
Program Aplikasi Pendeteksi Masker Dengan Menggunakan Algoritma Haarcascade

Ai Ilah Warnilah^{1*}, Herlan Sutisna², Aldi Jaya Mulyana³, Feni Siti Nuraeni⁴,
Thomas Aninditya Widiyanto⁵

^{1,2,3,4,5} Sistem Informasi Kampus Kota Tasikmalaya, Universitas Bina Sarana Informatika
Tasikmalaya, Indonesia

* Corresponding Author. E-mail: ai.aiw@bsi.ac.id

Abstract

The Covid-19 pandemic has caused millions of people to be infected with this virus and died. During this new normal era, 5M has been implemented (wearing masks, washing hands, maintaining distance, reducing mobility, avoiding crowds) when leaving the house and incessant vaccinations. The implementation of 5M is applied to face-to-face lectures, one of which is the use of masks so that it can run well, then a mask detection application is made using the haarcascade algorithm in the python programming language. This application uses a user interface display using the tkinter module, the opencv module, datetime, winsound and numpy which are used to detect masked and unmasked objects, and the web browser module. The results of this study are able to use login and register on the application, can run the system to detect masks and an alarm will sound if you don't use a mask then the photo will be saved. There is also an information menu combined with a tkinter and web browser that contains information on Covid-19 including Understanding, Covid-19 data info, vaccination centers, and hospitals in tasikmalaya city.

Keywords: haarcascade; mask detector; opencv; python; and tkinter

Abstrak

Pandemi Covid-19 telah menyebabkan jutaan manusia terinfeksi virus ini dan meninggal dunia. Pada saat era new normal ini telah diberlakukannya 5M (Memakai masker, Mencuci tangan, Menjaga Jarak, Mengurangi Mobilitas, Menghindari Kerumunan) pada saat keluar rumah serta gencarnya vaksinasi. Pemberlakuannya 5M diterapkan pada perkuliahan tatap muka, salah satunya adalah penggunaan masker agar dapat berjalan dengan baik, maka dibuatlah aplikasi pendeteksi masker menggunakan algoritma haarcascade pada bahasa pemrograman python. Aplikasi ini menggunakan tampilan user interface menggunakan modul tkinter, modul opencv, datetime, winsound dan numpy yang digunakan untuk mendeteksi objek bermasker dan tidak bermasker, dan modul webbrowser. Hasil dari penelitian ini adalah dapat menggunakan login dan register pada aplikasi, dapat menjalankan sistem mendeteksi masker dan alarm akan berbunyi jika tidak menggunakan masker maka fotonya akan tersimpan. Serta ada menu informasi yang di kombinasikan tkinter dan webbrowser yang memuat informasi Covid-19 mencakup Pengertian, Info data Covid-19, Sentra vaksinasi, dan Rumah sakit yang ada di kota Tasikmalaya.

Kata kunci: *haarcascade*; *opencv*; Pendeteksi masker; *python*; dan *tkinter*.

1. Introduction

Sejak Desember 2019, dunia digemparkan dengan munculnya COVID-19 yang merupakan penyakit pernafasan akut. Penyakit ini disebabkan oleh virus corona yang bernama *severe acute respiratory syndrome coronavirus 2* atau disingkat *SARS CoV-2*. Hingga 17 Juni 2020 *World Health Organization* (WHO) menjadikan virus ini menjadi pandemic cakupannya secara global. COVID-19 melanda Indonesia sejak Maret 2020. Jumlah kasus baru di Indonesia setiap harinya masih terjadi peningkatan. Pemerintah selaku pembuat sudah memerlakukan banyak perubahan pekerjaan di kantor menjadi *work from home*, melakukan *lock down* secara ketat pada 6 bulan pertama, PSBB, dan melakukan pengecekan bagi orang yang mau pergi dari dalam dan luar negeri dengan test PCR, *Swab* dan *genose*.

Indonesia sekarang ini berada di *era new normal* dari pandemi *virus covid-19* dengan adanya PPKM penerapan 5M (Memakai masker, Mencuci tangan, Menjaga Jarak, Mengurangi Mobilitas, Menghindari Kerumunan) pada saat beraktivitas diluar rumah. Penularan yang sangat mudah menjadikan virus ini menjadi ancaman bagi semua warga Indonesia dengan pedoman 5M dan melakukan vaksinasi untuk membentengi dari virus corona ini namun masih ada orang yang tidak menerapkan 5M Khususnya penggunaan masker karena mungkin kurangnya edukasi dan sosialisasi khususnya penerapan protocol kesehatan pada masa pembelajaran perkuliahan tatap muka maka dari itu kami membuat penelitian mengenai aplikasi pendeteksi

masker dengan algoritma *haarcascade* pada bahasa pemrograman *python* dan informasi mengenai covid-19 yang ada di kota Tasikmalaya..

