

PENDETEKSI PEMAKAIAN MASKER MENGGUNAKAN WEBCAM SEBAGAI ALTERNATIF PENCEGAHAN COVID-19 DI INSTITUT TEKNOLOGI PADANG

Busran

*Jurusan Teknik Informatika S1
Fakultas Teknik
Institut Teknologi Padang
E-mail : busran.nofit@gmail.com*

Abstract

The use of masks during the COVID-19 pandemic is a requirement in a person's health protocol if they want to travel out of the house, such as to the office, campus and other crowded places. Wearing a mask can reduce the spread of COVID-19. Currently, the detection of the use of masks is still done manually by local officers. The purpose of this study is to implement face detection in the use of masks for health protocol purposes. Mask detection is done by camera. In taking the face using the camera is done in real time. The captured image will be converted to grayscale in order to avoid unnecessary bits. After obtaining grayscale, then this grayscale is converted into data form. The data obtained will be matched with the existing data on the facial harcascade. If the face harcascade data is the same as the data that has been converted from grayscale earlier, it is considered that the area has faces. The presence of a face, then proceed to the nose, the data obtained will be matched with the nose harcascade. If the nose is covered it is considered to be wearing a mask. On the other hand, if the nose is detected, it will not use a mask. The conclusion of this study is that the tool can work well in detecting masks according to the health protocol requirements.

Keywords- Detection mask, Face Detection, Haar Cascade, Covid-19

Intisari

Pemakaian masker disaat pandemi COVID-19 merupakan hal yang menjadi syarat dalam protokol kesehatan seseorang jika ingin bepergian keluar rumah, seperti ke kantor, kampus dan ketempat keramaian lainnya. Dengan menggunakan masker dapat mengurangi penyebaran COVID-19. Di Institut Teknologi Padang, pendeteksian pemakaian masker dilakukan secara manual oleh petugas keamanan sewaktu memasuki kampus. Tujuan dari penelitian ini untuk mengimplementasikan pendeteksian wajah dalam penggunaan masker untuk keperluan protokol kesehatan. Pendeteksian masker dilakukan dengan kamera. Dalam pengambilan wajah menggunakan kamera dilakukan secara real time. Gambar yang diambil akan dikonversikan ke grayscale tujuannya menghindari bit-bit yang tidak diperlukan. Setelah didapatkan grayscale selanjutnya grayscale ini dikonversikan menjadi bentuk data. Data yang didapatkan akan dicocokkan dengan data yang ada pada harcascade wajah. Apabila data harcascade wajah sama dengan data yang sudah dikonversikan dari grayscale tadi maka dianggap wilayah tersebut ada wajah. Adanya wajah, maka dilanjutkan pada bagian hidung, data yang didapat akan dicocokkan dengan harcascade hidung. Apabila hidung tertutup maka dianggap menggunakan masker, sebaliknya hidung terdeteksi maka tidak menggunakan masker. Sehingga dapat ditarik kesimpulan bahwa prototype alat bekerja dalam mendeteksi masker dengan bantuan kamera.

Kata Kunci— Masker deteksi, Face Detection, Haar Cascade, Covid-19

1. PENDAHULUAN

Pandemi *Corona Virus-19 (Covid-19)*, membuat lahirnya kebiasaan baru (*new normal*) yang diterapkan pada berbagai institusi termasuk kampus. Institut Teknologi Padang sebagai institusi yang juga menerapkan protokol kesehatan dalam pencegahan *covid* ini melakukan protokol kesehatan yaitu pemakaian

masker [2]. Pengecekan masker di kampus Institut Teknologi Padang masih dilakukan dengan cara manual. Kendala yang dialami, jika masuk ramai maka tidak dapat terdeteksi dengan baik dalam pemakaian masker.

Penggunaan teknologi saat ini mampu membantu manusia dalam pendeteksian masker. Untuk dapat mengenali beberapa bagian yang akan dideteksi, diperlukan mekanisme yang

mampu membantu dalam pendeteksian masker seperti hidung.

Penelitian yang dilakukan menggunakan *Raspberry Pi 3* model B+ untuk memproses citra yang direkam menggunakan kamera *webcam*, lcd 3.5 inchi untuk menampilkan hasil pendeteksiannya dan perangkat lunak yang digunakan di penelitian ini ialah *OpenCv* yang dapat melakukan pengolahan citra dinamis secara *realtime*.

