

DETEKSI WAJAH MEMAKAI MASKER MENGGUNAKAN  
ALGORITMA HAAR CASCADE

*Face detection wearing mask using Haar cascade algorithm*

Untuk penelitian Skripsi S-1

Disusun oleh:  
Damai Prasetya  
17312270



29/10/2022  
Acc. Berisi  
Cek.  
Kecamatan Arie

09/11/22  
" " " " " "  
Ded. Darus

Acc. Cekh.  
18/11-2022  
" " " "

PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK DAN ILMU KOMPUTER  
UNIVERSITAS TEKNOKRAT INDONESIA  
BANDAR LAMPUNG  
2022

DETEKSI WAJAH MEMAKAI MASKER MENGGUNAKAN  
ALGORITMA HAAR CASCADE

*Face detection wearing mask using Haar cascade algorithm*

Untuk penelitian Skripsi S-1

Disusun oleh:  
Damai Prasetya  
17312270



**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK DAN ILMU KOMPUTER  
UNIVERSITAS TEKNOKRAT INDONESIA  
BANDAR LAMPUNG  
2022**

## HALAMAN PENGESAHAN

Skripsi

DETEKSI WAJAH MEMAKAI MASKER MENGGUNAKAN  
ALGORITMA HAAR CASCADE

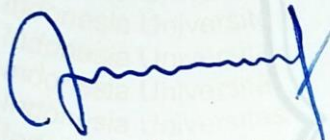
Dipersiapkan dan disusun oleh

**DAMAI PRASETIYA**  
**17312270**

Telah dipertahankan di depan Dewan Penguji  
Pada tanggal 13 Agustus 2022

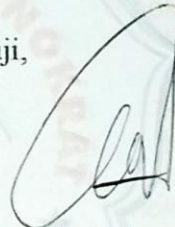
Penguji

Pembimbing,



**Dr. Si. Dedi Darwis, M.Kom.**  
NIK. 022 13 02 07

Penguji,



**Maulana Aziz Assuja, S.ST., M.T.**  
NIK. 022 10 03 02

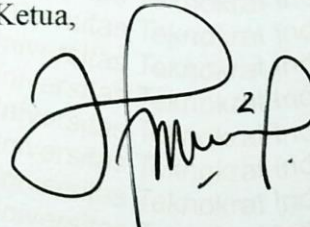
Skripsi ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar sarjana  
Tanggal 18 November 2022

Fakultas Teknik dan Ilmu Komputer  
Dekan,



**Dr. H. Mahathir Muhammad, S.E., M.M.**  
NIK. 023 05 00 09

Program S1 Informatika  
Ketua,



**Dyah Ayu Megawaty, M.Kom.**  
NIK. 022 09 03 05

## LEMBAR PERNYATAAN KEASLIAN PENELITIAN

Yang bertanda tangan dibawah ini:

Nama : **Damai Prasetya**  
NPM : **17312270**  
Program Studi : **SI Informatika**

Dengan ini menyatakan bahwa laporan skripsi:

Judul : **Deteksi Wajah Memakai Masker Menggunakan Algoritma Haar Cascade**

Pembimbing : **Dr. Si. Dedi Darwis, M.Kom.**

Belum pernah diajukan untuk diuji sebagai persyaratan untuk memperoleh gelar akademik pada berbagai tingkat di Universitas/Perguruan Tinggi manapun. Tidak ada bagian dalam skripsi ini yang pernah dipublikasikan oleh pihak lain, kecuali bagian yang digunakan sebagai referensi, berdasarkan kaidah penulisan ilmiah yang benar.

Apabila dikemudian hari ternyata laporan tugas akhir yang saya tulis terbukti hasil *saduran/plagiat*, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan sebenar-benarnya.

Bandarlampung, 12 Agustus 2022

Yang menyatakan,



**Damai Prasetya**  
NPM. 17312270

**LEMBAR PERNYATAAN PUBLIKASI  
SKRIPSI UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Program Studi S1 Informatika Fakultas Teknik dan Ilmu Komputer Universitas Teknokrat Indonesia, saya yang bertanda tangan dibawah ini:

Nama : Damai Prasetya  
NPM : 17312270  
Program Studi : Informatika  
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Program Studi S1 Informatika Fakultas Teknik dan Ilmu Komputer Universitas Teknokrat Indonesia **Hak Bebas Royalti Non Eklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul:

“Deteksi Wajah Memakai Masker Menggunakan Algoritma Haar Cascade”.

