PENERAPAN METODE HAAR CASCADE PADA APLIKASI DETEKSI MASKER

Galang Aprilian Anarki, Karina Auliasari, Mira Orisa
Program Studi Teknik Informatika S1, Fakultas Teknologi Industri
Institut Teknologi Nasional Malang, Jalan Raya Karanglo km 2 Malang, Indonesia
1718125@scholar.itn.ac.id

ABSTRAK

Saat ini menggunakan masker merupakan hal wajib yang harus dilaksanakan untuk meminimalisir atau mencegah wabah virus covid-19 terus menyebar. Pemeriksaan penggunaan masker tentunya membutuhkan tenaga manusia dalam melakukan pemeriksaan satu per satu. Tata cara pemeriksaan seperti ini memiliki beberapa keterbatasan yaitu tidak bisa dilakukan setiap waktu, jika pada kondisi malam hari di tempat-tempat umum tidak mungkin dilakukan karena petugas juga memiliki keterbatasan tenaga.

Dilakukannya penelitian ini bertujuan untuk menciptakan aplikasi yang dapat mendeteksi penggunaan masker untuk meminimalisir penularan virus Covid-19 yang saat ini menjadi wabah di Indonesia dengan fitur mengeluarkan peringatan yang berupa audio dan memotret jika ada terdeteksi tidak mengenakan masker, sehingga dapat meringankan beban kerja petugas di lapangan.

Pada penelitian ini metode yang digunakan ialah Haar Cascade. Haar Cascade adalah sebuah metode deteksi objek yang dibuat oleh Paul Viola dan Michael Jones. Pada tahun 2001, mereka mempresentasikan makalah yang disebut " Rapid Object Detection using a Boosted Cascade of Simple". Hasil dari penelitian ini adalah aplikasi dapat mendeteksi masker dari citra yang bersumber dari foto atau video dari *webcam internal* maupun eskternal dengan baik, dengan total keakuratan tertinggi 88,7% dan terendah 44,9%. Fitur peringatan yang berupa audio dan memotret juga dapat berkerja dengan baik.

***Kata Kunci** : haar cascade, deteksi masker, deteksi mulut, deteksi objek*

1. PENDAHULUAN

Saat ini menggunakan masker merupakan hal wajib yang harus dilaksanakan untuk meminimalisir atau mencegah wabah virus covid-19 terus menyebar. Pada kawasan perkantoran, tempat perbelanjaan, rumah sakit maupun tempat-tempat lain selalu ada proses pemeriksaan yang dilakukan pada setiap orang apakah memakai masker atau tidak. Selain pada beberapa tempat umum, pemeriksaan juga dilakukan pada pengendara kendaraan baik mobil, motor maupun transportasi umum untuk memakai masker.

Pemeriksaan penggunaan masker tentunya membutuhkan tenaga manusia dalam melakukan pemeriksaan satu per satu. Tata cara pemeriksaan seperti ini memiliki beberapa keterbatasan yaitu tidak bisa dilakukan setiap waktu, jika pada kondisi malam hari di tempat-tempat umum tidak mungkin dilakukan karena petugas juga memiliki keterbatsan tenaga. Selain keterbatasan waktu pemeriksaan, tempat pemeriksaan juga terbatas tidak bisa dilakukan di semua tempat secara mendetail karena keterbatasan jumlah petugas yang bersiaga maupun berkeliling melakukan pemeriksaan penggunaan masker.

Melihat beberapa keterbatasan yang ada pada proses pemeriksaan masker, peneliti berinisiatif mengembangkan suatu aplikasi yang dapat mendeteksi penggunaan masker atau tidak pada suatu tempat. Aplikasi pendeteksi penggunaan masker ini memiliki fitur peringatan dengan mengeluarkan suara jika ada orang yang terdeteksi tidak menggunakan masker. Aplikasi yang dikembangkan nantinya menggunakan metode haar cascade yang akan mendeteksi mulut yang melalui beberapa tahapan mulai dari input frame citra dari *webcam*, prapemrosesan, pemrosesan utama dan output deteksi. Diharapkan dengan dikembangkannya aplikasi deteksi masker ini dapat membantu meringankan peran petugas pemeriksa masker di tempat-tempat umum.

2. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Banyak penelitian terkait tentang bidang ilmu ekstraksi bentuk seperti penelitian yang dilakukan oleh Annga Wahyu Wibowo dkk pada tahun 2020. Penelitian ini membahas tentang "Pendeteksian dan pengenalan wajah pada foto secara real time dengan Haar cascade dan local Binary Pattern Histogram". Penelitian ini mampu mendeteksi dan mengenali wajah secara realtime dengan jarak 0 hingga 40 cm. (Wibowo et al. 2020)

Mahmudi Ali pada tahun 2014 pada penelitian yang judul "Deteksi Senjata Tajam Dengan Metode Haar Cascade Classifier Menggunakan Teknologi Sms Gateway". Tujuan penelitian ini yaitu untuk membuat aplikasi pendeteksi senajata tajam. Hasil dari penelitian ini adalah sistem dapat mengenali objek senjata tajam dengan akurasi 63,3% pada cahaya remang-remang, 70% pada cahaya normal dan 86% pada cahaya terang. (Mahmudi 2014)

Agustian di tahun 2012 membuat penelitian dengan judul "Mouse Kamera Dengan Deteksi Wajah Realtime Dan Deteksi Kedip Berbasis Metode Haarcascade Dan Surf". Tujuan penelitian ini adalah untuk membuat sistem pengganti penggunaan mouse

pada penyandang disabilitas, untuk menggerakkan mouse menggunakan metode haarcascade dengan mendeteksi wajah, sedangkan untuk klik pada mouse menggunakan kedipan mata dideteksi menggunakan SURF. Pengendalian kursor mouse dapat dilakukan dengan baik melalui penggunaan deteksi wajah realtime haarcascade, tanpa harus menggunakan optical flow, komputasi sistem menjadi lebih optimal. (Agustian,2012)

Puteri Rahma Tiara pada penelitian selanjutnya dengan judul "Deteksi Kantuk Menggunakan Kombinasi Haar Cascade dan Convolutional Neural Network". Tujuan penelitian ini adalah mendeteksi wajah mengantuk. Hasil penelitian ini sistem dapat mendeteksi wajah dan kantuk dengan sangat baik. (Puteri and Utaminingrum 2020)

Selanjutnya pada paper "Penerapan Haar Cascade Classification Model untuk Deteksi Wajah, Hidung, Mulut, dan Mata Menggunakan Algoritma Viola-Jones". Pada penelitiannya penulis menggunakan algoritma Viola-Jones dengan metode Haarcascade untuk mendeteksi Wajah, Hidung, Mulut, dan Mata dengan Bahasa Pemrograman Matlab. Sistem ini dapat mengenali dengan baik setiap bagian wajah yang disebutkan tadi. (Heryana, Rini Mayasari, and Kiki Ahmad Baihaqi 2020)

Muhammad Syarif pada penelitiannya yang berjudul "Deteksi Kedipan Mata Dengan Haar Cascade Classifier Dan Contour Untuk Password Login Sistem". Tujuan penelitian ini adalah untuk membuat deteksi kedip mata sebagai metode sandi baru untuk masuk ke dalam suatu sistem. Cara ini tidak lagi menggunakan keyboard sebagai alat masukan karena rawan untuk terjadi pencurian sandi dengan menekan tombol. (Syarif and Wijanarto 2015).

2.2 *Face Recognition*

Deteksi wajah adalah metode untuk menemukan dan mengekstrak sebuah fitur pada daerah wajah untuk keperluan pengenalan atau pendeteksian wajah. Pengenalan wajah merupakan salah satu teknologi dalam pengolahan citra (computer vision) yang dapat mengenali identitas seseorang atau informasi dari sebuah wajah (Hjelmås and Low 2001). Dari segi kegunaannya di bidang lain, teknologinya masih sangat luas, antara lain: keselamatan, robotika, atau Kesehatan (Zhao and Wei 2017)

Beberapa faktor yang bisa mempengaruhi pengenalan wajah meliputi:

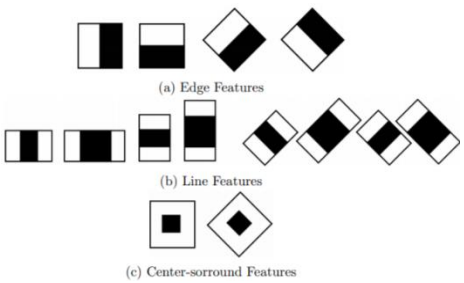
a. Daerah pada wajah ditunjukkan pada gambar pose mungkin berbeda (bagia depan terlihat jelas, beberapa wajah tidak terlihat).
b. Bagian wajah seperti jenggot, kumis, menggunakan kacamata, warna kulit berbeda, dll.
c. Mimik Wajah yang muncul pada gambar.
d. Arah gambar untuk mengambil gambar pada objek gambar. (Sinaga et al. 2017).

