

PENDETEKSI PELANGGAR RAMBU LARANGAN PARKIR DAN BERHENTI DI KAWASAN TELKOM UNIVERSITY

Detector of Traffic Violator of No Parking and Stop Signs Over Telkom University Area

Wildan Assegaf Nurza ¹, Gita Indah Hapsari, S.T., M.T. ², Muhammad Rizky Alfarisi, S.ST., M.T. ³

^{1,2,3}Program Studi D3 Teknologi Komputer, Universitas Telkom

¹wildanassegafnurza@gmail.com, ²gitaindahhapsari@tass.telkomuniversity.ac.id,

³mrizqy@tass.telkomuniversity.ac.id

Abstrak : Kemacetan yang terjadi di area Telkom University menjadi hal yang harus diperhatikan. Berbagai solusi telah dilakukan salah satunya yaitu pemasangan rambu larangan parkir dan berhenti hingga patroli keliling yang dilakukan oleh petugas. Namun, tetap saja beberapa pengguna kendaraan khususnya mobil melakukan pelanggaran yaitu parkir atau berhenti sembarangan tanpa memperhatikan rambu larangan tersebut. Sistem pendeteksi ini menggunakan metode pengolahan citra digital yang berfungsi untuk mendeteksi objek mobil yang berada di area rambu larangan. Apabila mobil terdeteksi dan berhenti melebihi waktu yang ditentukan maka alat akan memberikan peringatan melalui speaker serta mengirim pesan notifikasi kepada petugas. Dengan adanya alat ini maka petugas tidak perlu lagi melakukan patroli keliling dan pelanggar langsung mendapat peringatan melalui speaker yang berada area rambu larangan.

Kata Kunci: kendaraan, kamera, pengolahan citra

Abstract : A traffic jam that happens around Telkom University area is becoming a concern. Many solutions have been conducted to solve this phenomenon such as installing no parking and don't stop signs, also running a routine patrol done by officers. However, some drivers still violate the no -parking signs. This detector system uses an image processing program that functionates to detect cars that are on prohibited parking areas. Once a vehicle is detected violate the signs and pass over the parking-limit time, the system will send a warning through speakers and sending signals to the officers. With this security system, officers do not necessarily do a routine patrol around the campus and offenders can be warned immediately through the located speakers.

Keywords: vehicles, cameras, image processing.

1. Pendahuluan

Rambu lalu lintas menjadi suatu hal yang harus diperhatikan oleh pengguna jalan. Saat ini masih banyak pengguna jalan khususnya pengemudi kendaraan yang tidak memperhatikan rambu lalu lintas. Menurut UU No 22 Tahun 2009 Pasal 1 Ayat 17, Rambu Lalu Lintas adalah bagian perlengkapan Jalan yang berupa lambang, huruf, angka, kalimat, dan/atau perpaduan yang berfungsi sebagai peringatan, larangan, perintah, atau petunjuk bagi Pengguna Jalan.

Dalam kawasan Telkom University terdapat berbagai rambu lalu lintas, salah satunya yaitu rambu larangan parkir dan berhenti. Rambu larangan parkir dan berhenti memiliki ketentuan sendiri, yaitu memiliki jarak sampai 15meter dari tempat pemasangan rambu menurut arah lalu lintas, kecuali dinyatakan lain

dengan papan tambahan. Sering kali pengguna jalan di kawasan Telkom University tidak memperhatikan rambu larangan parkir dan berhenti yang sudah dipasang sehingga dapat menyebabkan kemacetan. Disamping itu rambu larangan parkir yang tidak selalu diawasi oleh petugas keamanan membuat pelanggar menjadi kurang sadar akan rambu larangan tersebut.

Melihat permasalahan diatas maka perlu adanya inovasi untuk mengamati dan mengawasi area rambu larangan parkir dan berhenti. Maka dari itu penulis mengusulkan ide mengenai alat pendeteksi pelanggar rambu larangan parkir dan berhenti di kawasan Telkom University. Dengan menerapkan pengolahan citra untuk mendeteksi kendaraan yang melanggar rambu larangan parkir dan berhenti.