Penelitian pendeteksi masker ini pernah dilakukan sebelumnya dengan metode algoritma *haarcascade* pada bahasa pemrograman *python*. (Aprilian Anarki et al., 2021) Namun disini menggunakan algoritma *haarcascade* wajah saja dengan dikombinasikan dengan suara peringatan sama dengan jurnal yang dikemukakan oleh (Luthfillah Ahmad et al., 2021)

Hasil penelitian dari (Thariq & Bakti, 2021) menggunakan algoritma *haarcascade* wajah saja dengan dikombinasikan dengan suara peringatan serta adanya dataset yang dibagi menjadi 2 variabel memakai masker dan tidak memakai masker maka wajahnya akan terdeteksi serta ada juga referensi menggunakan *raspberrypi* dalam penggunaan pendeteksi masker pada pintu.(Lambacing & Ferdiansyah, 2020), (Abdul et al., 2020), (Tiwikrama et al., 2021), (Radhi et al., 2021), penggunaan *Interet of things* (Irsyad et al., 2021), penggunaan pada *matlab design* (Irsyad et al., 2021) serta ada penggunaannya pada pendeteksi masker pada pengemudi (Alief et al., 2021) ada juga sistem pendeteksi masker menggunakan CNN (Naufal et al., 2021).

Tujuan dari penelitian ini yaitu untuk menguji modul GUI *tkinter* dan algoritma *haarcascade* pada modul *opencv python*. Dapat digunakan pada fasilitas umum khususnya pada tempat pendidikan dan pada laboratorium medis yang membutuhkan *sterilisasi* pada yang dijadikan sebagai alat pendeteksi masker yang bisa di kombinasikan dengan yang lain seperti perangkat *internet of things*.

2. Materials and Methods

2.1. Perubahan yang diamati

Data yang digunakan dalam penelitian adalah informasi Covid-19 mencakup Pengertian, Info data Covid-19, Sentra vaksinasi, dan Rumah sakit yang ada di kota Tasikmalaya yang di *redirect* oleh modul *webbrowser* akan terhubung ke *website* yang dituju.

Untuk pendeteksi masker menggunakan *open capture vision* dengan video kamera 2 klasifikasi Jika wajah terlihat dan hidung tidak terlihat (memakai masker) masker *on*, dan jika wajah dan hidung terlihat masker *off*. Proses pembacaan objek wajah dan hidung sehingga kamera akan mencocokkan dengan file algoritma *haarcascade* ketika masker *off* disana ada sistem untuk melakukan suara pemberitahuan

Beberapa batasan aspek untuk mendeteksi masker adalah, variasi posisi wajah, bentuk wajah, jarak wajah, dan pencahayaan dalam mendeteksi.

Untuk variasi posisi wajah yang dipraktikkan

- Menghadap kedepan
- Rotasi 30° ke kanan
- Rotasi 30° ke kiri
- Mengangkat dagu 15° ke atas
- Menunduk kepala 15° ke bawah
- Dua orang memakai masker
- Dua orang tidak memakai masker
- Satu orang memakai masker dan satu orang tidak memakai masker

Bentuk wajah sangat mempengaruhi karena pada *haarcascade* ini mendeteksi bagian wajah dan hidung secara umum serta khususnya untuk orang dengan bentuk wajah bulat harus disesuaikan dengan menghadap kedepan.

Jarak wajah yang optimal digunakan adalah 30 cm – 50 cm. dengan pencahayaan

yang cukup baik untuk *indoor* diutamakan untuk ruangan agak gelap menggunakan lampu dan *outdoor* diutamakan dengan ada cahaya matahari yang cukup.

2.2. Metode yang digunakan

Metode ini merupakan proses pengolahan pendeteksi masker menggunakan algoritma *haarcascade* dengan *opencv* pada bahasa pemrograman *python*. Aplikasi ini menggunakan tampilan user interface menggunakan modul *tkinter*, modul *opencv*, *datetime*, *winsound* dan *numpy* yang digunakan untuk mendeteksi objek bermasker dan tidak bermasker, dan modul *webbrowser*.