2.3 Haar Cascade

Deteksi objek menggunakan fungsi Haar berdasarkan cascade classifier adalah metode deteksi objek Paul Viola dan Michael Jones. Pada tahun 2001, mereka mempresentasikan makalah yang disebut "Deteksi Objek Cepat Menggunakan Perangkat Tambahan Sederhana". 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 atau biasa disebut juga dengan Haar cascade classifier adalah fungsi persegi panjang (persegi) yang memberikan indikasi spesifik pada gambar. Pengklasifikasi Haar Cascade berasal dari gabungan piksel hitam dan piksel putih yang membentuk kotak. (Muhammdad Syarif, 2015)

Untuk proses deteksi masker digunakan algoritma Haar Cascade. Umumnya, fungsi seperti haar digunakan untuk mendeteksi objek dalam gambar digital. Kata Haar sendiri dinyatakan sebagai fungsi matematika dalam bentuk kotak (wavelet Hhaar). Awalnya, pengolahan citra hanya didasarkan pada nilai RGB dari setiap piksel, tetapi cara pengolahan seperti itu tidak efektif. Kemudian, Viola dan Jones mengembangkan dan membentuk fungsi Haar-Like. Fitur Haar-like menangani gambar dalam kotak, di mana ada beberapa piksel dalam satu bingkai. Kemudian proses setiap kotak dan hasilkan nilai yang berbeda untuk menunjukkan area gelap dan terang. Nilai-nilai ini akan digunakan sebagai dasar untuk pengolahan citra. (Rahim Abdur, 2013)



Gambar 1. Haar Like Features

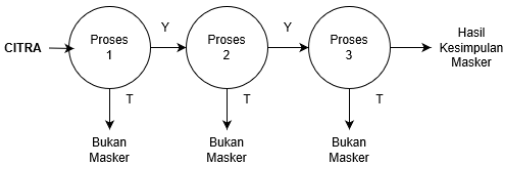
2.4 Klasifikasi Cascade

Pengklasifikasi kaskade adalah langkah penghitungan berulang kali nilai Fitur Haar untuk mendapatkan hasil yang lebih akurat. Gambar 2 menunjukkan alur kerja dari pengklasifikasi Cascade. Pada tahap klasifikasi 1, setiap sub-citra akan diklasifikasikan dengan ciri, jika hasilnya tidak memenuhi standar maka hasilnya akan ditolak. Pada klasifikasi tahap 2, setiap sub-citra akan direklasifikasi. Jika ambang batas yang diperlukan diperoleh, tahap filter berikutnya (tahap klasifikasi 3) dimasukkan. Sampai sub-gambar yang lewat dikurangi menjadi mendekati gambar dalam sampel yang telah diuji.

Pengklasifikasi Cascade terdiri dari setiap langkah yang berisi pengklasifikasi yang kuat. Karena itulah, semua fungsi dibagi menjadi beberapa langkah,

dan setiap tahapan memiliki sejumlah fungsi. Pekerjaan setiap tahap digunakan untuk menentukan apakah sub-jendela yang diberikan itu adalah sebuah objek atau bukan. Jika proses tidak menemui kesamaan nilai fitur, maka akan ditandai sebagai bukan objek.

Penulis menggunakan pustaka OpenCV Haarcascade untuk proses deteksi.



Gambar 2. Alur Klasifikasi Haar

2.5 OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah *library computer vision* dan *machine learning* (Opencv) yang berbasis *open source project*. Ini dibuat oleh Intel, yang berspesialisasi dalam pemrosesan citra, baik berupa gambar ataupun video.

OpenCV memiliki lebih dari 2500 algoritme pengoptimalan, termasuk satu set lengkap algoritme pembelajaran mesin dan pembelajaran komputer klasik dan mutakhir. Algoritma ini dapat digunakan untuk keperluan pendeteksi dan pengenalan wajah manusia ataupun hewan, mengetahui jenis objek, mengenali perilaku manusia dalam sebuah citra baik berupa video ataupun gambar, melacak pergeseran kamera, melacak pergerakan objek, dll.

3. METODE PENELITIAN

3.1. Analisis Sistem

Sistem ini dibangun menggunakan Bahasa Pemrograman Python dengan menggunakan Library OpenCV, citra inputan berupa gambar dan video, gambar didapatkan dari citra latih sedangkan untuk video diambil langsung menggunakan webcam yang lalu diolah tiap framenya.

Citra latih berupa citra yang telah disiapkan dengan total 736 citra dengan objek menggunakan masker. Sedangkan citra latih yang berupa citra masker untuk dimasukkan ke dalam klasifikasi berjumlah 150 citra, selanjutnya klasifikasi dilakukan menggunakan software cascade training GUI sehingga akan menghasilkan sebuah file berformat .xml yang berisi fitur dari citra masker.

Tahap akhir dilakukan pencocokan nilai fitur yang akan membedakan apakah objek mengenakan masker atau tidak, jika tidak mengenakan masker maka aplikasi akan menjalankan fitur peringatan yang berupa tulisan "Tolong Pakai Masker Anda", memainkan audio yang telah disediakan dan memotret objek tersebut.