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas **Royalti Noneklusif** ini Program Studi S1 Informatika Fakultas Teknik dan Ilmu Komputer Universitas Teknokrat Indonesia berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilih Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Bandar Lampung  
Pada tanggal : 12 Agustus 2022

Yang menyatakan,

Damai Prasetya  
NPM. 17312270

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar sarjana pada Program Studi SI Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia. Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan laporan ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Dr. H.M. Nasrullah Yusuf, S.E., M.B.A., selaku Rektor Universitas Teknokrat Indonesia.
2. Dr. H. Mahathir Muhammad, S.E.,M.M., selaku Dekan Fakultas Teknik dan Ilmu Komputer Universitas Teknokrat Indonesia.
3. Dyah Ayu Megawaty, S.Kom., M.Kom. Selaku Ketua Program Studi S1 Informasi Fakultas Teknik dan Ilmu Komputer Universitas Teknokrat Indonesia.
4. Bapak Dr. Si. Dedi Darwis, M.Kom. selaku dosen pembimbing yang telah membimbing saya pada skripsi ini.
5. Bapak Maulana Aziz Assuja, S.ST., M.T. selaku dosen penguji yang telah memberikan masukan pada skripsi ini.
6. Bapak atau Ibu dosen Fakultas Teknik dan Ilmu Komputer S1 Informatika
7. Teman-teman semuanya yang selalu mendukung dan memberi masukan kepada penulis.

Akhir kata, penulis berharap semoga Tuhan YME berkenan membalas segala kebaikan semua pihak yang telah membantu dan semoga Laporan Skripsi ini membawa manfaat bagi pengembangan ilmu.

Bandar Lampung, 12 Agustus 2022  
Penulis

Damai Prasetiya  
NPM. 17312270

## DAFTAR ISI

COVER .....	i
HALAMAN PENGESAHAN.....	ii
LEMBAR PERNYATAAN KEASLIAN PENELITIAN.....	iii
LEMBAR PERNYATAAN PUBLIKASI .....	iv
KATA PENGANTAR .....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR .....	x
DAFTAR TABEL.....	xi
DAFTAR LAMPIRAN.....	xii
ABSTRAK .....	xiii
ABSTRAK .....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
BAB II LANDASAN TEORI.....	5
2.1 Tinjauan Pustaka .....	5
2.1.1 Literatur 1 .....	10
2.1.2 Literatur 2.....	11
2.1.3 Literatur 3.....	12
2.1.4 Literatur 4.....	13
2.1.5 Literatur 5.....	13
2.1.6 Literatur 6.....	14
2.1.7 Literatur 7.....	15
2.1.8 Literatur 8.....	16
2.1.9 Literatur 9.....	16
2.1.10 Literatur 10.....	17
2.1.11 Tinjauan Literatur.....	18
2.2 Haar Cascade .....	18

2.2.1	Haar Features .....	19
2.2.2	Integral Image .....	20
2.2.3	AdaBoost.....	21
2.2.4	Cascade classifier.....	22
2.2.5	OpenCV .....	23
2.2.6	Computer Vision .....	23
2.3	Python.....	24
2.4	Masker .....	24
2.5	Deteksi wajah .....	24
2.6	Confusion Matrix .....	25
BAB III METODOLOGI PENELITIAN.....		26
3.1	Kerangka Penelitian .....	26
3.2	Tahapan Penelitian .....	27
3.2.1	Identifikasi Masalah .....	27
3.2.2	Studi Literatur .....	28
3.2.3	Pengumpulan Data .....	28
3.2.3.1	Dokumentasi.....	28
3.2.4	Preprocessing .....	28
3.2.4.1	Tahap Greyscaling.....	28
3.2.4.2	Data Training.....	29
3.2.5	Analisis sistem .....	29
3.2.5.1	Analisa Prinsip Kerja Haar Cascade .....	29
3.2.6	Perancangan Sistem .....	32
3.2.6.1	Input Citra.....	33
3.2.6.2	Grayscaling.....	34
3.2.6.3	Deteksi Wajah .....	34
3.2.6.4	ROI .....	34
3.2.6.1	Deteksi Hidung.....	34
3.2.6.5	Deteksi Masker.....	35
3.2.7	Skema Pengujian .....	36
3.2.7.1	Pengujian Fungsional .....	37
3.2.7.2	Pengujian Berdasarkan Sudut dan Jarak.....	37
3.2.7.3	Pengujian Berdasarkan Citra Gambar .....	38
3.2.7.4	Pengujian Menggunakan <i>Confusion Matrix</i> .....	38



BAB IV HASIL DAN PEMBAHASAN .....	41
4.1 Preprocessing.....	41
4.1.1 Training.....	41
4.2 Implementasi Program .....	43
4.2.1 Input Citra .....	43
4.2.2 Grayscaleing .....	43
4.2.3 Deteksi Wajah .....	43
4.2.4 Deteksi Hidung .....	44
4.2.5 Deteksi masker .....	45
4.3 Hasil Pengujian.....	45
4.3.1 Pengujian Fungsional .....	45
4.3.2 Pengujian Berdasarkan Sudut dan Jarak .....	47
4.3.3 Pengujian Dengan Citra Gambar .....	50
4.3.4 Pengujian Menggunakan <i>Confusion Matrix</i> .....	51
BAB V PENUTUP.....	56
DAFTAR PUSTAKA .....	58
LAMPIRAN.....	60