2.3. Inisialisasi kamera dengan opencv

Open Capture Vision merupakan sebuah *Application Programming Interface library* yang memungkinkan komputer dapat melihat objek seperti manusia atau benda yang mana komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek yang mana objek targetnya pada penelitian ini adalah wajah dan hidung dengan menggunakan algoritma *haarcascade* yang mana kamera laptop sebagai inisialisasi dari pendeteksi masker yang mendeteksi berdasarkan dua pengujian yaitu berdasarkan sudut variasi meliputi Menghadap kedepan, Rotasi 30° ke kanan, Rotasi 30° ke kiri, Mengangkat dagu 15° ke atas, dan Menunduk kepala 15° ke bawah. Serta berdasarkan dua orang objek dengan ketentuan Dua orang memakai masker, Dua orang tidak memakai masker, Serta Satu orang memakai masker dan satu orang tidak memakai masker, catatan harus dengan pencahayaan yang cukup.

2.4. Proses menentukan tampilan gui dengan tkinter

Modul *tkinter* merupakan modul yang dikombinasikan dengan *python* yang menyediakan cara cepat dan mudah membuat

aplikasi dengan tampilan secara gui (*graphic user interface*) yang artinya bentuk tampilan antar muka yang memungkinkan pengguna untuk berinteraksi melalui gambar ikon. Serta cocok digunakan untuk tampilan aplikasi gui pada desktop. Disini kami menggunakan tkinter sebagai tampilan *user interface* pada aplikasi karena untuk memudahkan para pengguna dalam mengakses aplikasi pendeteksi masker ini.

2.5. Proses menentukan haarcascade serta penerapan pada opencv

Haarcascade (Lia Farokhah, 2021) adalah algoritma yang memberikan indikasi secara spesifik pada sebuah gambar atau image yang dikemukakan oleh Viola & Jhon, (Al-Aidid & Pamungkas, 2018). Mengenali obyek berdasarkan nilai sederhana dari fitur tetapi bukan merupakan nilai piksel dari image obyek tersebut. Metode ini memiliki kelebihan yaitu komputasi yang sangat cepat, karena hanya tergantung pada jumlah piksel dalam persegi bukan setiap nilai piksel dari sebuah image Metode ini merupakan metode yang menggunakan statistikal model (*classifier*). Pendekatan untuk mendeteksi objek dalam gambar menggabungkan empat kunci utama yaitu (Luthfillah Ahmad et al., 2021):

- *Haar like feature* adalah fitur yang didasarkan pada *Wavelet* Haar. *Wavelet* Haar adalah gelombang tunggal bujur sangkar (satu interval tinggi dan satu interval rendah). Untuk dua dimensi, satu terang dan satu gelap. Selanjutnya kombinasi-kombinasi kotak yang digunakan untuk pendeteksian objek visual yang lebih baik. Setiap *Haar-like feature* terdiri dari gabungan kotak - kotak hitam dan putih klasifikasi ini dilakukan berdasarkan nilai dari sebuah fitur haar yang ditentukan dengan cara mengurangi rata – rata pixel pada daerah terang serta

adanya perbedaan antara jumlah nilai pixel *grey level* dalam daerah kotak hitam dan putih sehingga dapat menentukan bagian yang menandakan objek yang di deteksi.



Gambar 1 *Haarlike Feature*

- *Integral Image* digunakan untuk menentukan ada atau tidaknya dari ratusan fitur Haar pada sebuah gambar dan pada skala yang berbeda secara efisien. Proses pencarian nilai fitur ini dilakukan secara iteratif mulai dari ujung kiri atas hingga ujung kanan bawah dengan mumpunya integral menambahkan unit kecil secara bersamaan, dalam hal ini unit kecil ini disebut dengan nilai dari piksel. Nilai dari integral pada masing-masing piksel adalah penjumlahan dari semua pixel di atasnya dan di sebelah kirinya. Dimulai dari kiri atas sampai kanan bawah, gambar bisa diintegrasikan sebagai operasi matematika per piksel
- *Adaboost Machine learning* Untuk memilih *fitur Haar* yang dipakai dan untuk mengubah nilai *threshold*, dengan memakai metode ini menggabungkan banyak *classifier* untuk membuat satu *classifier*. Masing-masing *classifier* menetapkan suatu bobot dan gabungan dari bobot inilah yang akan membentuk satu *classifier* yang kuat. Masing-masing filter adalah *AdaBoost classifier* yang terpisah dengan jumlah *weak classifier* yang sedikit & sama
- *Cascade Classifier* adalah adanya klasifikasi bertingkat Klasifikasi pada algoritma ini terdiri dari beberapa tingkatan dan tiap tingkatan mengeluarkan sub

tampilan yang diyakin. Hal ini dilakukan karena lebih mudah untuk menilai objek.