2. Tinjauan Pustaka

2.1. Dasar Teori

Menurut [1] proses *object tracking* bertujuan untuk mendeteksi posisi objek yang bergerak dalam video yang ditangkap kamera. *Object tracking* ini menggunakan pengambilan ciri-ciri unik dari data yang diolah. Yaitu menggunakan perhitungan tinggi, lebar pada objek yang diproses, sehingga dengan perhitungan tersebut

dapat menentukan bahwa objek tersebut adalah kendaraan.

2.2. Raspberry Pi 3

Raspberry Pi3 merupakan mini PC yang dibuat oleh Raspberry Pi Foundation yang diluncurkan pada Maret 2018. Mini PC ini dapat digunakan untuk mempelajari coding dan membangun proyek elektronik, dan berbagai hal lain yang

dapat dilakukan oleh Desktop PC biasa. Seperti pengolahan data, browsing internet, bermain game, serta memutar video dalam definisi tinggi [2].



Gambar 2. 1 Raspberry Pi 3 Model B

2.3. HD Webcam Logitech C525

HD Webcam Logitech C525 merupakan produk kamera *portable* dari Logitech. Kamera ini memiliki fitur *autofocus* dan dapat diputar hingga 360 derajat serta dapat disesuaikan sudut kemiringannya. Dengan ketajaman yang cukup baik webcam ini juga mampu menghasilkan sebuah video dengan resolusi 720p. Kamera ini dihubungkan dengan Raspberry Pi3 menggunakan kabel USB dan jack aux 3.5 mm [3].



Gambar 2. 3 HD Webcam Logitech C525

2.4. Speaker Robot RS170

Pengeras suara atau speaker multimedia ini memiliki kualitas suara yang jernih dan detail. Speaker ini cocok dihubungkan dengan PC, smartphone dan perangkat dengan jack aux 3.5mm. Dengan ukuran sedang dan mudah diatur serta *powered* by USB.



Gambar 2. 4 Speaker Robot RS170

2.5. OpenCV

OpenCV (Open Source Computer Vision Library), adalah sebuah *library open source* yang dikembangkan oleh intel yang fokus untuk menyederhanakan programing terkait citra digital. Library ini dapat digunakan pada C++, Python, dan Java [4]. OpenCV dapat bekerja pada Raspberry Pi 1 (700MHz single core CPU, 256Mb memory). OpenCV juga dapat berjalan selama menggunakan sistem operasi berbasis linux maupun windows yang mendukung. Untuk mendapat proses pengolahan citra yang baik minimal menggunakan hardware dengan RAM minimal 1GB.



Gambar 2. 2 OpenCV

2.6. Python

Python adalah bahasa pemrograman yang mudah digunakan dan dapat diimplementasikan dengan Raspberry Pi untuk membuat proyek. Syntax pada Python sangat bersih, dengan penekanan pada pembacaan dan menggunakan standarisasi kata kunci bahasa inggris. Python disini digunakan sebagai bahasa pemrograman untuk membuat program pendeteksian objek pada sistem yang dibuat [5]. Bahasa pemrograman ini dipilih karena sudah umum digunakan pada Raspberry Pi. Dalam Raspberry Pi, penulisan serta menjalankan program dengan bahasa python ini dapat menggunakan aplikasi Thonny.



Gambar 2. 5 Python

2.7. Pengolahan Citra

2.7.1. Citra

Citra atau gambar merupakan kombinasi antara titik, garis, bidang dan warna untuk menciptakan suatu imitasi dari objek. Citra terdiri dari kumpulan piksel dalam sejumlah baris dan kolom, atau dapat disebut juga citra adalah matriks piksel. Piksel sendiri yaitu elemen terkecil yang menyusun suatu citra.

2.7.2. Konversi Warna

Konversi warna disini digunakan untuk mengubah warna citra asli yang

ditangkap kamera yaitu citraRGB menjadi citra abu-abu. Konversi warna dilakukan agar dapat diproses ke metode selanjutnya yaitu *gaussian blur* serta *threshold*.

2.7.3. Gaussian Blur

Gaussian blur merupakan suatu *filter blur* dengan menempatkan warna transisi yang signifikan terhadap suatu gambar. Filter ini menciptakan efek *autofocus* untuk mengurangi detail serta memberi efek seperti kabut.

2.7.4. Erode Operation

Erode operation yaitu salah satu jenis transformasi morfologis yang digunakan untuk mengubah struktur bentuk suatu objek yang terdapat pada citra. *Erode operation* ini mengurangi atau melakukan penyempitan terhadap suatu struktur dalam citra.

