

IMAGE PROCESSING WARNA UNTUK PROSES INTERUPSI LED PADA LAMPU LALU LINTAS BERBASIS OPEN CV

Prihatin Oktivasari¹, Erwin Ardiansyah²

^{1,2}Program Studi Teknik Informatika, Konsentrasi Teknik Komputer dan Jaringan
Politeknik Negeri Jakarta

Kukusan, Beji, Kota Depok, Jawa Barat 16425

Email : prihatin.oktivasari@tik.pnj.com

Diterima : 21 Juli 2017 | Direvisi : 21 Juli 2017 | Disetujui : 31 Juli 2017

Abstrak

Lampu lalu lintas merupakan alat yang digunakan untuk mengatur kendaraan bermotor pada sebuah persimpangan jalan agar tidak terjadi kemacetan dan arus kendaraan dapat berjalan dengan baik dan lancar. Kemacetan lalu lintas bisa menyebabkan pengurangan efisiensi waktu dan biaya. Kendaraan darurat seperti ambulans, pemadam kebakaran dan kendaraan polisi yang akan melintasi lampu lalu lintas juga pasti akan terhambat di perjalanan untuk melakukan pertolongan darurat ke tempat tujuan. Dalam penelitian ini, dilakukan perancangan sistem cerdas pengaturan lampu lalu lintas untuk kendaraan darurat dengan menggunakan *image processing* berbasis OpenCV dan *python programming*. Citra yang diolah berupa objek warna contessa. Sistem ini memiliki delay yang sangat kecil dalam pergerakan objek yang lamban, sehingga dapat bermanfaat bagi kendaraan darurat seperti ambulans, pemadam kebakaran dan kendaraan kepolisian agar tidak terjebak kemacetan pada persimpangan traffic light. Kondisi maksimum citra warna contessa yang dapat ditangkap terbaik oleh kamera dan sistem berjalan optimal yaitu pada jarak maksimum 90 cm, intensitas cahaya 9600 lumen dengan *delay* untuk proses interupsi LED berubah warna hijau adalah 0 detik..

Kata kunci : *image processing*, lampu lalu lintas, *python*, warna contessa

IMAGE PROCESSING COLOR FOR LED INTERRUPTION PROCESS ON TRAFFIC LIGHT BASED OPEN CV

Abstract

Traffic light is a tool used to set motor vehicles at a crossroads in order to avoid congestion and the flow of vehicles can run well and smoothly. Traffic congestion can lead to reduced efficiency of time and cost. Emergency vehicles such as ambulances, firefighters and police vehicles that will cross the traffic light will also be hampered on the way to emergency relief to the destination. In this research, intelligent system design of traffic light arrangement for emergency vehicle using OpenCV based image processing and python programming. The image is processed in the form of contessa color object. This system has a very small delay in the movement of objects that are slow, so it can be beneficial for emergency vehicles such as ambulances, firefighters and police vehicles so as not to get caught in traffic jam junction. Maximum condition of contessa color image that can be captured best by camera and optimal running system that is at maximum distance of 90 cm, light intensity 9600 lumen with delay for interruption process LED turn green color is 0 seconds.

Keywords : *image processing*, traffic light, *python*, color contessa

PENDAHULUAN

Lampu lalu lintas merupakan alat yang digunakan untuk mengatur kendaraan bermotor pada sebuah persimpangan jalan agar tidak terjadi kemacetan dan arus kendaraan dapat berjalan dengan baik dan lancar. Kemacetan juga dapat menyebabkan masalah penting untuk kendaraan darurat seperti ambulans, pemadam kebakaran dan kendaraan polisi yang pasti akan terhambat di perjalanan untuk

melakukan pertolongan darurat di tempat tujuan. Oleh karena itu, diperlukan rancangan sebuah sistem lalu lintas perempatan yang cerdas yang dapat mengatur dan memberikan prioritas utama bagi kendaraan darurat yang akan melewati lampu lalu lintas sehingga kendaraan darurat dapat lebih cepat sampai lokasi tujuan. Dengan adanya perkembangan teknologi lampu lalu lintas ini, kemacetan dapat diatasi dengan sebuah manajemen lalu lintas yang baik pada setiap persimpangan.

Penelitian ini membuat sistem cerdas pada lampu lalu lintas menggunakan *image processing* berbasis python programming. Penelitian ini merancang sebuah sistem lalu lintas dengan memanfaatkan teknologi pengolah citra untuk memberikan prioritas “lampu hijau” bagi kendaraan darurat yang terjebak didalam kemacetan pada persimpangan 4 arah.