Untuk *training* pada *haarcascade default file* yang mana sudah memuat *haarcascade* wajah dan hidung dengan memiliki beberapa tahapan untuk bisa mendeteksi masker yaitu :

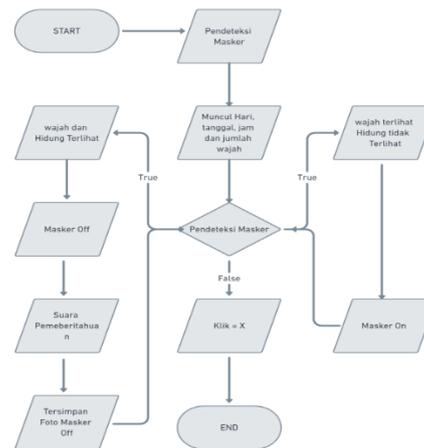
- Menentukan *Haarfeature* yaitu *haarcascade* wajah dan hidung
- Membuat program
- Menjalankan program
- *Opencv* menghitung integral gambar dengan target wajah dan hidung yang mengandung node atau titik dalam target objek
- Melakukan *machine learning* yang mana ini mengidentifikasi apakah objek yang ada pada depan kamera merupakan wajah dan hidung
- Proses *haarcascade* klasifikasi yang mana menentukan hasil yang didapat pada pendeteksi masker
- Pada *opencv* akan ada tampilan apakah masker *on* atau masker *off*

2.6. Perancangan login register pada tkinter

Perancangan aplikasi pendeteksi masker dengan menggunakan tampilan user interface menggunakan modul *tkinter* (Firmansyah, 2021), library *gui* ini yang dapat dikombinasikan *python* menyediakan cara mudah membuat aplikasi dengan bentuk tampilan antar muka untuk berinteraksi melalui ikon atau gambar. Sehingga pengguna menggunakan aplikasi ini dengan mudah yang memuat beberapa fitur diantaranya login dan register (Syarif & Wijanarto, 2015) pada aplikasi dengan penerapan sistem *Object Oriented Programming* pada penggunaannya dengan beberapa menu yang ada yaitu dapat menjalankan sistem mendeteksi masker dan alarm akan berbunyi jika tidak menggunakan masker maka fotonya akan tersimpan dengan cara *import file*

ditempatkan pada *button* pendeteksi masker Serta ada menu informasi yang di kombinasikan dengan *button* pada *tkinter* dan *webbrowser* yang memuat informasi Covid-19 mencakup Pengertian, Info data Covid-19, Sentra vaksinasi, dan lokasi Rumah sakit di kota Tasikmalaya.

2.7. Perancangan pendeteksi masker menggunakan algoritma haarcascade



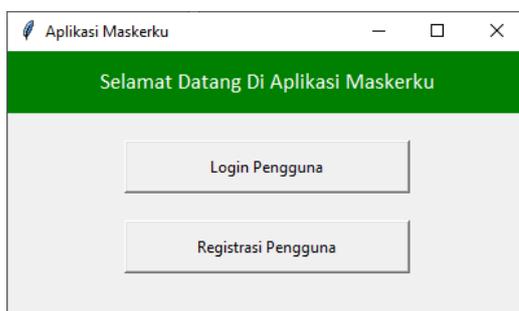
Gambar 2. Flowchart.