2.7.5. Threshold

Threshold adalah proses mengubah suatu citra berskala keabuan menjadi citra biner atau hitam putih. Sehingga dapat diketahui antara objek dan *background* dari citra secara jelas.

2.7.6. ContourArea

ContourArea ini digunakan untuk memperoleh luas yang menyatakan jumlah piksel dalam suatu objek.

2.7.7. BoundingRect

BoundingRect disini merupakan salah satu jenis *BoundingBox* atau kotak pembatas terkecil yang dapat melingkupi sebuah objek. *BoundingRect* yaitu kotak yang berorientasi pada citra milik suatu objek yang dapat diperoleh dengan memanggil `cv2.boundingRect()`.

3. Analisis dan Perancangan Sistem

3.1. Gambaran Sistem Saat Ini

Pada gambaran sistem saat ini dilihat berdasarkan pengamatan di lapangan. Penindakan pelanggaran rambu larangan parkir dan berhenti di kawasan telkom university masih menggunakan sistem patroli keliling oleh petugas. Sehingga penanganan yang dilakukan hanya bergantung pada sistem patroli keliling yang hanya dilakukan pada waktu tertentu saja.

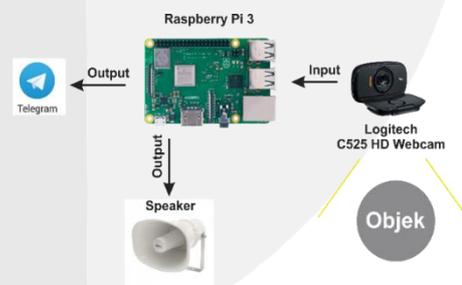
3.2. Analisis Kebutuhan Sistem

Berdasarkan sistem yang akan dibuat dalam Proyek Akhir ini, maka dibutuhkan beberapa alat berdasarkan fungsionalitas dan non-fungsionalitas, yaitu:

1. Kebutuhan Fungsionalitas
 - a. Adaptor 9 Volt sebagai catu daya utama sistem.
 - b. *Raspberry Pi* sebagai mikrokontroler
 - c. *Webcam Camera* untuk menangkap gambar area rambu larangan parkir dan berhenti
 - d. *Speaker* untuk memberikan peringatan kepada pelanggar
 - e. *TelegramBot* untuk mengirim pesan notifikasi kepada petugas.
2. Kebutuhan Non-Fungsionalitas
 - a. Perangkat keras pada sistem yang akan dirancang terdiri dari:
 - 1) *Raspberry Pi 3 Model B* dengan spesifikasi berikut
 - a) Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
 - b) 1 GB RAM
 - c) BCM43438 Wireless LAN
 - d) 4 USB 2.0 ports
 - e) Jack aux 3.5mm
 - f) Micro SD port
 - 2) HD Webcam Logitech C525
 - a) Resolusi maksimal : 720p/30fps
 - b) Build-in autofocus camera
 - c) 1 Microfon omni-directional
 - d) Hi-Speed USB 2.0 certified
 - 3) Speaker Robot RS170
 - a) 5 Volt USB powered
 - b) 2.1 sound channel desktop speaker
 - c) Support jack aux 3.5mm
 - b. Perangkat lunak pada sistem yang akan dirancang terdiri dari:
 - 1) Python IDE
 - 2) OpenCV

3.3. Gambaran Sistem Usulan

Sistem ini dirancang dan di implementasikan pada rambu larangan parkir dan berhenti di kawasan Telkom University. Fungsi secara umum sistem ini adalah sebagai pendeteksi pelanggaran dengan *input* melalui kamera, pengolahan citra oleh raspberry pi 3 dan *output* berupa peringatan suara serta pesan notifikasi.

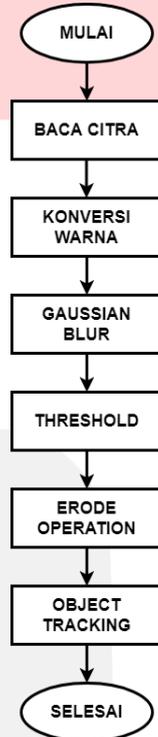


Gambar 3. 1 Gambaran Sistem Usulan

3.4. Blok Diagram

3.4.1. Pengolahan Citra

Pada Proyek Akhir ini dirancang sistem pengolahan citra yang dapat mendeteksi suatu kendaraan yang tertangkap kamera. Berikut adalah diagram alir pengolahan citra untuk pendeteksi kendaraan.