Hal ini berdasarkan pada penelitian sebelumnya, seperti yang dilakukan oleh K. B. Neelima dan T. Saravanan [1], pernah merancang sebuah sistem count dengan metode pendeteksi objek menggunakan pengolahan citra menggunakan OpenCV, output yang dihasilkan yaitu dengan melakukan perhitungan terhadap jumlah objek yang terdeteksi. Penelitian serupa telah dilakukan oleh Sonia et al [2], menggunakan metodologi Fuzzy Logic dalam kontrol lampu lalu lintas untuk melakukan observasi monitoring terhadap kepadatan lalu lintas 4 persimpangan dengan memberikan prioritas lebih untuk lajur yang memiliki kepadatan. Nidhi Dept. of Computer Applications [3] mendiskusikan tentang bagaimana menangkap gambar dari kamera web secara real time dan bagaimana cara memprosesnya. Mamata S. Kalas. More et al [4] membahas tentang perancangan sistem face detection secara real time dengan menggunakan OpenCV, dan menggunakan algoritma adaabost untuk melacak wajah pada platform OpenCV dan dikembangkan oleh intel. Vijayalaxmi et al [5] menguraikan tentang deteksi objek berdasarkan suatu warna tertentu menggunakan OpenCV. Naveen Kumar et al [6] mengangkat permasalahan tentang bagaimana membuat suatu sistem yang dapat mendeteksi tanda jalan secara otomatis.

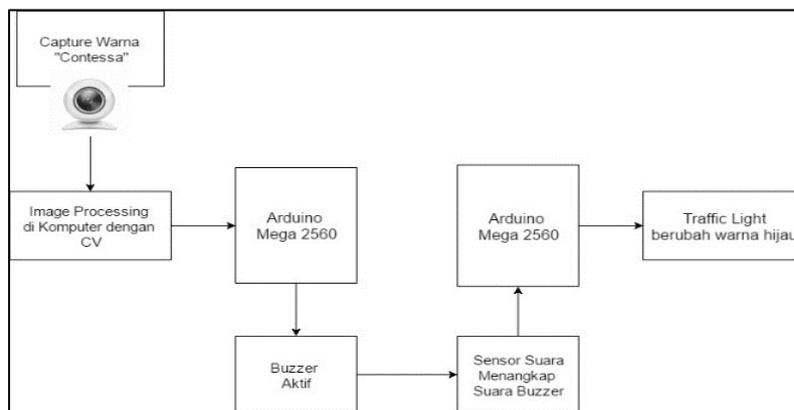
METODOLOGI PENELITIAN

Sistem ini bekerja menggunakan prinsip kamera, akan mendeteksi warna “contessa”, setelah itu

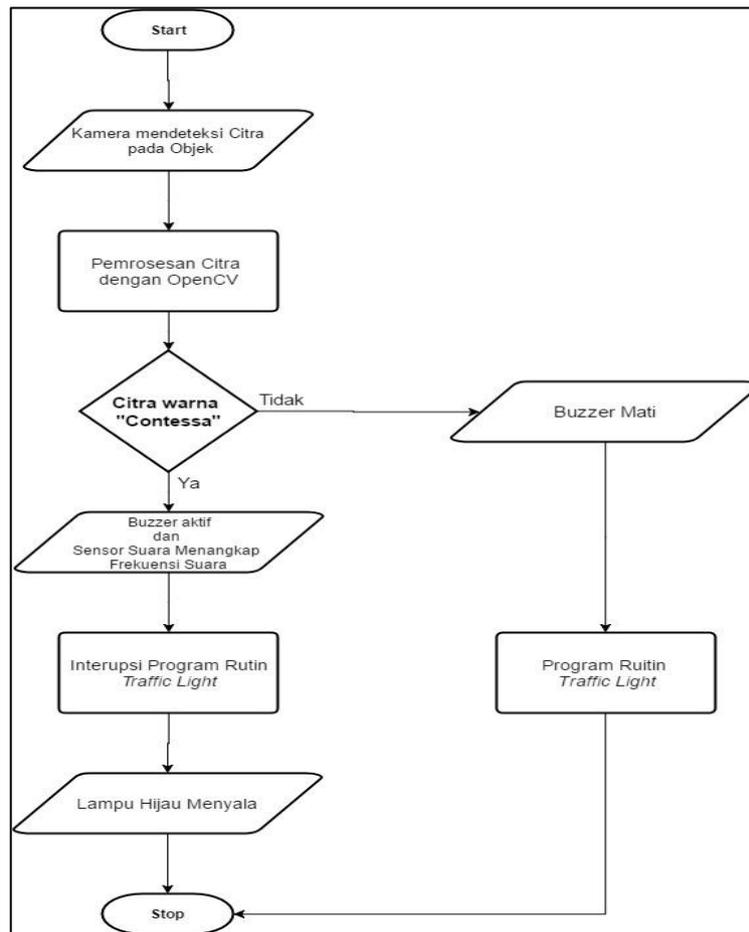
citra akan diproses pada OpenCV dengan pemberian nilai kepada Arduino menggunakan komunikasi serial. Nilai yang digunakan berupa karakter dari a – h. Nilai ini berfungsi untuk mengaktifkan buzzer. Ketika buzzer aktif maka sensor suara akan menangkap suara pada buzzer, setelah itu sensor suara akan mengaktifkan interupsi pada lampu lalu lintas dan merubahnya menjadi warna hijau. Sistem pengaturan lampu lalu lintas untuk memberikan prioritas lampu hijau pada kendaraan darurat seperti ambulans, pemadam kebakaran dan kendaraan kepolisian, untuk memberikan interupsi agar lampu tanda pada persimpangan jalan dapat berubah berwarna hijau. Sistem ini dapat membuat lampu lalu lintas lebih dinamis yang dapat mengatur dirinya sendiri dengan memberikan prioritas lampu hijau pada saat ada kondisi kendaraan darurat ingin melaju melewati persimpangan 4 arah.