Penelitian yang pernah dilakukan berkaitan dengan pendeteksian masker oleh Muzakkar Mu'minin, dkk [1]. Penelitian ini membahas tentang rancangan bangun *new normal covid-19* dalam penggunaan masker dengan hasil pendeteksiannya dikirim lewat telegram kepada keamanan setempat sebagai pemberitahuan kepada petugas keamanan.

Fathul, dkk Deteksi pemakai masker menggunakan metode *Haar cascade* sebagai pencegahan Covid-19 [2]. Hasil dari penelitian ini apabila wajah dan mulut terdeteksi maka program mengeluarkan *output* berupa teks yang diperingati dengan alarm. Jika mulut tidak terdeteksi maka program akan mengeluarkan *output* berupa teks yang menyatakan sudah memakai masker.

Galang, dkk [3] menggunakan metode *Haar cascade* pada aplikasi deteksi masker. Tujuan penelitian ini merancang alat yang mampu mengenali pemakaian masker untuk meminimalisir penularan Covid-19 di kampus Institut Teknologi Padang

2. METODOLOGI

2.1. Haar cascade

Haar cascade adalah kumpulan fungsi haar-like, yang digabungkan untuk membentuk pengklasifikasi. Fiturnya adalah jumlah nilai piksel putih yang dikurangkan dari nilai piksel pada area hitam.

Fungsi *haar-like feature* disebut juga dengan *haar cascade classifier* adalah fungsi persegi panjang yang memberikan indikasi spesifik pada gambar. Pengklasifikasi haar cascade berasal dari gabungan piksel hitam dan piksel putih yang membentuk kotak.[4]

Haar cascade classifier berasal dari gagasan Paul Viola dan Michael Jhon, karena itu dinamakan metode Viola dan Jhon. Ide dari *haar like feature* adalah mengenali objek berdasarkan nilai sederhana dari fitur tetapi bukan merupakan nilai piksel dari image objek tersebut. Metode ini memiliki kelebihan yaitu

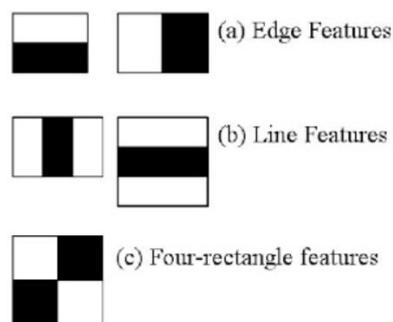
komputasi yang sangat cepat, karena hanya tergantung pada jumlah piksel dalam persegi bukan nilai piksel dari sebuah image [4].

Pendekatan untuk mendeteksi objek dalam gambar menggabungkan empat kunci utama yaitu *Haar Like Feature*, *Integral Image*, *Adaboost Learning* dan *Cascade Classifier*.

2.2. Haar like feature

Haar like feature merupakan fitur yang didasarkan pada algoritma *wavelet haar*. Setiap *haar like feature* terdiri dari gabungan kotak hitam dan putih. *Haar* adalah gelombang tunggal bujur sangkar (satu interval tinggi dan satu interval rendah) [5]. Untuk dua dimensi, satu terang dan satu gelap selanjutnya kombinasi-kombinasi kotak yang digunakan untuk pendeteksian objek visual yang lebih baik.

Pengelompokan image/citra dilakukan berdasarkan nilai fitur, ini bertujuan untuk memisahkan citra yang tidak diperlukan, dalam kasus ini warna latar tidak ikut dihitung[4]. Terdapat tiga jenis fitur berdasarkan jumlah persegi panjang (terang dan gelap) yang terdapat di dalamnya yaitu pola dua (*edge features*), pola tiga (*line features*), dan pola empat persegi panjang (*four-rectangle features*) seperti gambar 1.



Gambar 1. *Haar Like Feature* (https://docs.opencv.org/3.4/d2/d99/tutorial_js_face_detection.html)

Dengan adanya fitur haar ditentukan dengan cara mengurangi rata-rata piksel pada daerah gelap dari rata-rata piksel pada daerah terang. Jika nilai perbedaannya itu diatas nilai ambang atau *threshold*, maka dapat dikatakan bahwa difitur tersebut ada. Nilai dari haar like feature adalah perbedaan antara jumlah nilai piksel greyscale level dalam daerah kotak hitam dan daerah kotak putih.

2.3. Integral Image

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 interatif mulai dari ujung kiri atas hingga ujung kanan bawah dengan pergeseran sebesar x dan y . Semakin kecil nilai x dan y maka semakin akurat deteksi citra tersebut. Nilai x dan y yang sering digunakan adalah 1.