Gambar 3. 2 Diagram Alir Pengolahan Citra

3.4.1.1. Pembacaan Citra

Proses ini adalah proses awal pengolahan citra dengan input berupa video kondisi area rambu larangan yang ditangkap oleh kamera.



Gambar 3. 3 Pembacaan Citra

3.4.1.2. Konversi Warna

Proses konversi warna ini menggunakan syntax cv2.Color(citraRGB, cv2.COLOR_BGR2GRAY). Dengan citraRGB merupakan

citra asli yang ditangkap kamera dan cv2.COLOR_BGR2GRAY adalah hasil konversi citra menjadi berskala keabu-abuan.



Gambar 3. 4 Konversi Warna

3.4.1.3. Gaussian Blur

Pada proses ini citra yang ditangkap diberi efek kabut atau blur untuk mengurangi detail citra, dengan menggunakan metode konversi citra yang sebelumnya dilakukan.



Gambar 3. 5 Gaussian Blur

3.4.1.4. Threshold

Proses *threshold* disini adalah mengkonversi citra yang sebelumnya diberi efek kabut dan berskala keabu-abuan menjadi citra biner atau hitam putih. Sehingga dapat membedakan antara objek dan *background* dari citra yang proses. Objek yang ditandai berupa warna putih sedangkan background berwarna hitam. Citra hitam putih ditunjukkan pada gambar berikut



Gambar 3. 6 Citra sebelum di Threshold



Gambar 3. 7 Citra Sesudah di Threshold

3.4.1.5. Erode Operation

Pada proses *erode* ini dilakukan pengikisan atau memperkecil tepi objek warna putih yang telah dilakukan pada proses sebelumnya yaitu *threshold*. Berikut gambar citra sesudah diberi *erode operation*.



Gambar 3. 8 Erode Operation

4. Implementasi dan Pengujian

4.1. Implementasi

4.1.1. Penginstalan OpenCV

OpenCV yang di install adalah versi 4.1.2

1. Download file bash untuk menginstall opencv

```
git clone
https://github.com/opencv/opencv_contrib.git
```

2. Buat directory baru

```
mkdir -p
~/opencv_build/opencv/build
cd ~/opencv_build/opencv/build
```

3. Compile file

```
cmake -D
CMAKE_BUILD_TYPE=RELEASE
```

4. Install

```
make -j4
make install
```

5. Tunggu hingga proses penginstalan selesai

4.1.2. Konfigurasi Telegram Bot

1. Install package request pada raspberry pi

```
sudo pip install request
```

2. Buka aplikasi telegram, dan tambahkan akun BotFather

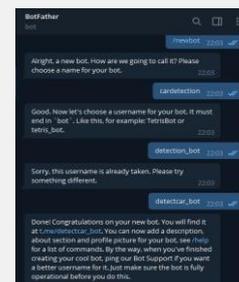
3.4.1.6. Object Tracking

Object tracking disini bertujuan untuk mendeteksi posisi objek yang bergerak dalam citra yang ditangkap setelah diberi metode-metode sebelumnya. *Object tracking* dilakukan dengan cara menghitung luas citra berwarna putih dari citra yang sudah dilakukan *erode operation*. Perhitungan luas ini menggunakan metode *contour area*, apabila *contour area* melebihi batas X yang ditentukan maka objek dapat diklasifikasikan sebagai kendaraan. Selanjutnya dilakukan *bounding rect* yaitu pemberian kotak pembatas berorientasi citra yang melebihi *contour area* yang sudah ditentukan, sehingga objek yang teridentifikasi mendapat sebuah kotak pembatas dan menegaskan bahwa objek merupakan kendaraan



Gambar 4. 1 Penambahan Akun BotFather

3. Buat username untuk bot yang akan digunakan.



Gambar 4. 2 Pembuatan Username Telegram Bot

Dalam pembuatan username ini harus menggunakan username yang belum pernah digunakan oleh pengguna BotFather lain. Kemudian gunakan token yang didapat untuk mengakses API yang nantinya akan ditaruh di program dengan nama `bot_token`.