Sistem memakai memiliki 2 fungsi yang berbeda, yaitu sebagai penerima data dari kamera yang mendeteksi warna “Contessa” pada objek, dan yang lainnya sebagai penerima trigger berupa frekuensi suara yang mengolah penerimaan suara untuk melakukan interupsi pada program rutin lampu lalu lintas yang sedang berjalan. PC akan menerima citra warna “Contessa” dari kamera, jika kamera mendapatkan warna citra yang sesuai, maka buzzer diaktifkan oleh Arduino pertama.

Arduino kedua yang berfungsi sebagai pengatur lampu lalu lintas dihubungkan ke sensor suara. Setelah itu sensor suara menerima frekuensi dari buzzer yang terhubung ke Arduino kedua. Jika frekuensi suara diterima oleh sensor suara, maka lampu lalu lintas berubah secara otomatis menjadi warna hijau, sehingga sistem ini sangat berguna bagi kendaraan darurat yang sedang terjebak kemacetan di persimpangan.



Gambar 1. Blok Diagram Sistem Pengaturan lampu Lalu lintas dengan *Image Processing*



Gambar 2. Flowchart Sistem Pengaturan lampu lalu lintas dengan Image Processing

Alur sistem bekerja berdasarkan penangkapan citra warna objek yang diterima oleh kamera. Kamera yang digunakan pada sistem ini menggunakan USB Webcam. Selanjutnya citra yang ditangkap kamera akan diproses menggunakan teknik image processing dengan menggunakan aplikasi OpenCV dan Python Programming. Setelah itu PC akan menandai objek yang diterima jika warna sesuai dengan warna yang telah ditentukan pada pemrograman Python. Pada pemrograman Python telah ditentukan warna yang dapat dijadikan trigger, warna tersebut adalah warna "CONTESSA".

Warna RGB (Red, Green, Blue) Contessa memiliki nilai (R = 193, G = 111, B = 104) digunakan sebagai trigger. Jika warna "CONTESSA" terdeteksi, maka ATmega 2560 akan menerima nilai dari pemrograman Python. Komunikasi ini menggunakan komunikasi serial dalam pengiriman nilainya. Setelah nilai dikirim ke ATmega 2560, maka Arduino akan menjalankan programnya dengan menyalakan buzzer sesuai pin yang telah diprogram. Setelah itu sensor suara akan menerima frekuensi suara dari buzzer.

Selanjutnya sensor suara akan melakukan interupsi terhadap program rutin lampu pengatur lalu lintas yang sedang berjalan normal agar supaya merubah warna lampu lalu lintas menjadi warna hijau dari arah jalan ambulance berasal selama warna "CONTESSA" terdeteksi oleh kamera. Jika kamera tidak mendeteksi adanya warna "CONTESSA", program pengaturan lampu lalu lintas akan berjalan normal kembali untuk mengatur persimpangan 4 arah.

HASIL DAN PEMBAHASAN

Pengujian terhadap kamera meliputi pengujian kondisi apakah kamera dapat menyala dan bekerja dengan baik, lalu pengujian jarak maksimum kamera dapat mendeteksi objek warna yang telah ditentukan, dengan menganalisis jarak paling maksimal untuk dapat mendeteksi.

Tabel 1. Pengujian warna contesa

Kategori Pengujian	Hasil Pengujian
Pengujian warna dengan jarak 30 cm	
Pengujian warna dengan jarak 60 cm	
Pengujian warna dengan jarak 90 cm	

Semakin dekat jarak objek dengan kamera, maka semakin banyak berkas cahaya yang tertangkap dan semakin jelas terdeteksi, sehingga akan lebih mengoptimalkan sistem otomatisasi berjalan. Warna contessa juga dapat terdeteksi sampai kejauhan 90 cm dari posisi kamera.

Pengujian Pengaturan Warna pada Python

Pengujian pada Python merupakan hal yang tidak kalah penting agar kamera dapat mendeteksi warna objek dengan baik. Fungsi pemrograman Python disini bertujuan agar kamera dapat mengenali warna pada objek yang telah ditentukan dari *source code* python. Pada sistem ini objek yang dikenali oleh kamera yaitu warna contessa. Warna ini dipilih karena merupakan warna unik yang tidak ada pada kendaraan bermotor.

```
28 colorLower = (-3, 100, 100)
29 colorUpper = (7, 255, 255)
```

Gambar 3. Pengaturan warna pada Python



Gambar 4. Hasil Pendeteksian warna pada Python

Kamera berhasil menangkap citra lalu melakukan *tagging* terhadap warna yang telah ditentukan. Pengaturan warna yang digunakan pada sistem ini menggunakan *library* `cv2.cvtColor` dan `BGR2HSV`, sehingga nilai warna *upper* dan *lower* dapat terhitung secara otomatis.