3.2. Pengujian Datav

Pengujian data latih dilakukan dengan menyiapkan data latih yang dibagi menjadi 2 tipe, yaitu positif dan negative dengan total citra latih yaitu sebanyak:

- Total data latih positif sebanyak: 192 citra masker.
- Total data latih negative sebanyak: 900 citra tanpa masker

3.3. Citra RGB

Langkah pertama pada proses ini ialah memasukkan citra yang akan diuji, di sini penulis menguji citra RGB.

Citra RGB atau biasa dikenal dengan citra berwarna merupakan sebuah kumpulan piksel yang mempunyai 3 warna (Biru, Hijau, Merah). Mempunyai nilai (*range)* dari 0 (hitam) hingga 255 (putih).

Gambar 3. Citra RGB

3.4. Konversi Citra RBG ke Gray

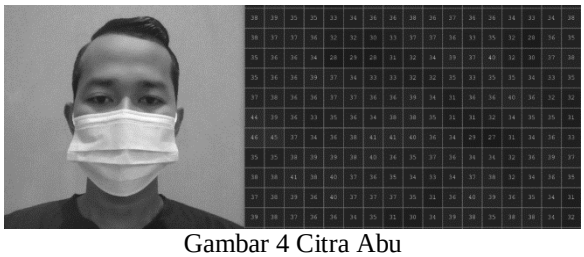
Citra Grayscale atau biasa juga disebut dengan citra skala keabuan merupakan sebuah kumpulan piksel yang terdiri dari 2 warna yaitu *Black* (hitam), *White* (Putih). Mempunyai nilai (*range)* dari 0 (hitam) hingga 255 (putih).

Mengubahan citra dari RGB ke Grayscale bisa dengan cara membagikan total nilai pada piksel dibagi 3.

$$Gray = (R + G + B) / 3$$

Atau bisa juga menggunakan persamaan 1:

$$Abu = 0.299*R + 0.587*G + 0.114*B \ldots\ldots(1)$$



Gambar 4 Citra Abu

3.5. Deteksi Masker

Deteksi adalah proses menentukan apakah suatu gambar berisi masker. Sebagai contoh, penulis menggunakan gambar seperti yang ada di gambar 3 yang telah diubah ke dalam *grayscale* sehingga menjadi seperti pada gambar 4.

OpenCV hadir dengan banyak pengklasifikasi terlatih. Kita dapat menggunakan pengklasifikasi wajah frontal untuk mendeteksi posisi wajah. Dalam

pengklasifikasi ini, ada modul detectMultiScale. Fungsi ini akan mengembalikan persegi panjang dengan koordinat (x, y, w, h) yang mengelilingi wilayah yang diinginkan. Parameternya yaitu:

- Scalefactor = Value menunjukkan seberapa besar ukuran gambar dikurangi di bawah setiap rasio gambar.
- minNeighbours = tentukan berapa banyak "tetangga" yang harus dimiliki setiap calon persegi panjang.
- minSize = ukuran objek minimum.

Di sini penulis menggunakan nilai yang berbeda untuk ScaleFactor dan minNeighbours.

```
mask = masker.detectMultiScale(gray,1.2,4)
```

3.6. ROI Masker

Langkah terakhir adalah melakukan perulangan semua koordinat yang dikembalikan dan menggambar *Region of Interest* dalam persegi panjang. Dalam contoh ini, penulis telah menggambar persegi panjang biru dengan ketebalan 2 dengan garis berwarna biru.

Sehingga menghasilkan seperti pada gambar 5

Gambar 5. Deteksi Masker

3.7. Deteksi Wajah

Jika masker tidak terdeteksi maka akan dijalankan fungsi untuk mendeteksi wajah. Sama seperti proses Deteksi Masker, apabila wajah terdeteksi maka akan dibuat sebuah kotak di area wajah seperti gambar 4. Jika tidak terdeteksi maka akan memunculkan tulisan “Wajah tidak terdeteksi”.