4. Dapatkan Chat ID

Cara mendapatkan chat id yaitu dengan membuka tautan <https://api.telegram.org/bot<TOKEN>/getUpdates>. Pada bagian <TOKEN> isikan token yang telah didapatkan sebelumnya dari BotFather.

5. Konfigurasi telegram bot telah selesai

4.1.3. Konfigurasi Sound

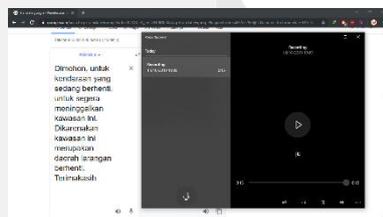
1. Install package pygame
sudo pip install pygame
2. Tunggu hingga proses install selesai
3. Perekaman suara

Perekaman suara ini akan digunakan sebagai peringatan suara untuk kendaraan yang terdeteksi melanggar rambu larangan. Perekaman suara disini menggunakan aplikasi Voice Recorder Windows dan output suara dari google translate. Terdapat 2 kalimat yang digunakan dalam perekaman ini yaitu untuk rambu larangan parkir dan larangan berhenti

Rambu larangan parkir, “Dimohon untuk kendaraan yang sedang berhenti untuk segera meninggalkan kawasan ini. Dikarenakan kawasan ini merupakan daerah larangan parkir. Terimakasih”.

Rambu larangan berhenti, “Dimohon untuk kendaraan yang sedang berhenti untuk segera meninggalkan kawasan ini. Dikarenakan kawasan ini merupakan daerah larangan berhenti. Terimakasih”.

Berikut adalah proses perekaman suara menggunakan aplikasi Voice Recorder Windows dan google translate

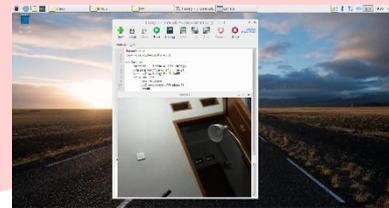


Gambar 4. 3 Proses Perekaman Suara

4.1.4. Konfigurasi Kamera

Penggunaan kamera disini menggunakan pemanggilan melalui program python. Dikarenakan alat pendeteksi ini hanya menggunakan bahasa pemrograman python untuk pendeteksian kendaraan menggunakan library OpenCV. Konfigurasi kamera ini dilakukan dengan memasang kabel usb

pada kamera ke port usb pada Raspberry Pi. Raspberry Pi disini hanya dapat memproses video atau kamera dengan resolusi 720x480. Selanjutnya langsung dilakukan pengecekan menggunakan program untuk mengetahui apakah kamera dapat berfungsi dengan baik.



Gambar 4. 4 Konfigurasi Kamera

4.1.5. Pembuatan Program Pengolahan Citra

Program pengolahan citra disini dibuat dengan bahasa python. Program ini digunakan untuk mendeteksi objek pada citra yang tertangkap kamera. Dalam program ini dilakukan beberapa metode yang sudah dijelaskan sebelumnya yaitu *Gaussian Blur*, *Threshold*, *Erode Operation*, *ContourArea* dan *BoundingRect*. Berikut gambar source code dari program pengolahan citra yang telah dibuat.

```

import cv2

def default = 0
counter = 0

blur_kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (1, 3))
cap = cv2.VideoCapture(0)

for i in range(10):
    success, frame = cap.read()
    if not success:
        print()
        break

img_background = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
img_background = cv2.bilateralFilter(img_background, 9)

success, frame = cap.read()

while success:
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    img_frame = cv2.cvtColor(gray_frame, cv2.COLOR_GRAY2RGB)
    diff = cv2.absdiff(img_frame, img_background)
    diff = cv2.threshold(diff, 0, 255, cv2.THRESH_BINARY)
    contours, hierarchy = cv2.findContours(diff, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    for i in contours:
        x, y, h, w = cv2.boundingRect(i)
        cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 255, 0), 2)
    cv2.imshow('frame', frame)
    cv2.waitKey(1)
    success, frame = cap.read()
    print()
    counter += 1
    if counter == 10:
        break
    
```