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Teknik integral image .....	20
<b>Gambar 2.2</b> Proses cascade classifier .....	22
<b>Gambar 3.1</b> Kerangka penelitian .....	26
<b>Gambar 3.2</b> Tahapan penelitian.....	27
<b>Gambar 3.3</b> Flowchart Metode Haar Cascade.....	30
<b>Gambar 3.4</b> Fitur Haar.....	31
<b>Gambar 3.5</b> Representasi Integral Image .....	31
<b>Gambar 3.6</b> Diagram Alir Program .....	33
<b>Gambar 3.7</b> Flowchart Deteksi Masker.....	36
<b>Gambar 3.3</b> <i>Confusion Matrix</i> .....	39
<b>Gambar 4.1</b> Data Negatif.....	41
<b>Gambar 4.2</b> Data Positif .....	41
<b>Gambar 4.3</b> Proses Training .....	42
<b>Gambar 4.4</b> Kode input citra .....	43
<b>Gambar 4.5</b> Kode merubah citra grayscale .....	43
<b>Gambar 4.6</b> Kode program cascade.....	44
<b>Gambar 4.7</b> Kode deteksi wajah dan ROI.....	44
<b>Gambar 4.8</b> Kode deteksi hidung, ROI .....	45
<b>Gambar 4.9</b> Kode deteksi masker, ROI.....	45
<b>Gambar 4. 10</b> Data Positif .....	51
<b>Gambar 4. 11</b> Data Negatif.....	52
<b>Gambar 4. 12</b> <i>Data True positive</i> .....	53
<b>Gambar 4. 13</b> <i>Data True negative</i> .....	53
<b>Gambar 4. 14</b> <i>Data False positive</i> .....	54
<b>Gambar 4. 15</b> <i>Data False negative</i> .....	54

## DAFTAR TABEL

<b>Tabel 2.1</b> Tinjauan Literatur.....	5
<b>Tabel 3.1</b> Pengujian Fungsional .....	37
<b>Tabel 3.2</b> Pengujian berdasarkan Sudut dan Jarak .....	37
<b>Tabel 4.1</b> Pengujian Fungsional .....	46
<b>Tabel 4.2</b> Pengujian Berdasarkan Sudut dan jarak.....	47
<b>Tabel 4.3</b> Pengujian dengan citra gambar .....	50
<b>Tabel 4.4</b> Confusion matrix.....	52
<b>Tabel 4.5</b> Hasil Confusion Matrix .....	55

## DAFTAR LAMPIRAN

Lampiran 1 Kode Program Deteksi Menggunakan Image.....	60
--	----

## **ABSTRAK**

Masks are one of the means to prevent the transmission of COVID-19 and have become a government regulation from 2020. Therefore, in every place such as companies, hospitals, and other public places, masks are tested strictly, many still use human labor before enter the place. Of course, this method is less effective and efficient because it can take a lot of energy and time.

Seeing some of the limitations that exist in the mask inspection process, the researcher took the initiative to develop a system that can detect whether someone is wearing a mask or not. By using the haar cascade algorithm as a method of making face detection programs wearing masks.

In this study, several tests were carried out including functional testing, testing based on angles and distances, testing with images and testing using a confusion matrix. The results of this study are the system can detect faces wearing masks in real time and use images with an 81% accuracy rate.

**Keywords:** Haar cascade, face detection wearing a mask, nose detection

## **ABSTRAK**

Masker merupakan salah satu sarana untuk pencegahan penularan covid-19 dan sudah menjadi peraturan pemerintah dari tahun 2020. Maka dari itu di setiap tempat seperti perusahaan, rumah sakit, dan tempat umum lainnya di lakukan pemeriksaan masker secara ketat masih banyak yang menggunakan tenaga manusia sebelum memasuki tempat tersebut. Tentunya cara tersebut kurang efektif dan efisien karna dapat memakan banyak tenaga dan waktu.

Melihat beberapa keterbatasan yang ada pada proses pemeriksaan masker, peneliti berinisiatif mengembangkan suatu sistem yang dapat mendeteksi apakah seseorang menggunakan masker atau tidak. Dengan menggunakan algoritma haar cascade sebagai metode pembuatan program pendeteksian wajah memakai masker.

Pada penelitian ini dilakukan beberapa pengujian diantaranya, pengujian fungsional, pengujian berdasarkan sudut dan jarak, pengujian dengan citra gambar dan pengujian menggunakan confusion matrix. Hasil dari penelitian ini adalah sistem dapat mendeteksi wajah memakai masker secara real time dan menggunakan gambar dengan memiliki tingkat akurasi 81%.

Kata kunci : Haar cascade, deteksi wajah memakai masker, deteksi hidung

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

*Corona virus* jenis baru yang ditemukan pada manusia di Wuhan Cina pada Desember 2019, kemudian diberi nama *Severe Acute Respiratory Syndrome Coronavirus 2* (SARS-COV2), dan menyebabkan penyakit *Coronavirus Disease-2019* (COVID-19). Penyebaran virus 2019-nCoV sangat cepat hingga ke 204 negara di dunia. Pada akhirnya, bulan Maret 2020 *World Health Organization* (WHO) menetapkan penyakit ini sebagai pandemi global dan menyebut virus corona ini dengan sebutan COVID-19. Di Indonesia sendiri sudah terdampak COVID-19 sejak Maret 2020 hingga saat ini.