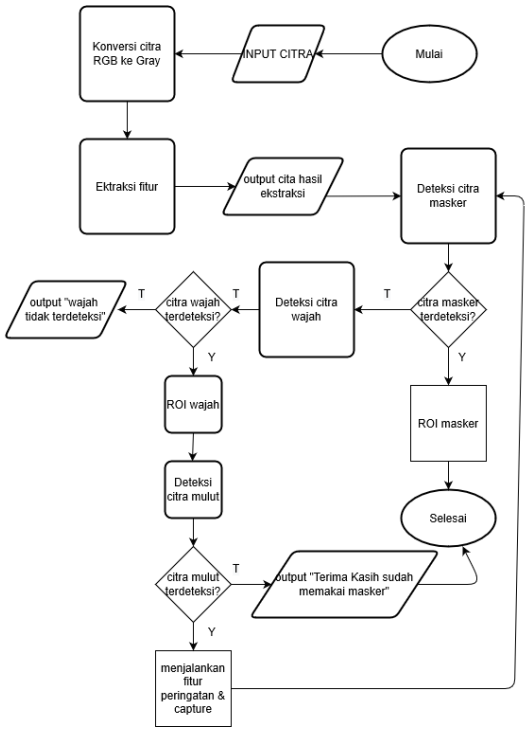
Gambar 6. Deteksi Wajah

3.8. Deteksi Mulut

Proses terakhir yaitu, apabila wajah terdeteksi dan mulut terdeteksi berarti orang itu tidak menggunakan masker. Maka akan muncul tulisan “Tolong Pakai Masker Anda”.

3.9. Diagram Alir Aplikasi

Diagram alir pada gambar 3 ini menjelaskan proses berjalananya aplikasi.



Gambar 7 Flowchart Sistem

4. HASIL DAN PEMBAHASAN

4.1. Pengujian Hardware

Pertama dilakukan pengujian komponen internal dari laptop yaitu webcam untuk pendeteksian secara *realtime* dapat berjalan dengan baik. Untuk pengujain menggunakan kamera eksternal hasilnya dapat dilihat pada table 1.

Gambar 8. Webcam

Gambar 8 merupakan hasil capture dari webcam pada kondisi malam hari.

Tabel 1. Pengujian Kamera Eksternal

No	Merk	Resolusi	Gambar	Hasil
1	Mi Max 3	64 MP		✓

2	Realme XT	64MP		✓
3	Samsung J7 Pro	13MP		✓
4	Xiaomi Redmi Note 3 Pro	16MP		✓

Keterangan:

✓ = terdeteksi

Berdasarkan table pengujian 1, dapat disimpulkan aplikasi tetap dapat berjalan dan mendeteksi masker dengan baik walaupun menggunakan kamera eksternal.

4.2. Pengujian Aplikasi

Pengujian dilakukan dengan total data yang diujikan yaitu 723 citra, pengujian dilakukan dengan 2 cara; manual dan otomatis.

Untuk cara manual, citra yang diuji di antaranya:

- Pengujian berdasarkan jarak terdekat & terjauh
- Pengujian berdasarkan kebutuhan daya pancar cahaya.
- Pengujian berdasarkan warna masker
- Pengujian File Executable

4.3. Pengujian Berdasarkan Jarak & Sudut

Pengujian dilakukan dengan dipengaruhi oleh 2 kondisi yaitu jarak dan sudut. Pertama dengan jarak objek ke kamera webcam yaitu mulai dari 20cm hingga 100cm. Pengujian dilakukan dengan 5 varia posisi sudut objek saat pengambilan gambar sebagai berikut;

1. Objek tegak lurus dengan kamera.
2. Objek bergeser 45° miring ke kanan
3. Objek bergeser 45° miring ke kiri
4. Objek bergeser 15° ke atas
5. Objek bergeser 15° ke atas

Gambar 9. Pengujian Jarak & Sudut

Tabel 2. Pengujian Berdasarkan Jarak & Sudut

No	Jarak	Sudut	Hasil
1	50, 100, & <=180 cm	Tegak Lurus	✓
2	50, 100, & <=180 cm	Miring Kanan 45°	✓
3	50, 100, & <=180 cm	Miring Kiri 45°	✓
4	50, 100, & <=180 cm	Nunduk 15°	✓
5	50, 100, & <=180 cm	Dongak 15°	✓
6	20cm & >180cm	Tegak Lurus	×
7	20cm & >180cm	Miring Kanan 45°	×
8	20cm & >180cm	Miring Kiri 45°	×
9	20cm & >180cm	Nunduk 15°	×
10	20cm & >180cm	Dongak 15°	×

Keterangan:

✓ = terdeteksi

× = tidak terdeteksi

Berdasarkan tabel 2 maka dapat disimpulkan aplikasi deteksi masker ini dapat mendeteksi masker dengan baik berdasarkan jarak dan juga posisi objek. Dapat disimpulkan pula jarak minimum untuk aplikasi deteksi masker ini yaitu 20cm dan maksimum yaitu 180cm, lebih dari 180cm aplikasi tidak bisa mendeteksi masker lagi.

4.4. Pengujian Berdasarkan Daya Pancar Cahaya

Pengujian ini dilakukan berdasarkan kondisi intensitas cahaya, di mana terdapat 3 kondisi saat pengujian dilakukan, yaitu gelap, normal, dan terang.