Gambar 4. 5 Pembuatan Program Pengolahan Citra

4.1.6. Pembuatan Program Telegram Bot

Program telegram bot disini digunakan untuk mengirim pesan notifikasi kepada petugas. Telegram bot disini membutuhkan chat id dan token dari telegram yang dituju yaitu petugas itu sendiri. Source code dari program telegram bot ini dapat dilihat pada gambar berikut.

```

import requests

def telegram_bot_sendText(bot_message):
    bot_token = "*****"
    chat_id = "*****"
    send_text = "https://api.telegram.org/bot" + bot_token + "/sendMessage?chat_id=" + chat_id + "&parse_mode=MarkdownV1" + bot_message
    response = requests.get(send_text)
    return response.json()

text = telegram_bot_sendText("Coba kirim")
print(text)
    
```

Gambar 4. 6 Pembuatan Program Telegram Bot

4.1.7. Pembuatan Program Peringatan Suara

Program peringatan suara ini bertujuan untuk menjalankan suara yang telah di rekam sebelumnya. Program ini menggunakan package pygame yang mudah digunakan, yaitu dengan langsung memanggil file perekaman suara. Program peringatan suara ini dapat dilihat pada gambar berikut.

```

1 import pygame
2
3 pygame.mixer.init()
4 s = pygame.mixer.Sound("Recording.wav")
5 s.play()

```

Gambar 4. 7 Pembuatan Program Peringatan Suara

4.1.8. Parameter Klasifikasi Objek

Parameter yang digunakan dalam klasifikasi objek adalah luas kontur, tinggi dan lebar. Parameter ini nantinya menjadi dasar untuk menentukan bahwa objek yang tertangkap kamera adalah kendaraan mobil. Data parameter objek dapat dilihat pada Tabel 4.11 di Lampiran. Luas kontur diperoleh dengan memanggil `cv2.contourArea()`. Luas kontur ini diambil dari citra yang telah diproses oleh metode sebelumnya, dan diambil luas dari citra berwarna putih. Kotak pembatas memiliki argumen berupa kontur dan nilai baliknya berupa x , y , w , dan h dengan (x, y) menyatakan koordinat pojok kiri-atas kotak pembatas. Sedangkan w menyatakan lebar kotak pembatas dan h menyatakan tinggi kotak pembatas. Sehingga parameter tinggi dan lebar disini diambil menggunakan metode *bounding rect* atau kotak pembatas melalui argumen w dan h .

Dari data diatas dapat diambil kesimpulan bahwa kendaraan yang akan dideteksi yaitu mobil dapat diambil nilai tengah luas kontur, nilai minimum tinggi mobil, dan nilai maksimum lebar motor. Diambilnya nilai minimum melalui tinggi mobil karena nilai tinggi dari motor yang terdeteksi terdapat pengendara sehingga nilai tinggi motor kadang melebihi dari tinggi mobil yang diproses oleh program. Serta nilai lebar minimum diambil melalui lebar minimum motor, karena jarak nilai maksimum antara mobil dan motor hanya 20. Untuk nilai tengah luas kontur didapat melalui rumus berikut.

$$\text{nilai tengah luas kontur} = \frac{\text{nilai max kontur motor} + \text{nilai min kontur mobil}}{2}$$

Dari rumus dan analisis diatas dapat diambil nilai untuk mendeteksi mobil yaitu dengan luas kontur lebih dari 94177, tinggi lebih dari 299 dan lebar lebih dari 480. Sehingga apabila program mendeteksi objek dan melebihi nilai-nilai yang telah ditentukan maka objek dapat diklasifikasikan sebagai mobil, sebaliknya jika objek yang tertangkap tidak memenuhi nilai-nilai diatas maka objek tidak terdeteksi atau tidak diklasifikasikan sebagai mobil.