Pengujian Intensitas Cahaya

Pengujian intensitas cahaya dilakukan untuk mendapatkan *flux* cahaya yang tepat agar pendeteksian objek dapat dilakukan walaupun dengan menggunakan cahaya lampu. Pengujian dilakukan pada pencahayaan yang berbeda dengan menghitung *flux* cahaya masing-masing lampu yang digunakan. Pencahayaan yang digunakan pada pengujian yaitu menggunakan lampu philips 10 watt, 5 watt, dan 0 watt. Rumus untuk menghitung *flux cahaya* (Lumen) adalah:

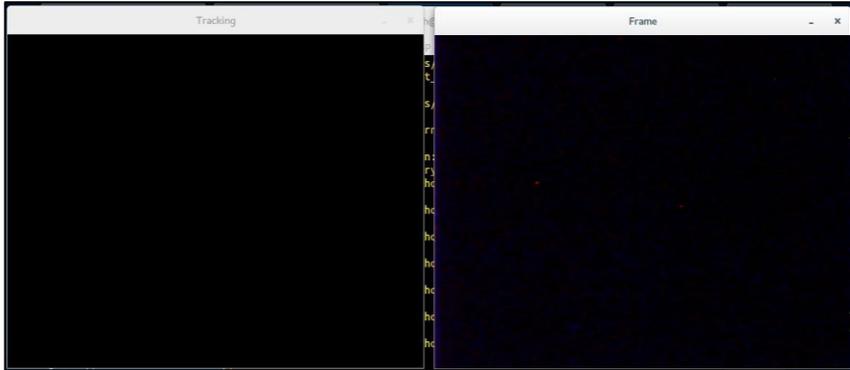
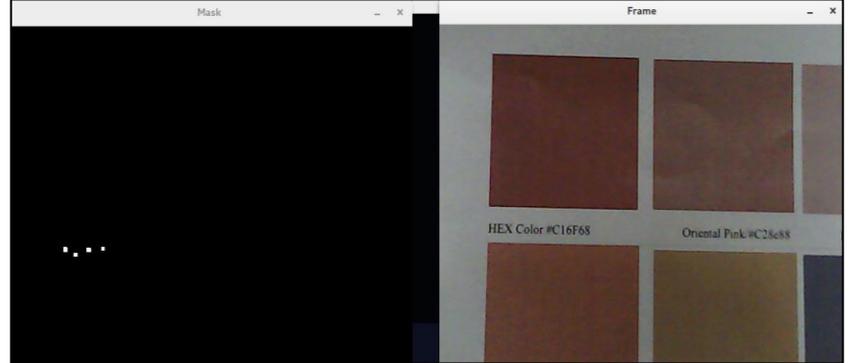
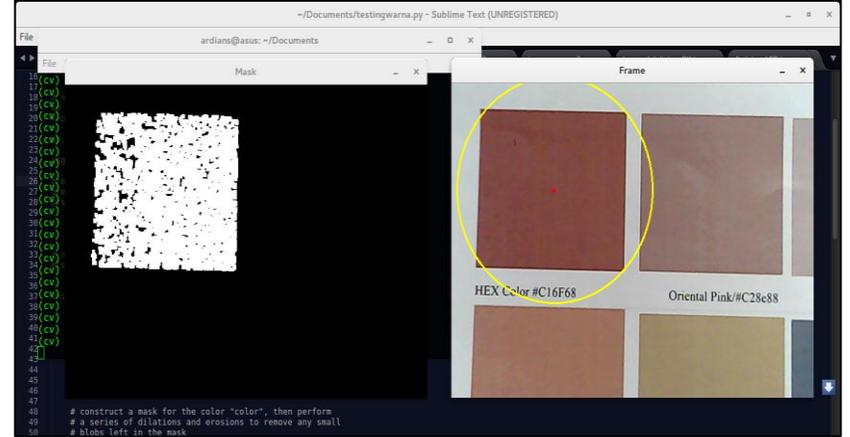
$$\Phi = W \times L/w$$

Φ = Flux cahaya (lumen)

W = Daya lampu (watt)

L/w = Luminous Efficacy Lamp (Lumen/watt): 1 watt cahaya = 680 lumen

Tabel 2. Pengujian Intensitas Cahaya

Daya (Watt)	Flux cahaya (lumen)	Pengujian deteksi warna <i>contessa</i>
0	0	
5	3400	
10	6800	

Pada nilai lumen 3400 sudah mulai terdeteksi warna *contessa*. Pada perlakuan dengan warna selain *contessa*, walaupun pada nilai lumen berapa pun tidak terdeteksi, sehingga ketepatan hasil alat mendekati 100 %.

Pengujian Pendeteksian Warna dengan Objek Bergerak

Pengujian pendeteksian warna objek berdasarkan pergerakan berfungsi untuk mengamati pendeteksian warna pada objek yang sedang bergerak. Pengujian ini menggunakan 3 kategori pergerakan, antara lain rendah, sedang, dan cepat.