Gambar 10. Pengujian Berdasarkan Daya Pancar Cahaya

Tabel 3. Pengujian Berdasarkan Daya Pancar Cahaya

No	Warna Masker	Kondisi	Lux	Hasil	Waktu Deteksi
1	Biru	Terang	13000	✓	< 1 detik
2	Biru	Normal	400	✓	< 1 detik
3	Biru	Gelap	5	✓	< 5 detik
4	Putih	Terang	10000	✓	< 1 detik
5	Putih	Normal	400	✓	< 1 detik
6	Putih	Gelap	5	✓	< 3 detik
7	Hitam	Terang	10000	✓	< 1 detik
8	Hitam	Normal	30	✓	20 detik
9	Hitam	Gelap	5	×	53 detik

Keterangan:
✓ = terdeteksi
× = tidak terdeteksi

Berdasarkan tabel 3 maka dapat disimpulkan aplikasi deteksi masker ini dapat mendeteksi masker dengan baik jika masker yang digunakan berwarna terang. Intensitas cahaya juga berpengaruh pada proses pendeteksian, karena semakin baik intensitas cahaya semakin baik kinerja aplikasi dalam mengenali masker dan tidak memerlukan waktu yang lama.

4.5. Pengujian File *Executable*

Untuk mempermudah penggunaan, pembuatan file executable diperlukan, agar pengguna dapat langsung menggunakannya tanpa perlu menjalankan sebuah IDE/Server terlebih dahulu

Gambar 11. Pengujian File Exe

Pada gambar 11 dapat dilihat pengujian file exe dapat berjalan dengan baik dan menjalankan fitur deteksi masker.

Tabel 4. Pengujian File Exe

No	Merk	Versi Windows	Processor	RAM	VGA	Resolusi Layar	Konsumsi RAM ketika aplikasi dijalankan	Hasil Uji
1	Asus	10	I7	4GB	GTX 9000	1366x766	109MB	Berjalan Lancar
2	Asus	8.1	Amd	4GB	AMD	1366x768	110MB	Berjalan Lancar
3	Dell	7	Intel	4GB	IntelHD	1366x768	107MB	Berjalan Lancar
4	Dell	10	I5 8th gen	8GB	Nvidia GTX 1050	1920x1080	117MB	Berjalan Lancar
5	MSI	10	I7 4th	16GB	GTX 960M	1920x1080	111MB	Berjalan Lancar

Berdasarkan hasil pengujian pada table 4.8 dapat disimpulkan aplikasi dapat berjalan dengan lancar walaupun pada laptop dengan spesifikasi tergolong rendah.

4.6. Pengujian Dengan Data Citra Gambar

Dilakukan pengujian secara manual yaitu dengan menguji citra berupa gambar satu persatu. Didapatkan hasil seperti pada table 5.

Tabel 5. Pengujian Citra Gambar

No	Jumlah	Jenis Kelamin	Warna	Jenis	Jumlah Terdeteksi
1	10	Pria	Biru	Kain	10
2	10	Pria	Biru Muda	Bedah	8
3	10	Pria	Putih	N95	10
4	5	Pria	Putih	Kain	5
6	5	Wanita	Putih	Kain	3
7	3	Wanita	Hitam	Kain	2
8	2	Wanita	Hitam	Cadar	1
9	3	Pria	Hitam	Kain	2

Dari tabel 5 dapat dilihat bahwa total data uji sebanyak 48 citra, citra masker yang dapat dideteksi sebanyak 41 citra (positif). Dilakukan perhitungan dengan menggunakan persamaan 4.1 dan didapatkan nilai keakuratannya sebesar:

$$akurat = \frac{41}{48} \times 100 = 83\%$$

Sedangkan untuk pengolahan secara otomatis 1 folder dataset, berisi 723 citra positif. Di dapatkan hasil yang berbeda-beda tergantung dengan factor skala.

Tabel 6. Hasil Pengujian Otomatis

Skala	Terdeteksi	Keakuratan	Waktu
1.1	517	71.5 %	183 s
1.2	642	88,7 %	172 s
1.3	325	44,9 %	170 s

Berdasarkan pengujian didapatkan hasil seperti pada table 5 dan didapatkan kesimpulan bahwa aplikasi deteksi masker ini memiliki tingkat deteksi yang tinggi pada factor skala 1.2 dengan tinggkat keakuratan sebesar 88,7% dengan waktu penyelesaian 172 detik.

4.7. Pengujian Fungsional Aplikasi

Pengujian fungsionalitas aplikasi dilakukan dengan melakukan uji pada fitur yang ada di aplikasi sesuai dengan kebutuhan dan rancangan aplikasi seperti pada tabel 7. Pengujian dilakukan dengan menggunakan peramban Google Chrome v87 (64bit), Mozilla Firefox v75 (64 bit), Microsoft Edge v 44.18362.449.0 (64bit), dan Opera v73 (64 bit).