Untuk menentukan ada tidaknya *Haar Feature* disetiap lokasi gambar, *Viola* dan *Jones* memakai teknik yang disebut *integral image*. Nilai dari integral pada masing-masing piksel adalah penjumlahan dari semua piksel di atasnya dan disebelah kirinya. Dimulai dari integral pada masing-masing piksel adalah jumlah dari semua piksel di atasnya dan disebelah kirinya. Dimulai dari kiri atas sampai kanan bawah, gambar bisa diintegrasikan sebagai operasi matematika perpiksel.

2.4. Adaboost Learning

Adaboost menggabungkan banyak classifier untuk membuat satu classifier. Masing-masing classifier menetapkan satu bobot dan gabungan dari bobot inilah yang akan membentuk satu classifier yang kuat. *Viola* dan *Jones* [6] menggabungkan serangkaian *adaboost classifier* sebagai rantai filter.

Dalam prakteknya tidak satupun fitur yang mampu melakukan pengklasifikasian dengan *error* yang kecil. Algoritma *adaboost* berfungsi untuk mencari fitur-fitur yang memiliki tingkat pembeda yang tinggi. Hal ini dilakukan dengan mengevaluasi setiap fitur terhadap terbesar antara wajah dan non wajah.

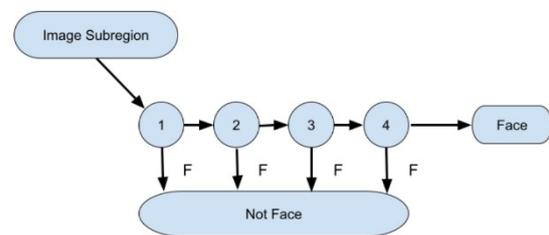
2.5. Cascade Classifier

Karakteristik dari *haar cascade* adalah adanya klasifikasi bertingkat (*cascade classifier*). Klasifikasi pada algoritma ini terdiri dari beberapa tingkatan dan tiap tingkatan mengeluarkan subcitra yang diyakini bukan wajah. Hal ini dilakukan karena lebih mudah untuk menilai subcitra yang bukan wajah daripada menilai apakah tersebut berisi wajah.

```
faceCascade =  
cv2.CascadeClassifier("/home/pi/opencv/data/haarcascades/haarcascae_frontend_default.xml")
```

dari code ini dapat dijelaskan bahwa kumpulan data wajah sudah dimasukkan dalam file xml bernama *haarcascade_frontend_default.xml*

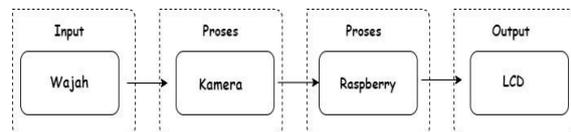
Seperti gambar 2, dijelaskan bahwa setiap *subwindows* dibandingkan dengan setiap fitur di setiap *stage*. Jika tidak mencapai target maka *subwindows* akan bergerak ke *subwindows* berikutnya dan melakukan perhitungan yang sama dengan proses sebelumnya. Pada proses selanjutnya didapatkan hasil yaitu *subwindows* yang terdeteksi sebagai wajah dan berlanjut ke subcitra berikutnya sampai pada akhirnya didapatkan kandidat kuat yang terdeteksi sebagai wajah.



Gambar 2. Cascade Classifier

<https://towardsdatascience.com/viola-jones-algorithm-and-haar-cascade-classifier-ee3bfb19f7d8>

2.6. Alur Pembacaan Sistem



Gambar 3. Alur Pembacaan sistem

3. HASIL DAN PEMBAHASAN

3.1. Mengaktifkan Kamera

Pendeteksian wajah dilakukan dengan *OpenCv* yang telah diinstallkan terlebih dahulu dengan menggunakan sebuah *typeface detector* yang disebut dengan *Haar-cascade classifier*, *Classifier* terus menerus membaca beberapa kali untuk mencari wajah pada gambar tersebut. *Classifier* menggunakan data yang disimpan pada file XML. Pembacaan wajah pada sistem menggunakan aplikasi *python* yang dibantu dengan *library opencv* dan *numpy*.

```
video_capture = cv2.VideoCapture(0)
```



Gambar 4. Kamera Aktif

Setelah kamera diaktifkan pada raspberry seperti gambar 4, kemudian atur resolusi kamera yang akan digunakan untuk pendeteksian wajah pada objek, berikut program untuk mengatur resolusi kamera:

```
video_capture.set(3,320)
video_capture.set(4,480)
```

Agar sistem dapat mencocokkan dengan *harcascade* yang diinputkan, sistem mengubah citra warna ke *grayscale*. Seperti gambar 5.