Pemerintah selaku pembuat kebijakan memberlakukan peraturan baru agar setiap orang dapat keluar rumah dan melakukan pekerjaan tanpa harus khawatir terpapar virus COVID-19 ini. Salah satunya adalah peraturan yang dibuat oleh Gubernur DKI Jakarta, yaitu bagi setiap orang yang akan keluar rumah wajib menggunakan masker dan juga tetap melakukan *physical distancing*. Penggunaan masker saat ini merupakan salah satu hal yang sangat penting untuk melindungi diri dari virus COVID-19. Karena penularan COVID-19 itu sendiri dapat melalui bersin atau batuk antar manusia.

Pemeriksaan masker biasanya menggunakan tenaga manusia secara satu persatu. Cara pemeriksaan seperti ini memiliki kekurangan yaitu tidak bisa dilakukan setiap waktu. Pada kondisi di tempat-tempat umum seperti di pintu masuk mall, perusahaan dan sekolah tidak mungkin terus dilakukan setiap saat karena petugas juga memiliki keterbatasan tenaga. Perkembangan teknologi yang

semakin berkembang pesat mempermudah masyarakat dalam segala hal, salah satunya yaitu untuk mendeteksi atau mengenali bentuk wajah manusia menggunakan *computer vision*. *Computer Vision* membuat komputer dapat melihat dan mengenali bentuk yang meniru mata dan otak manusia. Penggunaan *computer vision* dapat dipadukan dengan *machine learning* untuk memahami dan menirukan sifat manusia dengan baik (Zulkhaidi et al., 2020).

Pengembangan *computer vision* banyak menggunakan module yang telah disediakan para pengembang seperti Intel yang menyediakan module OpenCV secara *open source* yang dapat digunakan untuk bahasa pemrograman Python dan C++. Pendeteksian hidung dan wajah mengambil data yang telah disediakan oleh Intel yaitu Haar Cascade yang mengidentifikasi wajah dan hidung dengan pixel pada kamera. Perhitungan numeric menggunakan numpy modul pada python agar dapat menghitung koordinat dengan tepat pada wajah dan hidung.

Melihat beberapa keterbatasan yang ada pada proses pemeriksaan masker, peneliti berinisiatif mengembangkan suatu sistem yang dapat mendeteksi apakah seseorang menggunakan masker atau tidak. Pada sistem pendeteksi masker ini terdapat fitur jika tidak menggunakan masker akan ditandai dengan kotak berwarna merah, terdapat tulisan tidak menggunakan masker dan jika menggunakan masker akan ditandai dengan kotak berwarna hijau terdapat tulisan menggunakan masker. Sistem yang dikembangkan nantinya akan menggunakan Algoritma Haar Cascade yang akan mendeteksi wajah terutama bagian mulut dan hidung.

Beberapa peneliti menggunakan berbagai metode untuk mendeteksi wajah seperti Principal Component Analysis atau PCA atau Scale Invariant Feature



Transform (SIFT). Namun metode-metode tersebut membutuhkan banyak komputasi untuk pengolahannya. Untuk mempercepat proses komputasi maka algoritma haar cascade dipergunakan (Al-Aidid & Pamungkas, 2018). Algoritma Haar Cascade Classifier adalah salah satu algoritma yang digunakan untuk mendeteksi wajah atau objek. Algoritma tersebut mampu mendeteksi dengan cepat dan realtime sebuah benda termasuk wajah manusia. Algoritma Haar Cascade Classifier memiliki kelebihan yaitu perihal komputasi yang cepat karena hanya bergantung pada jumlah piksel dalam persegi dari sebuah gambar (Abidin, 2018).

Penelitian yang dilakukan oleh (Aprilian Anarki et al., 2021), yang berjudul “Penerapan Metode Haar Cascade Pada Aplikasi Deteksi Masker”. Pada penelitian tersebut untuk menentukan apakah seseorang menggunakan masker atau tidak berdasarkan deteksi mulut dan wajah. Berdasarkan pengujian aplikasi dapat mengenali objek masker dari sebuah webcam dengan nilai fitur yang telah dilatih dengan jarak minimum 20cm dan maksimum 80cm dan 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 keakuratan sebesar 44,9%.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang masalah yang telah di uraikan di atas, maka rumusan masalah pada penelitian ini adalah:

1. Bagaimana menerapkan Algoritma Haar Cascade untuk deteksi wajah menggunakan masker?

2. Bagaimana tingkat akurasi Algoritma Haar Cascade dalam deteksi wajah menggunakan masker?

### **1.3 Batasan Masalah**

Pembatasan masalah sangat diperlukan untuk memfokuskan suatu penelitian agar hasil yang didapat lebih- maksimal. Batasan masalah pada penelitian ini diantaranya:

1. Hanya dapat mendeteksi pemakaian masker medis
2. Wajah yang dideteksi adalah wajah manusia
3. Tidak dapat menyimpan data gambar yang terdeteksi
4. Sistem yang dikembangkan hanya sebatas pembuktian pendeteksian wajah memakai masker dan sebagai bentuk implementasi pengembangan Haar Cascade

### **1.4 Tujuan Penelitian**

1. Tujuan penelitian ini adalah untuk mendeteksi wajah memakai masker dengan menggunakan Algoritma Haar Cascade Classifier.
2. Mengetahui tingkat akurasi Algoritma Haar Cascade Classifier.