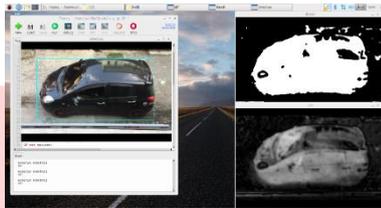
4.2. Pengujian

4.2.1. Pengujian Pengolahan Citra

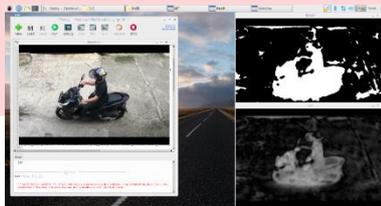
Pengujian program pengolahan citra ini dilakukan menggunakan video yang telah di rekam sebelumnya. Dalam pengambilan video alat diletakkan dengan ketinggian 4meter diatas rambu larangan dan sudut kemiringan menyesuaikan area rambu larangan yang sudah ditentukan. Terdapat 3 video yang diuji dengan objek yang berbeda yaitu objek mobil, motor, dan manusia. Sehingga terdapat 3 kali pengujian dengan objek yang berbeda, dan menghasilkan 3 parameter yaitu luas kontur, tinggi *bounding rect*, dan lebar *bounding rect*. Dalam pengujian ini juga menggunakan nilai-nilai minimum yang didapat dari klasifikasi objek sebelumnya, yaitu luas kontur lebih dari 94177, tinggi lebih dari 299 dan lebar lebih dari 480. Berikut adalah hasil pengujian terhadap ketiga objek tersebut.

No	Objek	Luas Kontur	Tinggi Kotak Pembatas	Lebar Kotak Pembatas
1	Mobil	134571	301	573
2	Motor	58493	280	348
3	Manusia	11356	218	114

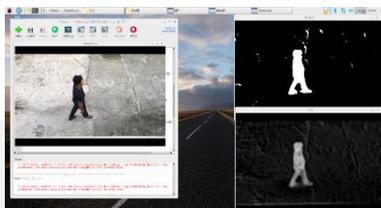
Hasil pengujian diatas menyatakan bahwa nilai minimum yang telah ditetapkan sebelumnya dapat bekerja dalam 3 objek yang diuji. Sehingga pendeteksian objek mobil dapat dinyatakan akurat. Berikut adalah pengujian terhadap ketiga objek



Gambar 4. 8 Pengujian dengan Objek Mobil



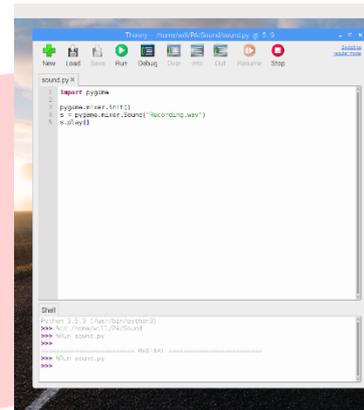
Gambar 4. 9 Pengujian dengan Objek Motor



Gambar 4. 10 Pengujian dengan Objek Manusia

4.2.2. Pengujian Peringatan Suara

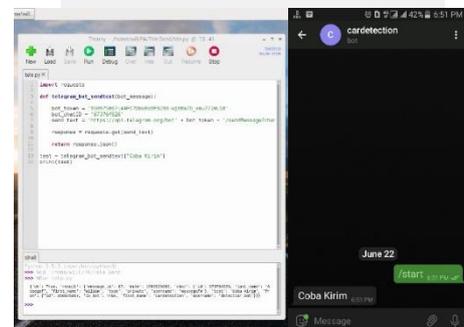
Pengujian peringatan suara ini dilakukan untuk mengetahui apakah output suara dapat keluar menggunakan program yang telah dibuat. Dalam pengujian ini menggunakan suara yang telah direkam sebelumnya. Pengujian suara ini pada raspberry pi harus menggunakan file suara dengan ekstensi (.wav). Hasil pengujian suara ini adalah output suara dapat keluar dengan baik dan tanpa *delay*. Pengujian peringatan suara ini dapat dilihat pada gambar berikut.



Gambar 4. 11 Pengujian Peringatan Suara

4.2.3. Pengujian Telegram Bot

Pengujian ini dilakukan dengan langsung menjalankan program untuk mengirim pesan notifikasi. Pengujian ini dilakukan untuk mengetahui apakah program dapat mengirim pesan tanpa kendala kepada id telegram yang dituju. Dalam pengujian ini dilakukan pengiriman pesan berupa teks “Coba Kirim”. Pengujian ini juga memerlukan koneksi internet pada raspberry pi dan *smartphone* dengan aplikasi telegram yang dituju. Pengujian pengiriman pesan ini dapat dilihat pada gambar berikut.