Gambar 5. Konversi Ke grayscale

3.2. Pendeteksian Wajah

Dalam pengambilan wajah menggunakan kamera yang dilakukan secara *realtime*. Gambar yang diambil akan dikonversikan ke *grayscale* tujuannya adalah menghindari bit-bit yang tidak diperlukan. Setelah didapatkan *grayscale* tahap selanjutnya dikonversikan menjadi bentuk data. Data yang didapatkan akan dicocokkan dengan data yang ada pada *harcascade*. Apabila data *harcascade* wajah sama dengan data yang sudah dikonversikan dari *grayscale* tadi maka dianggap wilayah tersebut ada wajah.

```
faces =
faceCascade.detectMultiScale(
    gray,
    scaleFactor=1.09,
    minNeighbors=3,
```

```
    minSize=(50, 50)
)
```

Selanjutnya membuat brooding box di area sekitar wajahnya, agar sistem mengetahui wajahnya. Sehingga dibuat perulangan seperti program berikut:

```
for (x, y, w, h) in faces:
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = frame[y:y+h, x:x+w]
    cv2.rectangle(frame, (x, y), (x+w,
y+h), (0, 255, 0), 1)
```

Akan muncul kotak persegi pada wajah, seperti gambar 6.



Gambar 6. Wajah terdeteksi

Jika wajah sudah terdeteksi selajutnya bagian hidung, data yang didapat akan dicocokkan dengan *harcascade* hidung yang diambil datanya dari file *haarcascade_mcs_nose.xml*.

```
nose_cascade =
cv2.CascadeClassifier('/home/pi/opencv/data/haarcascades/haarcasca
de_mcs_nose.xml')
```

Pola *mcs_nose.xml* apabila hidung tertutup maka dianggap menggunakan masker pada gambar 7. Sebaliknya hidung terdeteksi maka tidak menggunakan masker seperti gambar 8.



Gambar 7. Masker Terdeteksi



Gambar 8. Masker Tidak Terdeteksi

```
hidung =
nose_cascade.detectMultiScale(roi_gray)

for (sx,sy,sw,sh) in hidung:
    cv2.rectangle(roi_color, (sh,
sy),(sx+sw, sy+sh), (0,0,250),1)
```



Gambar 9. Deteksi Wajah Masker On atau Off

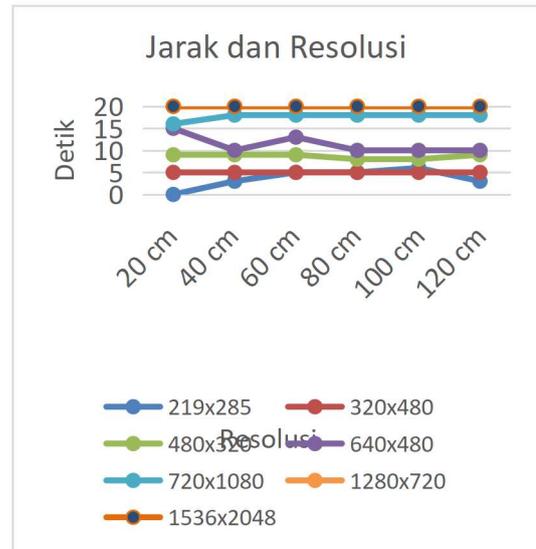
program diatas dapat dijelaskan bahwa jika sistem mendeteksi hidung maka akan muncul boarding box berwarna merah.

3.3. Pengujian Jarak dan Resolusi

Pengujian jarak dilakukan untuk mengabil kondisi terbaik antara kemampuan kamera (resolusi) terhadap konversi citranya menjadi data.