Pada Gambar 2 merupakan *flowchart* penggunaan aplikasi pendeteksi masker (Ngodein, 2020) Disini pendeteksi masker digunakan pada *button* *tkinter* lalu munculkan hari, tanggal dan jam dengan menggunakan modul *datetime* membuat perulangan pada pendeteksi masker dengan menerapkan algoritma *haarcascade* dan pengulangan jumlah wajah pada *opencv*. Jika wajah terlihat dan hidung tidak terlihat (memakai masker) maka muncul tampilan text masker *on*, dan jika wajah dan hidung terlihat maka muncul tampilan text masker *off* serta akan muncul suara pemberitahuan tidak memakai masker menggunakan modul *winsound* untuk menjalankan audio dan menyimpan *capture* gambar menggunakan modul *numpy* yang mana untuk menangkap gambar. Untuk keluar dari aplikasi dengan membuat *waitkey* "x" untuk keluar dari aplikasi perulangan pada pendeteksi

3. Results and Discussion

Dalam penelitian aplikasi pendeteksi masker dengan algoritma *haarcascade* ini memiliki sistem login register yang digunakan untuk mengakses aplikasi ini pada Gambar 2 yang mana berguna untuk mengakses aplikasi serta sistemnya sudah terintegrasi dengan penyimpanan data akun pengguna.

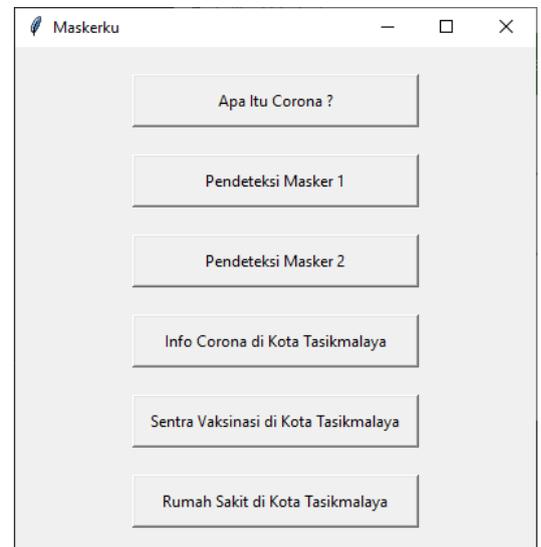
Ketika sudah masuk aplikasi maka akan ditampilkan menu pada aplikasi dengan fungsinya masing – masing pada Gambar 3 disana ada *button* yang bisa mengakses halaman *web* dengan *responsive* yaitu menu Apa itu corona ?, Info corona di kota tasikmalaya, Sentra Vaksinasi di kota tasikmalaya, dan Rumah sakit di kota tasikmalaya.



Gambar 3. Tampilan login dan register.

Pada penelitian ini mengidentifikasi penggunaan masker dengan algoritma *haarcascade* wajah dan hidung melakukan proses pengecekan dengan 2 klasifikasi Jika wajah terlihat dan hidung tidak terlihat (memakai masker) masker *on*, dan jika wajah dan hidung terlihat masker *off*. Proses pembacaan objek wajah dan hidung sehingga kamera akan mencocokkan dengan file algoritma *haarcascade* ketika masker *off* disana ada sistem untuk melakukan suara pemberitahuan menggunakan *winsound* yang mengeksekusi file

berformat *.wav* dan menangkap gambar dengan *numpy* yang ditampilkan di Gambar



Gambar 4. Tampilan menu.

Pengujian deteksi dilakukan menggunakan algoritma *haarcascade* dengan mengambil wajah dan hidung orang melalui kamera laptop dengan klasifikasi masker *on* dan masker *off*. Jika wajah terlihat dan hidung tidak terlihat (memakai masker) maka muncul tampilan text masker *on*, dan jika wajah dan hidung terlihat maka muncul tampilan text masker *off* serta akan muncul suara pemberitahuan tidak memakai masker menggunakan modul *winsound*. Untuk masker *off* kurang *responsive* karena melakukan perulangan dengan kamera. ditampilkan juga jumlah wajah dengan menggunakan *len* pada *haarcascade* wajah serta menampilkan hari, tanggal dan jam menggunakan *datetime* sehingga penggunaannya dapat digunakan secara *realtime*.

```

if mask_on:
    cv2.rectangle(frame, (x,y), (x+w, y+h), (0, 255, 0), 3)
    cv2.putText(frame, 'Masker On', (x,y), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
else:
    cv2.rectangle(frame, (x,y), (x+w, y+h), (0, 0, 255), 3)
    cv2.putText(frame, 'Masker Off', (x,y), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
    print('Image'+str(count)+' saved')
    file = 'D:/pendeteksi/data/gambar'+str(count)+'.jpg' #disesuaikan dengan dengan temp
    cv2.imwrite(file, frame)
    count+=1
    winsound.PlaySound("D:\\pendeteksi\\musik\\Maskeroff.wav", winsound.SND_FILENAME) #dl

hidung = nosecascade.detectMultiScale(rooi_gray, 1.18, 35,)
for (sx,sy,sh) in hidungs:
    cv2.rectangle(rooi_color, (sh,sy), (sx+sw, sy+sh), (255, 0, 0), 2)
    cv2.putText(frame, 'Hidung', (x+sx, y+sy), 1, 1, (0, 255, 0), 1)

if len(hidung)>0:
    mask_on = False
else:
    mask_on = True
    
```

Gambar 5. Klasifikasi Pendeteksi Masker.