Gambar 4. 12 Pengujian Pengiriman Pesan Telegram Bot

5. Kesimpulan

5.1. Kesimpulan

Dari hasil pembangunan pendeteksi pelanggaran rambu larangan parkir dan berhenti dapat disimpulkan bahwa:

- Dari hasil pengujian pengolahan citra, sistem dapat menangkap gambar atau citra melalui kamera. Selain itu, sistem dapat mendeteksi objek mobil melalui nilai dari klasifikasi objek yang telah ditentukan.
- Peringatan secara langsung kepada pelanggaran dapat dikeluarkan dalam bentuk suara melalui speaker yang terpasang pada rambu larangan.

- Pengiriman pesan notifikasi dapat terkirim antara sistem dengan id telegram yang dituju yaitu petugas, sehingga petugas mendapatkan pesan notifikasi apabila terdapat pelanggaran tanpa perlu melakukan patrol keliling. Dalam pengiriman pesan ini, pesan dapat dikirim secara realtime apabila tidak ada kendala pada jaringan internet.

5.2. Saran

Adapun saran dari penulis untuk sistem pendeteksi pelanggar rambu larangan parkir dan berhenti ini, yaitu

- Penggunaan kamera dengan yang spesifikasi yang lebih baik dapat meningkatkan akurasi pendeteksian objek.
- Penempatan kamera atau alat dapat ditinggikan lagi, sehingga penangkapan gambar pada area rambu larangan lebih luas.

6. Daftar Pustaka

- [1] C. Z. PRATIWI, "REALISASI SISTEM DETEKSI KENDARAAN DENGAN MENGGUNAKAN PENGOLAHAN VIDEO." Universitas Telkom, S1 Teknik Telekomunikasi, 2015.
- [2] R. P. Foundation, "FAQs - Raspberry Pi Documentation." [Online]. Available: <https://www.raspberrypi.org/documentation/faqs/#introduction>. [Accessed: 05-Apr-2019].
- [3] "HD Webcam C525." [Online]. Available: <https://www.logitech.com/id-id/product/hd-webcam-c525#specification-tabular>. [Accessed: 23-Jun-2020].
- [4] Binus Malang, "Introduction to Open CV." [Online]. Available: <http://binus.ac.id/malang/2017/10/introduction-to-open-cv/>. [Accessed: 05-Apr-2019].
- [5] R. P. Foundation, "Python - Raspberry Pi Documentation." [Online]. Available: <https://www.raspberrypi.org/documentation/usage/python/README.md>. [Accessed: 26-Apr-2019].

7. Lampiran

Tabel 4. 1 Data Parameter Klasifikasi Objek

No	Objek	Warna	Luas Kontur	Tinggi Kotak Pembatas	Lebar Kotak Pembatas
1	Orang 1	-	11829	224	141
2	Orang 2	-	12203	220	108
3	Orang 3	-	12111	237	87
4	Orang 4	-	10966	223	130
5	Orang 5	-	11144	227	113
6	Sepeda 1	Hitam	14147	288	269
7	Sepeda 2	Hitam	26931	200	290
8	Sepeda 3	Biru	39401	295	332
9	Motor Roda 3	Biru	67055	260	554
10	Motor Roda 3	Hitam	39126	202	396
11	Motor 1	Hitam	31442	218	281
12	Motor 2	Abu	30948	268	275
13	Motor 3	Hitam	54690	352	295
14	Motor 4	Kuning	50652	387	303
15	Motor 5	Merah	38516	361	264
16	Motor 6	Merah	78892	379	387
17	Motor 7	Hitam	60106	346	471
18	Motor 8	Biru	32364	358	381
19	Motor 9	Hijau	28490	248	358
20	Motor 10	Merah	36822	249	480
21	Mobil 1	Hitam	131371	310	606
22	Mobil 2	Merah	147004	394	563
23	Mobil 3	Abu	122622	302	640
24	Mobil 4	Putih	162288	331	500
25	Mobil 5	Hitam	137980	299	562
26	Mobil 6	Cokelat	1377439	438	680
27	Mobil 7	Hijau	164021	338	640
28	Mobil 8	Silver	143871	410	610
29	Mobil 9	Orange	109463	346	545
30	Mobil 10	Kuning	121314	341	595
31	Mobil 11	Putih	114316	407	531
32	Mobil 12	Cream	152551	301	666
33	Mobil 13	Hijau	121692	304	571
34	Mobil 14	Merah	167562	360	622
35	Mobil 15	Biru	121555	343	604