrum. Energy*, 1970.
- [16] S. Sjogelid, *Raspberry Pi for Secret Agents*. 2013.
- [17] D. J. Norris, *Beginning Artificial Intelligence with the Raspberry Pi*. 2017.
- [18] DA Prasetya, EFA Phong Thanh Nguyen, Rinat Faizullin, Iswanto Iswanto, “Resolving the Shortest Path Problem using the Haversine Algorithm,” in *Journal of critical reviews* 7 (1), 62-64, 2020.
- [19] Zhang, Y., Yasuno, T., DA Prasetya, “Adaptive walking for quadrupe robot on irregular terrain by using CPG network,” in *52nd Annual Conference of the Society of Instrument and Control Engineers of Japan*, Pages 1734-1737, 2013.

“ Halaman Ini Sengaja Dikosongkan ”