Tabel 3. Hasil Pengujian warna dengan Objek Bergerak

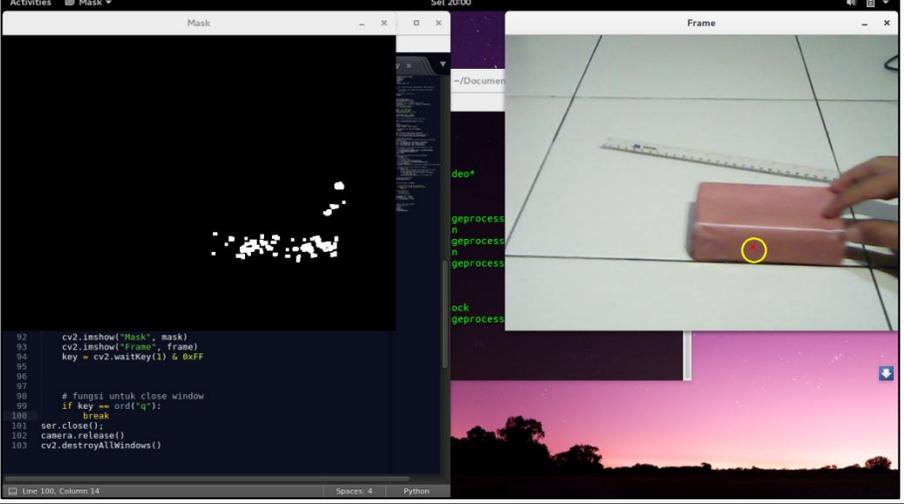
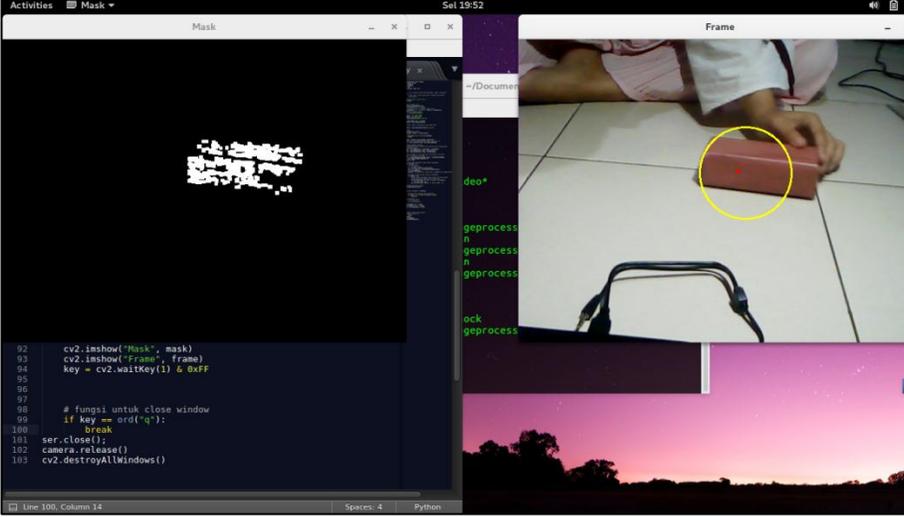
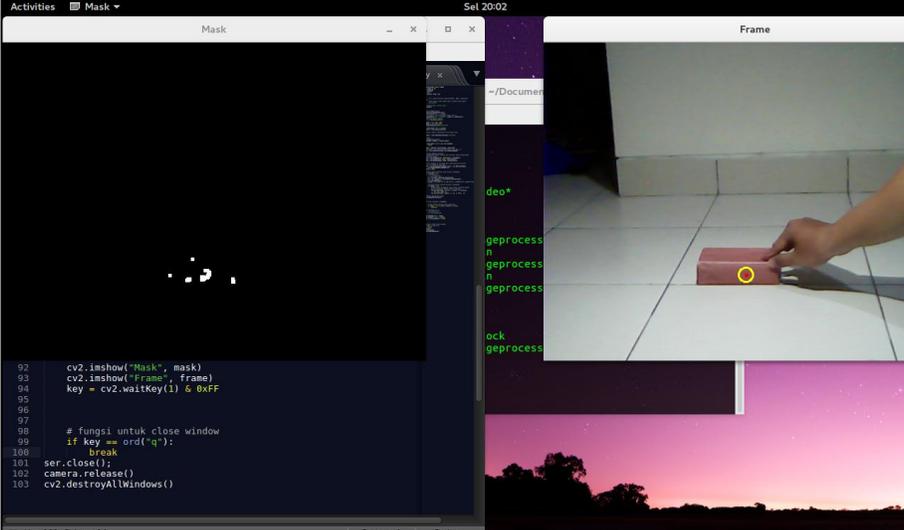
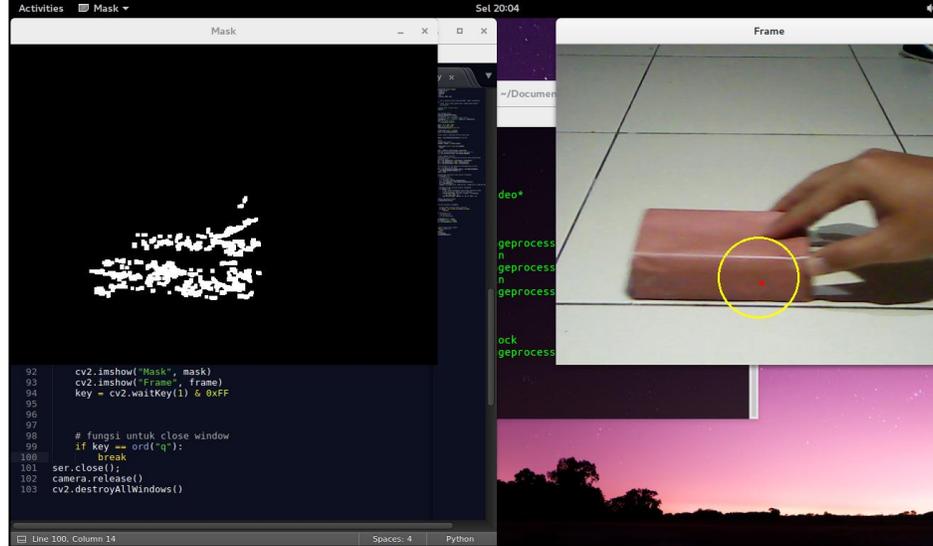
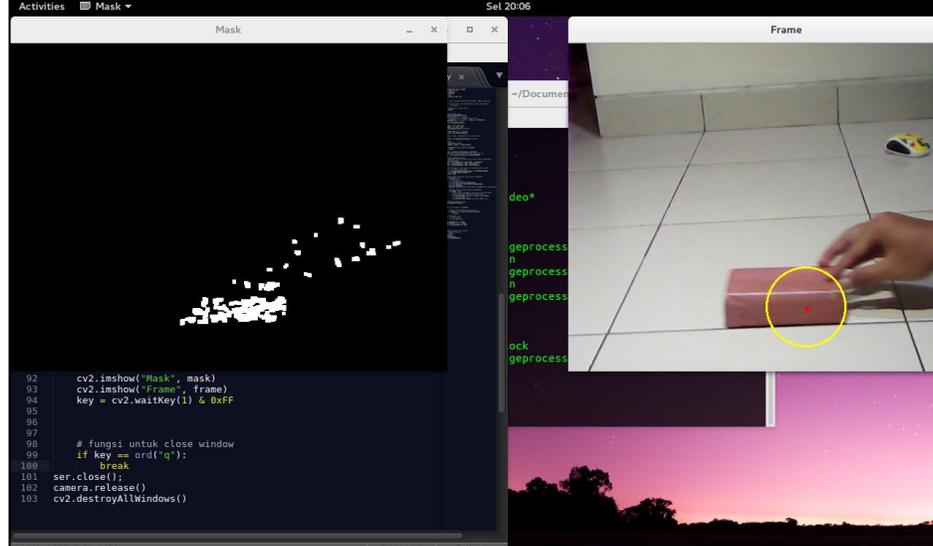
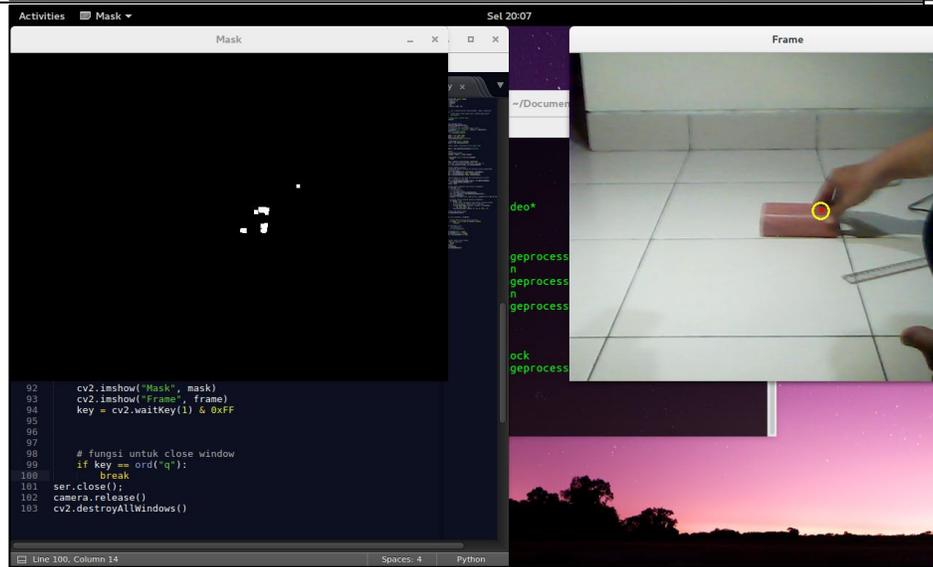
Kategori Pengujian	Hasil Pengujian
<p>Pergerakan objek rendah dan jarak 30 cm</p>	 <pre> 92 cv2.imshow("Mask", mask) 93 cv2.imshow("Frame", frame) 94 key = cv2.waitKey(1) & 0xFF 95 96 97 98 # fungsi untuk close window 99 if key == ord("q"): 100 break 101 ser.close() 102 camera.release() 103 cv2.destroyAllWindows() </pre>
<p>Pergerakan objek rendah dan jarak 60 cm</p>	 <pre> 92 cv2.imshow("Mask", mask) 93 cv2.imshow("Frame", frame) 94 key = cv2.waitKey(1) & 0xFF 95 96 97 98 # fungsi untuk close window 99 if key == ord("q"): 100 break 101 ser.close() 102 camera.release() 103 cv2.destroyAllWindows() </pre>
<p>Pergerakan objek rendah dan jarak 90 cm</p>	 <pre> 92 cv2.imshow("Mask", mask) 93 cv2.imshow("Frame", frame) 94 key = cv2.waitKey(1) & 0xFF 95 96 97 98 # fungsi untuk close window 99 if key == ord("q"): 100 break 101 ser.close() 102 camera.release() 103 cv2.destroyAllWindows() </pre>