### **1.5 Manfaat Penelitian**

1. Membantu dalam pendeteksian masker
2. Sebagai bahan perbandingan bagi peneliti lain yang tertarik dengan permasalahan objek deteksi secara realtime dan Algoritma Haar Cascade Classifier.

## BAB II LANDASAN TEORI

### 2.1 Tinjauan Pustaka

Tinjauan pustaka merupakan kumpulan dari penelitian-penelitian sebelumnya yang dapat digunakan untuk mendukung penelitian yang sedang dilakukan. Berikut adalah tinjauan literatur yang ditampilkan dalam Tabel 2.1

**Tabel 2.1** Tinjauan Literatur

No	Peneliti	Judul	Metode/Algoritma	Hasil
1	Sayed Al-Aidid, Daniel S. Pamungkas	Sistem Pengenalan Wajah dengan Algoritma Haar Cascade dan Local Binary Pattern Histogram	Haar Cascade, Local Binary Pattern Histogram	Hasil dari penelitian tersebut adalah sebuah system pengenalan wajah yang dapat mendeteksi walaupun di sekitar manusia terdapat beberapa objek lain, kecepatan yang dihasilkan dalam mendeteksi juga cukup cepat
2	Dicky Giancini, Eva Yulia Puspanin	Identifikasi Penggunaan Masker Menggunakan	You Only Look Once (YOLO)	Hasil dari penelitian tersebut terdapat 2 jenis dataset yaitu dataset yang telah diaugmentasi dapat mendeteksi secara

	grum, Yisti Vita Via	Algoritma CNN YOLOv3-Tiny		realtime adanya masker atau tidak dengan tingkat keakurasian yang cukup tinggi sedangkan data yang tidak di augmentasi pada proses realtime dapat mendeteksi masker atau tidak tetapi dengan tingkat akurasi yang sangat buruk .
3	Galang Aprilian Anarki, Karina Auliasari , Mira Orisa	PENERAPAN METODE HAAR CASCADE PADA APLIKASI DETEKSI MASKER	Haar Cascade	Hasil penelitian tersebut adalah sebuah aplikasi deteksi masker yang dapat mengenali objek masker dari video/gambar yang di dapat melalui webcam dengan jarak minimum 20 cm dan maksimum 80 cm.
4	Suhepy Abidin	Deteksi Wajah Menggunakan Metode Haar	METODE HAAR CASCADE CLASSIFIER	Hasil penelitian tersebut adalah sebuah aplikasi deteksi wajah

		Cascade Classifier Berbasis Webcam Pada Matlab		menggunakan webcam, jika citra wajah terhalang oleh objek lain maka wajah tidak akan terdeteksi, intensitas cahaya, posisi/kemiringan pada wajah berperan penting untuk dapat mendeteksi dengan tepat.
5	Tri Septiana Nadia Puspita Putri, Mohama d Al Fikih , Novendra Setyawan	FACE MASK DETECTION COVID-19 USING CONVOLUTIONAL NEURAL NETWORK (CNN)	CONVOLUTIONAL NEURAL NETWORK (CNN)	Hasil penelitian tersebut adalah sebuah system untuk deteksi masker dengan akurasi 0.9933% dan training loss 0.0213% dengan dataset sebanyak 2000 95,2% training dan 4,8% pengujian.
6	Musakka rul Mu'mini	RANCANG BANGUN NEW NORMAL	Deep Learning Convolutional Neural Network	Hasil penelitian tersebut adalah sistem masker detektor dengan

	m Lambaci ng, Ferdians yah	COVID-19 MASKER DETEKTOR DENGAN NOTIFIKASI TELEGRAM BERBASIS INTERNET OF THINGS		notifikasi telegram dengan internet of things berjalan dengan baik, mampu mendeteksi masker dan mengirimkan langsung melalui telegram sebagai pemberitahuan.
7	Truong quang vinh, Nguyen tran ngoc anh	Real-Time Face Mask Detector Using YOLOv3 Algoritehm and Haar Cascade Classifier	YOLOv3, Haar Cascade Classifier	Hasil penelitian tersebut adalah sebuah sistem yang dapat mendeteksi masker atau tidak dan dapat bekerja secara realtime  Dangan 30fps, sistem bertujuan untuk diterapkan guna mengurangi penyebaran penyakit COVID-19
8	Mohd Suhairi Md Suhaimin	Real-time mask detection and face recognition using eigenfaces	Eigenfaces and local binary pattern histogram (LBPH)	Hasil penelitian tersebut adalah sebuah sisitem yang dapat mendeteksi masker secara realtime