Tabel 1. Resolusi Kamera

Resolusi Kamera	Jarak Objek	Waktu Pembacaan
320x480	± 20 cm	5 s
320x480	± 40 cm	5 s
320x480	± 60 cm	5 s
320x480	± 80 cm	5 s
320x480	± 100 cm	5 s
320x480	± 120 cm	5 s



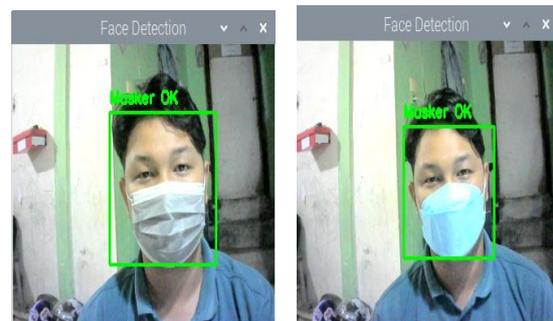
Gambar 10. Grafik Jarak dan Resolusi

Dari gambar 10, grafik tentang pengujian jarak dan resolusi yang akan digunakan untuk pendeteksian masker, dari grafik diatas pada resolusi 320x480 piksel dapat mendeteksi masker dengan cepat tanpa harus menunggu waktu yang lama dan dari pengujian jarak 20 sampai 120 cm dapat mendeteksi selama 5 detik.

3.4. Pengujian Warna Masker

pengujian ini menggunakan variabel untuk membantu sistem dalam mendeteksi masker. Potongan kode untuk variabel pendeteksian masker.

```
mask_on = False
```



Gambar 11. Warna Masker Terdeteksi

```
if mask_on:
    cv2.rectangle(frame, (x, y),
(x+w, y+h), (0, 250, 0), 1)
    cv2.putText(frame, 'Masker
OK', (x, (y-10)),
cv2.FONT_HERSHEY_SIMPLEX,
0.50, (0,255,0),1)
else:
```

```

cv2.rectangle(frame, (x, y),
(x+w, y+h), (0, 0, 255), 1)
cv2.putText(frame, 'NO
Masker', (x, (y-10)),
cv2.FONT_HERSHEY_SIMPLEX,
0.50, (0, 0, 255), 1)

```

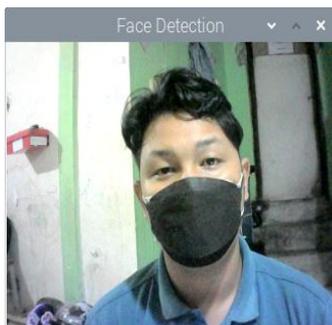
Variabel *mask_on* tersebut digunakan saat sistem mendeteksi hidung pada wajah yang ada di depan kamera.

```

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

```

Program pembaca masker terdapat di logika hidung yang menggunakan *len* (hidung), jika sistem mendeteksi hidung maka objek tidak menggunakan masker dan sebaliknya jika sistem tidak mendeteksi hidung maka sistem akan menganggap objek menggunakan masker.



Gambar 12. Warna Masker Tidak Terdeteksi

Dari pengujian yang dilakukan terdapat warna masker yang dapat dideteksi dan sulit untuk dideteksi. Warna masker yang dapat dideteksi yaitu warna masker yang terang seperti gambar 11, sedangkan warna yang sulit dideteksi yaitu warna hitam seperti gambar 12.

4. KESIMPULAN DAN SARAN

Hasil penelitian ini berupa alat prototype yang dapat bekerja mendeteksi masker sesuai syarat protokol kesehatan, alat bekerja dengan baik pada resolusi 320x480 piksel dan jarak 20 cm hingga 120 cm. sistem bekerja dengan baik mendeteksi masker berwarna terang serta dipengaruhi oleh cahaya sekitar.

DAFTAR PUSTAKA

- [1] Lambacing, Musakkarul Mu'minain, and Ferdiansyah Ferdiansyah, "Rancang Bangun New Normal Covid-19 Masker Detektor Dengan Notifikasi Telegram Berbasis Internet of Things ", *Dinamik*, vol 25, pp. 77-84,2020.
- [2] Ahmad, Fathul Luthfillah, Anan Nungroho and Alfa Faridh Suni, "Deteksi Pemakaian Masker Menggunakan Metode Haar Cascade Sebagai Pencegahan COVID 19", *Edu Elektrika*, Vol 10, 2021.
- [3] Anarki, Galang Aprilian, Karina Auliasari and Mira Orisa, "Penerapan Metode Haar Cascade Pada Aplikasi Deteksi Masker", *Jati*, vol.5, 2021.
- [4] Muhammad, Syarif, Wijanarto, "Deteksi Kedipan Mata Dengan Haar Cascade Classifier Dan Contour Untuk Password Login Sistem", *Techno.COM*, vol 14, pp. 242-249,2015.
- [5] 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>
- [6] Tyagi, Mrinal, (2021). Viola Jones Algorithm and Haar Cascade Classifier. *Internet* : <https://towardsdatascience.com/viola-jones-algorithm-and-haar-cascade-classifier-ee3bfb19f7d8>