Image Processing Warna utk Proses Interupsi LED pada Lampu Lalu Lintas Berbasis Open CV

<p>Pergerakan objek sedang dan jarak 30 cm</p>	 <pre>92 cv2.imshow("Mask", mask) 93 cv2.imshow("Frame", frame) 94 key = cv2.waitKey(1) & 0xFF 95 96 97 98 # fungsi untuk close window 99 if key == ord("q"): 100 break 101 ser.close() 102 camera.release() 103 cv2.destroyAllWindows()</pre>
<p>Pergerakan objek sedang dan jarak 60 cm</p>	 <pre>92 cv2.imshow("Mask", mask) 93 cv2.imshow("Frame", frame) 94 key = cv2.waitKey(1) & 0xFF 95 96 97 98 # fungsi untuk close window 99 if key == ord("q"): 100 break 101 ser.close() 102 camera.release() 103 cv2.destroyAllWindows()</pre>
<p>Pergerakan objek sedang dan jarak 90 cm</p>	 <pre>92 cv2.imshow("Mask", mask) 93 cv2.imshow("Frame", frame) 94 key = cv2.waitKey(1) & 0xFF 95 96 97 98 # fungsi untuk close window 99 if key == ord("q"): 100 break 101 ser.close() 102 camera.release() 103 cv2.destroyAllWindows()</pre>

<p>Pergerakan objek cepat dan jarak 30 cm</p>	
<p>Pergerakan objek cepat dan jarak 60 cm</p>	
<p>Pergerakan objek cepat dan jarak 90 cm</p>	

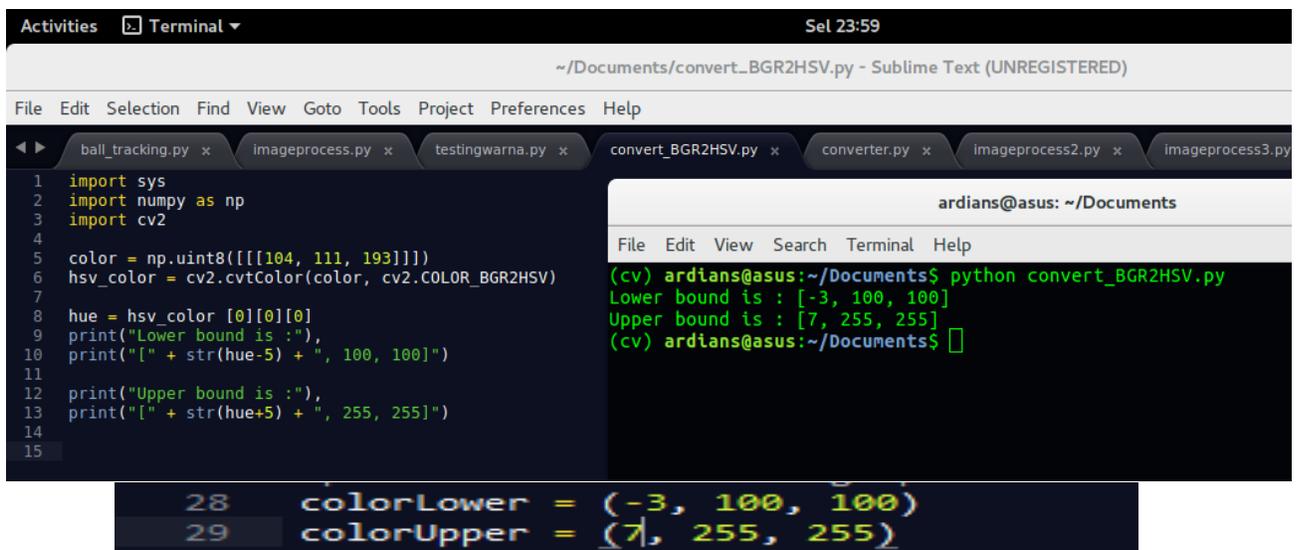
Berdasarkan hasil pengujian pada objek bergerak, dapat diambil kesimpulan, jika jarak semakin jauh maka warna sedikit terdeteksi, dan juga apabila gerakan semakin cepat, juga semakin mengecil nilai terdeteksinya. Pada jarak 90 cm dengan pergerakan cepat, kamera sudah tidak dapat menangkap warna pada objek

Pengujian Proses Interupsi LED

Program interupsi yang dibuat berjalan, memiliki delay "0" second dari pendeteksian suara sampai perubahan warna led menjadi hijau. Fungsi komunikasi serial Python, data yang dikirim berupa data bertipe string dari a – h. Selanjutnya jika pengiriman data berhasil, Arduino akan menjalankan perintah untuk menghidupkan/mematikan *buzzer* sesuai dengan nilai yang diterima dari OpenCV. Implementasinya sistem ini menggunakan 8 karakter (a, b, c, d, e, f, g, h) untuk mengubah mode LOW/HIGH pada *buzzer*.