	, Mohd Hanafi Ahmad Hijazi, Chung Seng Kheau, Chin Kim On	and local binary pattern histogram for attendance system		dan pengenalan wajah untuk absensi, deteksi masker dalam pengenalan wajah bertujuan untuk menyesuaikan dimasa pandemi seperti sekarang.
9	G. Jignesh Chowdary, Narinder Singh Punn, Sanjay Kumar Sonbhadr a, and Sonali Agarwal	Face Mask Detection using Transfer Learning of InceptionV3	Transfer Learning of InceptionV3	Hasil penelitian tersebut adalah sebuah sistem untuk mengklasifikasi orang yang memakai masker atau tidak, guna mencegah penularan COVID-19, model transfer learning yang diusulkan mencapai akurasi dan spesifitas 99,92%, 99,9% pada pelatihan dan 100%, 100% selama pengujian pada dataset.

10	Li Cuimei, Qi Zhiliang , Jia Nan , Wu Jianhua	Human face detection algorithm via Haar cascade classifier combined with three additional classifiers	Haar-cascade Classifier	Hasil penelitian tersebut adalah sebuah sistem pendeteksi wajah dalam sistem tersebut pengklasifikasian lemah berdasarkan histogram warna kulit manusia kemudian dilakukan pengklasifikasian tambahan berdasarkan deteksi mulut dan mata.
----	---	--	----------------------------	---

Berikut adalah penjelasan singkat mengenai literatur pada tabel 2.1 penelitian sebelumnya :

### 2.1.1 Literatur 1

Perkembangan teknologi multimedia, membuat penggunaan kamera dan pengolah gambar/video meningkat pesat. Pendeteksian dan pengenalan wajah adalah salah satu penggunaan dari kamera dan pengolah gambar/video. Absensi, kamera keamanan, aplikasi keamanan pada smartphone adalah beberapa aplikasi pengenalan wajah. Pengenalan dengan menggunakan wajah tidak dapat digandakan, dicuri, ataupun terlupa. Untuk pengenalan beberapa peneliti menggunakan berbagai metoda seperti *Principal Component Analysis* (PCA) atau *Scale Invariant Feature Transform* (SIFT). Sedangkan pada penelitian ini untuk mendeteksi wajah digunakan algoritma *Local Binary Pattern* (LBP), sedangkan



untuk mempercepat komputasi menggunakan algoritma *Haar Cascade*. Pada penelitian ini untuk membedakan wajah dan bukan wajah menggunakan webcam, serta bekerja secara *real time* untuk mengenali orang yang telah terdaftar pada database. Hasil yang didapat menunjukkan bahwa system pengenalan wajah yang dapat mendeteksi walaupun di sekitar manusia terdapat beberapa objek lain, kecepatan yang dihasilkan dalam mendeteksi juga cukup cepat.

Perbedaan penelitian yang akan diteliti dengan literature 1 terdapat pada algoritma yang digunakan dan objek yang dideteksi, pada penelitian yang akan diteliti penulis menggunakan algoritma haar cascade dan pada penelitian yang akan diteliti penulis mendeteksi wajah yang menggunakan masker sedangkan pada literatur 1 menggunakan 2 algoritma yaitu *Local Binary Pattern* (LBP) dan *Haar Cascade* sedangkan pada objek yang dideteksi pada literatur 1 adalah wajah manusia untuk mengenali wajah.

### **2.1.2 Literatur 2**

COVID-19 adalah penyakit yang menular yang disebabkan oleh virus yang baru ditemukan yaitu jenis coronavirus. Dengan adanya virus ini maka World Health Organization (WHO) menerapkan protokol kesehatan guna mencegah penularan coronavirus lebih luas, protokol kesehatan yang di berlakukan berupa menjaga jarak, menggunakan masker, mencuci tangan. Dengan adanya arahan tersebut masih banyak orang yang tidak menggunakan masker di tempat umum dan masih dilakukan pemeriksaan secara manual. Sehingga dibutuhkan sistem yang dapat mendeteksi masker sebagai opsi untuk peringatan dini. Munculnya *Deep Learning* menciptakan berbagai riset untuk menciptakan sebuah deteksi wajah, benda, dan beberapa jenis lainnya. Pada penelitian ini menggunakan algoritma You

Only Look Once (YOLO). Hasil dari penelitian ini terdapat 2 jenis secara akurasi untuk dataset yang tidak dilakukan secara *real time* akurasi yang dihasilkan untuk pendeteksian apakah menggunakan masker atau tidak berkisar antara 98-100% , sedangkan dataset yang dilakukan secara *real time* berakurasi antara 95-100%.

Perbedaan penelitian yang akan diteliti dengan literature 2 terdapat pada algoritma yang digunakan, pada penelitian yang akan diteliti penulis menggunakan algoritma haar cascade sedangkan pada literatur 2 algoritma yang digunakan adalah YOLO.