Pengujian dilakukan dalam dua tahap, yang pertama menggunakan variasi sudut pada Tabel 1 merupakan pengujian dengan posisi Menghadap kedepan, Rotasi 30° ke kanan, Rotasi 30° ke kiri, Mengangkat dagu 15° ke atas, dan Menunduk kepala 15° ke bawah.

Tabel 1. Pengujian Pendeteksi Masker Berdasarkan sudut variasi.

Masker On	Masker Off
Menghadap kedepan	
	
Rotasi 30° ke kanan	
	
Rotasi 30° ke kiri	
	
Mengangkat dagu 15° ke atas	
	
Menunduk kepala 15° ke bawah	
	

Pengujian kedua berdasarkan dua orang objek dengan ketentuan Dua orang memakai masker, Dua orang tidak memakai masker, Serta Satu orang memakai masker dan satu orang tidak memakai masker, yang di tujukan pada Tabel 2 dapat mendeteksi dengan baik namun pada pengujian Satu orang memakai masker dan satu orang tidak memakai masker pendeteksi masker lebih objektif pada kondisi yang tidak memakai masker

Tabel 2. Pengujian Pendeteksi Masker Berdasarkan dua orang objek

Masker On	Masker Off
Dua orang memakai masker	Dua orang tidak memakai masker
	
Satu orang memakai masker dan satu orang tidak memakai masker	
	

Hasil pengujian pendeteksi masker berdasarkan sudut variasi dan dua orang objek pada Tabel 3 adalah sebagai berikut :

Tabel 3. Hasil Pengujian Pendeteksi Masker.

Ketentuan	Hasil
Masker on Menghadap kedepan	Terdeteksi
Masker on Rotasi 30° ke kanan	Terdeteksi
Masker on Rotasi 30° ke kiri	Terdeteksi
Masker on Mengangkat dagu 15° ke atas	Terdeteksi
Masker on Menunduk kepala 15° ke bawah	Terdeteksi
Masker off Menghadap kedepan	Terdeteksi
Masker off Rotasi 30° ke kanan	Terdeteksi
Masker off Rotasi 30° ke kiri	Terdeteksi
Masker off Mengangkat dagu 15° ke atas	Terdeteksi
Masker off Menunduk kepala 15° ke bawah	Terdeteksi
Dua orang memakai masker	Terdeteksi
Dua orang tidak memakai masker	Terdeteksi
Satu orang memakai masker dan satu orang tidak memakai masker	Terdeteksi

Conclusions

Sistem aplikasi pendeteksi masker menggunakan algoritma *haarcascade* dapat berjalan dengan baik berdasarkan

berdasarkan sudut variasi dan dua orang objek mampu mendeteksi masker dan membunyikan suara pemberitahuan jika tidak menggunakan masker serta memiliki hari, tanggal dan jam secara *realtime*. Kekurangannya agar bisa terdeteksi masker harus ditempatkan dengan cahaya yang cukup agar bisa berjalan dengan baik serta kurangnya *responsible* ketika tidak menggunakan masker karena melakukan perulangan dengan suara pemberitahuan. Untuk kedepannya aplikasi ini dapat mengambil tampilan masker *off* secara *responsible* dan menambahkan dataset penggunaan masker agar mendeteksinya lebih objektif. Kedepannya dapat digunakan sebagai alat pendeteksi masker yang diperuntukan khusus untuk tempat yang steril seperti *laboratorium* dan kesehatan medis.