Tabel 7. Pengujian Fungsional Aplikasi

No	Fitur	Chrome	Firefox	Opera	Edge
1	Deteksi Masker	✓	✓	✓	✓
2	Deteksi Wajah	✓	✓	✓	✓
3	Deteksi Mulut	✓	✓	✓	✓
4	Memotret objek tidak memakai masker	✓	✓	✓	✓
5	Memainkan peringatan audio	✓	✓	✓	✓
6	Membuat bingkai persegi pada objek	✓	✓	✓	✓
7	Memunculkan peringatan berupa teks pada citra	✓	✓	✓	✓

Keterangan:
✓ = Bekerja
x = Tidak bekerja
Dilihat dari tabel 7 di atas menunjukkan bahwa keseluruhan fungsional aplikasi dapat bekerja dengan sangat baik di browser berbeda.

4.2.6 Pengujian User

Tabel 8. Pengujian User

No	Pertanyaan	Penilaian		
		Iya	Mungkin	Tidak
1	Apakah aplikasi deteksi masker yang menerapkan metode haar cascade ini mudah digunakan?	80%	20%	0%
2	Apakah menurut anda aplikasi deteksi masker yang menerapkan metode haar cascade ini dapat membantu memeprmudah kinerja petugas di lapangan?	70%	30%	0%
3	Apakah hasil dari aplikasi deteksi masker yang menerapkan metode haar cascade ini akurat?	60%	40%	0%
4	Apakah fitur dari aplikasi deteksi masker yang menerapkan metode haar cascade ini sudah cukup?	80%	20%	0%

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan beberapa scenario pengujian didapatkan kesimpulan berupa :

1. Hasil pengujian fungsional menunjukkan hasil 100 %, artinya semua fitur program dapat berjalan sesuai dengan fungsinya walaupun di browser berbeda.
2. Berdasarkan pengujian, aplikasi dapat mengenali objek masker dari sebuah video/gambar yang didapat langsung dari *webcam* dengan mengenali nilai fitur yang telah dilatih dengan jarak minimum 20cm dan maksimum 180cm.
3. Berdasarkan pengujian, aplikasi dapat mengenali objek masker dari sebuah video/gambar dengan ketentuan daya pancar cahaya minimal 5lx untuk citra masker berwarna terang, sedangkan untuk gelap yaitu 10 lx. Sedangkan 13000lx merupakan maksimal saat aplikasi ini diuji.
4. Berdasarkan pengujian, aplikasi dapat dijalan pada sistem operasi Windows 7, Windows 8.1, hingga Windows 10 dengan penggunaan RAM terendah 107MB dan tertinggi 117MB.
5. Berdasarkan pengujian, aplikasi dapat berjalan dengan baik dengan kamera *internal* maupun eksternal.
6. Berdasarkan pengujian kepada responden, aplikasi deteksi masker ini mudah digunakan.
7. Metode Haarcascade bisa diterapkan untuk aplikasi pendeteksi sebuah objek dengan melakukan *training* data terlebih dahulu.
8. Aplikasi dapat mendeteksi masker dengan berbagai warna dan bahan yang telah diuji.
9. Aplikasi dapat berjalan baik di browser Opera, Chrome, Edge, dan Firefox.
10. Posisi objek sangat mempengaruhi hasil pendeteksian, objek sebaiknya berada pada posisi tegak lurus dari sumber pengambilan citra.
11. Intensitas cahaya berperan penting dalam mendeteksi objek, pencahayaan yang bagus akan menghasilkan pendeteksian yang baik.
12. Jarak objek dari sumber pengambilan citra mempengaruhi proses deteksi, jarak yang terlalu dekat atau terlalu jauh akan mengurangi hasil pendeteksian.
13. Hasil pengujian bisa berbeda-beda hasilnya tergantung parameter yang digunakan seperti faktor skala atau minNeighbour.
14. Hasil pengujian menggunakan faktor skala 1.2 saat ini merupakan yang terbaik dengan tingkat keakuratan tertinggi yaitu sebesar 88,7% sedangkan factor skala 1.3 adalah yang terburuk dengan tingkat keakuratans sebesar 44,9%.

5.2 Saran

Beberapa saran yang dapat penulis berikan setelah melakukan penelitian ini untuk pengembangan berikutnya antara lain :

1. Memberikan notifikasi kepada petugas terdekat.
2. Menambahkan citra masker untuk dilatih lagi agar hasil lebih maksimal.
3. Menambahkan *GUI* agar tampilan lebih menarik.
4. Menambahkan fitur untuk CCTV/IPCAM.
5. Bisa dipadukan dengan metode LBPH.
6. Bisa dikembangkan menggunakan Bahasa Pemrograman lain seperti C++, Matlab, Java.
7. Bisa dipadukan dengan *Internet of Things* (IOT).