### **2.1.3 Literatur 3**

Pada masa pandemi saat ini menggunakan masker adalah hal yang wajib untuk mencegah wabah COVID-19 terus menyebar. Untuk pemeriksaan masker biasanya menggunakan tenaga manusia secara satu per satu . Cara pemeriksaan seperti ini memiliki kekurangan yaitu tidak bisa dilakukan setiap waktu, jika pada kondisi malam hari di tempat-tempat umum tidak mungkin dilakukan karena petugas juga memiliki keterbatasan tenaga. Tujuan dilakukannya penelitian ini adalah untuk menciptakan aplikasi yang dapat mendeteksi apakah seseorang menggunakan masker atau tidak untuk mencegah penularan COVID-19 dengan fitur berupa peringatan berupa audio dan memotrit jika ada yang terdeteksi tidak menggunakan masker, dalam penelitian ini menggunakan algoritma *Haar Cascade* adalah salah satu algoritma yang dapat mendeteksi objek. Hasil penelitian ini adalah sebuah aplikasi deteksi masker yang dapat mengenali objek masker dari video/gambar yang di dapat melalui webcam dengan jarak minimum 20 cm dan maksimum 80 cm.

Perbedaan penelitian yang akan diteliti dengan literature 3 terdapat pada objek yang dideteksi, pada penelitian yang akan diteliti penulis mendeteksi wajah dan hidung untuk mendeteksi apakah seseorang menggunakan masker atau tidak sedangkan pada literatur 3 objek yang di deteksi yaitu wajah dan mulut untuk mendeteksi seseorang menggunakan masker.

#### **2.1.4 Literatur 4**

Saat ini telah banyak yang memanfaatkan fitur deteksi wajah seperti sistem akses keamanan pada pintu maupun smartphone. Deteksi wajah dapat dilakukan dengan berbagai cara atau metode, salah satu metode nya adalah *Haar Cascade Classifier* yang di gunakan pada penelitian ini. Penelitian ini bertujuan untuk mengimplementasikan algoritma *Haar Cascade Classifier* untuk sebuah aplikasi deteksi wajah. Hasil penelitian tersebut adalah sebuah aplikasi deteksi wajah menggunakan webcam, jika citra wajah terhalang oleh objek lain maka wajah tidak akan terdeteksi, intensitas cahaya, posisi/kemiringan pada wajah berperan penting untuk dapat mendeteksi dengan tepat.

Perbedaan penelitian yang akan diteliti dengan literature 4 terdapat pada objek yang dideteksi, pada penelitian yang akan diteliti penulis mendeteksi wajah dan hidung untuk mendeteksi apakah seseorang menggunakan masker atau tidak sedangkan pada literatur 3 objek yang di deteksi hanya wajah.

#### **2.1.5 Literatur 5**

Coronavirus Disease (COVID-19) telah menyebar hampir ke seluruh dunia ter masuk di Indonesia, sehingga pemerintah melakukan pembatasan sosial bersekala besar (PSBB) di setiap daerah, menurut WHO salah satu cara pencegahan

untuk mengurangi penyebaran COVID-19 adalah dengan menaati protokol kesehatan antarlain mencuci tangan, menjaga jarak dan memakai masker saat di luar rumah, selama ini pendeteksia masker masih dilakukan oleh manusia. Dimana petugas keamanan tidak dapat selalu menjaga atau memantau siapa saja yang menggunakan masker. Penelitian ini menerapkan pengolahan citra untuk mendeteksi masker. *Mechine learning* berbasis *neural network* Banyak di terapkan pada pendeteksian objek. Pada penelitian ini menggunakan metode *Convolutional Neural Networks* (CNN). Tujuan penelitian ini dilakukan untuk dapat mendeteksi apakah seseorang menggunakan masker atau tidak sehingga dapat membantu petugas keamanan dalam menerapkan protokol kesehatan. Hasil penelitian ini adalah sebuah system untuk deteksi masker dengan akurasi 0.9933% dan training loss 0.0213% dengan dataset sebanyak 2000 95,2% training dan 4,8% pengujian. Dataset yang di ambil bervariasi dengan gambar masker menggunakan topi, hijab, dan tidak menggunakan atribut.

Perbedaan penelitian yang akan diteliti dengan literature 6 terdapat pada algoritma yang digunakan, pada penelitian yang akan diteliti penulis menggunakan algoritma haar cascade sedangkan pada literatur 5 algoritma yang digunakan adalah CNN.