Pengujian pada Prototype Pengaturan lampu Lalu Lintas

Berdasarkan analisis untuk pendeteksian warna contessadigunakan sebagai *trigger*, maka warna ini diimplementasikan pada prototype kendaraan darurat seperti *ambulance*, mobil polisi, mobil pemadam kebakaran, sehingga kamera akan menangkap warna dan akan interupsi LED lampu lalu lintas menjadi hijau, sehingga kendaraan darurat dapat tetap jalan tanpa hambatan lalu lintas yang merah, sehingga diharapkan keadaan darurat segera dapat teratasi. Pengujian kamera dan lampu lalu lintas terlihat pada gambar 5, 6 dan 7. Sistem ini memerlukan penentuan warna dalam operasionalnya. Warna tersebut didapat dari fungsi konversi CV yang menggunakan fungsi *cvtColor* dan *BGR2HSV* yang ditunjukkan oleh gambar 5. Sistem ini menggunakan warna RGB contessa dengan nilai (R = 193, G = 111, B = 104). Konversi ini merupakan metode yang telah disediakan oleh *library cv2*. Nilai dari warna BGR akan langsung dikonversi untuk mendapatkan nilai *range* warna pada objek.



```
1 import sys
2 import numpy as np
3 import cv2
4
5 color = np.uint8([[104, 111, 193]])
6 hsv_color = cv2.cvtColor(color, cv2.COLOR_BGR2HSV)
7
8 hue = hsv_color [0][0][0]
9 print("Lower bound is :"),
10 print("[ " + str(hue-5) + " , 100, 100]")
11
12 print("Upper bound is :"),
13 print("[ " + str(hue+5) + " , 255, 255]")
14
15
```

```
(cv) ardians@asus:~/Documents$ python convert_BGR2HSV.py
Lower bound is : [-3, 100, 100]
Upper bound is : [7, 255, 255]
(cv) ardians@asus:~/Documents$
```

```
28 colorLower = (-3, 100, 100)
29 colorUpper = (7, 255, 255)
```

Gambar 5. Konversi warna Contessa



Gambar 6. Pengujian pada *prototype* lampu lalu lintas



Gambar 7. Pengujian warna contessa pada *prototype* kendaraan ambulance

KESIMPULAN

Berdasarkan pada hasil pengamatan dan pengujian alat dan sistem, maka dapat diambil kesimpulan bahwa kondisi maksimum citra warna yang dapat ditangkap terbaik oleh kamera dan sistem berjalan optimal yaitu pada jarak maksimum 90 cm, intensitas cahaya 9600 lumen dengan *delay* untuk proses interupsi LED berubah warna hijau adalah 0 detik.

DAFTAR PUSTAKA

K.B.Neelima, Dr.T.Saravanan. "Image Detection and Count Using Open Computer Vision (Opencv)." *Journal of Engineering*

Research and Applications Vol. 04 : 107-109. Web. September 2014

Sonia Dian Maniswari, Angga Rusdinar, Bedy Purnama. "Smart Traffic Light Menggunakan Image Processing dan Metode Fuzzy Logic". *International Journal of Advanced Research in Computer and Communication Engineering*, Volume (5) Issue (9), Juni 2015.

Nidhi Dept. of Computer Applications. "Image Processing and Object Detection"; *International Journal of Applied Research*, Volume (1) Issue (9), 2015.

- Mamata S.Kalas. "Real Time Detection and Tracking Using OpenCV" International Journal of Soft Computing and Artificial Intelligence (2014): 2321-404X. Web. May 2014.
- Vijayalaxmi, K.Anjali, B.Srujana, and P. Rohith Kumar. "Object Detection and Tracking Using Image Processing"; Global Journal of Advanced Engineering Technologies, Special Issue (CTCNSF-2014).
- Naveen Kumar, Pradeepa Katta, Madhuri Basa, Durga Prasad Lola, P.S.Suhasini M.S, P.Venkata Ganapathi. "Open CV Implementation of Object Recognition Using Artificial Neural Networks." International Journal of Computational Engineering Research Vol (05) Issue (02). (2015): 2250-3005. Web. February 2015.
- Okunade Oluwasogo Adekunle and Osunade Oluwaseyitan. "Emergency Vehicle Priority Preference at Multiple Wireless Network Sensors (MWNS) Intelligent Road Traffic Signal Control System"; IOSR Journal of Computer Engineering, Volume (16) Issue (4), July-August 2014.
- Panth Shah and Tithi Vyas. "Interfacing of Matlab with Arduino for Object Detection Algorithm Implementation using Serial Communication"; International Journal of Engineering Research & Technology, Volume (13) Issue (10), October 2014.
- Farheena Shaikh and Dr.Prof.M.B.Chandak. "An Approach Traffic Management System Using Density Calculation and Emergency Vehicle Alert"; IOSR Journal of Computer Science, 2014.
- Sarika B. Kale and Prof. Gajanan P.Dhok. "Embedded System for Intelligent Ambulance and Traffic Control Management"; International Journal of Computer and Electronics Research, Volume (2) Issue (2), April 2013.
- Sommerville; "Waterfall Method"; Software Engineering, Page (9), 2011