References

- [1] Abdul, M., Irham, R., & Prasetya, D. A. (2020). Prototipe Pendeteksi Masker Pada Ruangan Wajib Masker Untuk Kendali Pintu Otomatis Berbasis Deep Learning Sebagai Pencegahan Penularan Covid-19. *Prototipe Pendeteksi Masker Pada Ruangan Wajib Masker Untuk Kendali Pintu Otomatis Berbasis Deep Learning Sebagai Pencegahan Penularan Covid-19*, 47–55. <https://publikasiilmiah.ums.ac.id/xmlui/bitstream/handle/11617/12377/108.pdf?sequence=1&isAllowed=y>
- [2] Al-Aidid, S., & Pamungkas, D. (2018). Sistem Pengenalan Wajah dengan Algoritma Haar Cascade dan Local Binary Pattern Histogram. *Jurnal Rekayasa Elektrika*, 14(1), 62–67. <https://doi.org/10.17529/jre.v14i1.9799>
- [3] Alief, Effendi, M. K., & Pramono, A. S. (2021). Sistem Pendeteksi Masker pada Pengemudi. *JEPIN (Jurnal Edukasi Dan Penelitian Informatika)*, 7(2), 250–254. doi: <http://dx.doi.org/10.26418/jp.v7i2.46877>
- [4] Aprilian Anarki, G., Auliasari, K., & Orisa, M. (2021). Penerapan Metode Haar Cascade Pada Aplikasi Deteksi Masker. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 5(1), 179–186. <https://doi.org/10.36040/jati.v5i1.3214>
- [5] Firmansyah, A. F. (2021). *Membuat Login & Register User GUI Dengan Python*. Ardian F Firmansyah. <https://www.youtube.com/watch?v=mudv6BsZUIs>
- [6] Irsyad, S., Putra, R., & Kurniawan, B. (2021). Rancang Bangun Human Interface Pendeteksi Suhu Serta Masker Menggunakan Matlab Design and Build a Temperature Detection Human Interface and Mask Using Matlab. *TELEKONTRAN*, 9(2).
- [7] Lambacing, M. M., & Ferdiansyah, F. (2020). Rancang Bangun New Normal Covid-19 Masker Detektor Dengan Notifikasi Telegram Berbasis Internet of Things. *Dinamik*, 25(2), 77–84. <https://doi.org/10.35315/dinamik.v25i2.8070>
- [8] Lia Farokhah. (2021). Perbandingan Metode Deteksi Wajah Menggunakan OpenCV Haar Cascade, OpenCV Single Shot Multibox Detector (SSD) dan DLib CNN. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 5(3), 609–614. <https://doi.org/10.29207/resti.v5i3.3125>
- [9] Luthfillah Ahmad, F., Nugroho, A., & Alfa Faridh Suni, D. (2021). Deteksi Pemakai Masker Menggunakan

- Metode Haar Cascade Sebagai Pencegahan COVID 19. *Edu Elekrika Journal*, 10(1), 13–18. <https://journal.unnes.ac.id/sju/index.php/eduel/article/view/47861/19493>
- [10] Naufal, M. F., Kusuma, S. F., Surabaya, U., Korespondensi, P., & Learning, T. (2021). PENDETEKSI CITRA MASKER WAJAH MENGGUNAKAN CNN DAN TRANSFER. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 8(6), 1293–1300. <https://doi.org/10.25126/jtiik.202185201>
- [11] Ngodein. (2020). *Deteksi Masker Otomatis! Membuat Aplikasi Covid dengan Python!* Ngodein. <https://www.youtube.com/watch?v=su6eA8nzRz4>
- [12] Radhi, T., Fitrah, M., & Nurdin, Y. (2021). Rancangn Bangun Pengembangan Pintu Otomatis Pendeteksi Masker dan Suhu Tubuh Menggunakan. *KITEKTRO*, 6(2), 7–14.
- [13] Syarif, M., & Wijanarto. (2015). Deteksi Kedipan Mata Dengan Haar Cascade Classifier Dan Contour Untuk Password Login. *Techno.Com*, 14(4), 242–249.
- [14] JThariq, A., & Bakti, R. Y. (2021). Sistem Deteksi Masker dengan Metode Haar Cascade pada Era New Normal COVID-19. *Jurnal Sistem Dan Teknologi Informasi*, 9(2), 241–244. <https://doi.org/10.26418/justin.v9i2.44309>
- [15] Tiwikrama, K., Rabi, A., Arifuddin, R., Elekto, T., Teknik, F., & Malang, U. M. (2021). Implementasi Palang Pintu Otomatis dengan Pendeteksi Masker Berbasis Raspberry Pi 3B +. *UN PGRI Kediri*.