DAFTAR PUSTAKA

[1] Abidin, Suhepy. 2018. "Deteksi Wajah Menggunakan Metode Haar Cascade Classifier Berbasis Webcam Pada Matlab." *Jurnal Teknologi Elekterika* 15(1): 21.

[2] Agustian, Indra, Risanuri Hidayat, and Th Sri Widodo. "Mouse Kamera Dengan Deteksi Wajah Realtime Dan Deteksi Kedip Berbasis Metode Haarcascade.": 1–5.

[3] Heryana, Nono, Rini Mayasari, and Kiki Ahmad Baihaqi. 2020. "Penerapan Haar Cascade Classification Model Untuk Deteksi Wajah, Hidung, Mulut, Dan Mata Menggunakan Algoritma Viola-Jones." *Techno Xplore: Jurnal Ilmu Komputer dan Teknologi Informasi* 5(1): 21–25.

[4] Hjelmås, Erik, and Boon Kee Low. 2001. "Face Detection: A Survey." *Computer Vision and Image Understanding* 83(3): 236–74.

[5] Kadir, Abdul, and Adhi Susanto. 2013. Buku Image *Pengolahan Citra Teori Aplikasi*. Yogyakarta: Andi.

[6] Kuhlman, Dave. 2009. "A Python Book: Beginning Python, Advanced Python, and Python Exercises." *A Python Book*.

[7] Kurniawan, Luthfi Maslichul. 2015. "Metode Face Recognition Untuk Identifikasi Personil Berdasar Citra Wajah Bagi Kebutuhan Presensi Online Universitas Negeri Semarang." *Scientific Journal of Informatics* 1(2): 210–20.

[8] Madenda, Sarifuddin. 2013. Teori Aplikasi dan Pemrograman Menggunakan Matlab *Pengolahan Citra Dan Video Digital*. ed. Ade M. Drajat. Erlangga.

[9] Mahmudi, Ali. 2014. "Deteksi Senjata Tajam Dengan Metode Haar Cascade Classifier Menggunakan Teknologi Sms Gateway." *Matics* 1(1): 27–30.

[10] Nugroho, Hendro. 2017. "Deteksi Citra Objek Lingkaran Dengan Menggunkan Metode Ekstraksi Bentuk Circularity." 2(1): 54–59.

[11] Pavani, Sri Kaushik, David Delgado, and Alejandro F. Frangi. 2010. "Haar-like Features with Optimally Weighted Rectangles for Rapid Object Detection." *Pattern Recognition*.

[12] Purwanto, Panji, Burhanuddin Dirgantoro, and Agung Nugroho Jati. 2015. "Implementasi Face Identification Dan Face Recognition Pada Kamera Pengawas Sebagai Pendeteksi Bahaya." *eProceedings of Engineering*.

[13] Puteri, Rahma Tiara, and Fitri Utaminingrum. 2020. "Deteksi Kantuk Menggunakan Kombinasi Haar Cascade Dan Convolutional Neural Network." 4(3): 816–21.

[14] Rahim, Abdur, N. Hossain, T. Wahid, and S. Azam. 2013. "Face Recognition Using Local Binary Patterns (LBP)." *Global Journal of Computer Sience and Technology Graphics & Vision*.

[15] Sinaga, Ijon Posmarohatta, Ig Prasetya, Dwi Wibawa, and Ekki Kuniawan. 2017. "Background Substraction Dan Haar Cascade People Counter and Face Identification System With Background." *e-Proceeding of Engineering* 4(2): 1544–51.

[16] Sutoyo, T. et al. 2009. 1 *Teori Pengolahan Citra Digital*. Andi.

[17] Syarif, Muhammad, and Wijanarto. 2015. "Deteksi Kedipan Mata Dengan Haar Cascade Classifier Dan Contour Untuk Password Login." *Techno.com* 14(4): 242–49.

[18] Viola, Paul, and Michael Jones. 2001. "Rapid Object Detection Using a Boosted Cascade of Simple Features." In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[19] Wibowo, Angga Wahyu et al. 2020. "Pendeteksian Dan Pengenalan Wajah Pada Foto Secara Real Time Dengan Haar Cascade Dan Local Binary Pattern Histogram." *JTET (Jurnal Teknik Elektro Terapan)* Vol. 9 No.: 6 – 11.

[20] Zhao, Xue Mei, and Cheng Bing Wei. 2017. "A Real-Time Face Recognition System Based on the Improved LBPH Algorithm." In *2017 IEEE 2nd International Conference on Signal and Image Processing, ICSIP 2017*.