### **2.1.6 Literatur 6**

Dimasa pandemi COVID-19 diberlakukan nya protocol kesehatan dimana setiap orang wajib menaati protokol seperti mencuci tangan, menjaga jarak dan menggunakan masker. Ini adalah salah satu kebijakan baru yang di tetapkan pemerintah sehingga menjadi kebiasaan baru atau disebut *new normal*. Peraturan ini juga berlaku di lingkungan perusahaan yang sudah mulai kembali berjalan

dimana karyawan wajib mengenakan masker dan menjaga jarak saat di kantor. Agar protokol kesehatan dapat berjalan dengan baik dan tertib maka dibutuhkan sebuah system *New Normal* COVID-19 Masker Detektor dengan Notifikasi Telegram berbasisi Internet Of Things. Pada penelitian ini menggunakan mini komputer yaitu Raspberry Pi dan menggunakan metode Deep Learning Convolutional Neural Network. Tujuan penelitian ini mendeteksi masker agar para karyawan dapat disiplin dalam menjalankan protokol kesehatan. Hasil dari penelitian ini berupa masker detektor dengan notifikasi telegram dengan internet of things berjalan dengan baik, mampu mendeteksi masker dan mengirimkan langsung melalui telegram sebagai pemberitahuan.

### **2.1.7 Literatur 7**

Selama pandemi COVID-19 diberlakukan nya protokol kesehatan salah satunya adalah memakai masker, penggunaan masker diberlakukan di beberapa perusahaan, departemen dan kantor kerja untuk mengurangi penyebaran COVID-19. Untuk pemeriksaan apakah seseorang menggunakan masker atau tidak, dibutuhkan computer vision yang dapat melakukan pendeteksian jenis ini. Akhir-akhir ini, deteksi objek menggunakan *deep learning* banyak digunakan berkat performa yang di hasilkan luar biasa. Untuk penelitian ini menggunakan algoritma YOLOv3 algoritma ini memberikan kinerja yang tinggi dalam masalah deteksi dan klasifikasi. Tujuan penelitian ini adalah meyajikan detector masker wajah secara *real time* yang dapat menjadi alarm saat mendeteksi seseorang tanpa memakai masker. Hasil dari penelitian tersebut adalah sebuah sistem yang dapat mendeteksi apakah seseorang memakai masker atau tidak dan dapat bekerja secara realtime

Dengan 30fps, sistem bertujuan untuk diterapkan guna mengurangi penyebaran penyakit COVID-19.

### 2.1.8 Literatur 8

Face recognition semakin populer untuk indentifikasi, control akses dan register dalam suatu organisasi. Baru-baru ini banyak sistem absensi yang diusulkan menggunakan pengenalan wajah. Hal ini mengurangi beban absensi secara manual, mencegah penipuan dan kecurangan dalam absensi. Di antara teknik biometrik, pengenalan wajah menyediakan metode tanpa kontak yang memenuhi persyaratan skenario saat ini dibandingkan dengan analisis sidik jari, dan pengenalan iris. Untuk mencegah penyebaran pandemi COVID-19 yang mewajibkan memakai masker, pemakaian masker untuk menutupi hidung dan mulut sudah diidentifikasi sebagai salah satu langkah pencegahan. Dengan demikian, sistem absensi konvensional harus diganti dengan sistem absensi digitalisasi untuk menyediakan lingkungan yang aman bagi para peserta. Tujuan penelitian ini adalah membuat sistem pengenalan wajah dan deteksi masker supaya dapat mencegah penyebaran penyakit COVID-19. Metode yang digunakan untuk membuat sistem tersebut dengan menggunakan *Eigenfaces* and *local binary pattern histogram* (LBPH). Hasil penelitian ini adalah sebuah sistem yang dapat mendeteksi masker secara realtime dan pengenalan wajah untuk absensi, deteksi masker dalam pengenalan wajah bertujuan untuk menyesuaikan dimasa pandemi seperti sekarang.

### 2.1.9 Literatur 9

Dimasa pandemi COVID-19 sekarang ini maka disarankan oleh Organisasi Kesehatan Dunia (WHO) ke berbagai negara untuk memastikan warganya agar

memakai masker di tempat umum. Dengan cepatnya penularan COVID-19, WHO telah menyatakannya sebagai global pandemi. Menurut WHO, kasus yang terinfeksi di seluruh dunia mendekati 22 juta. Mayoritas kasus positif ditemukan di area padat dan padat. Oleh karena itu, telah ditentukan oleh para ilmuwan bahwa memakai masker di tempat umum dapat mencegah penularan penyakit. Dalam penelitian ini, model *transfer learning* diusulkan untuk mengotomatiskan proses mengidentifikasi orang-orang yang tidak memakai masker. Model yang diusulkan dibangun dengan menyempurnakan model pembelajaran mendalam, InceptionV3. Model yang diusulkan dilatih dan diuji pada Simulated Masked Face Dataset (SMFD). Tujuan dari penelitian ini adalah untuk mengembangkan model pembelajaran yang mendalam untuk mendeteksi orang yang tidak menggunakan masker. Hasil penelitian ini berupa sebuah sistem untuk mengklasifikasi orang yang memakai masker atau tidak, guna mencegah penularan COVID-19, model transfer learning yang diusulkan mencapai akurasi dan spesifitas 99,92%, 99,9% pada pelatihan dan 100%, 100% selama pengujian pada dataset.

#### **2.1.10 Literatur 10**

Deteksi wajah manusia memiliki peran penting dalam interaksi manusia-mesin dan aplikasi berbasis computer vision. Sebagai informasi identitas manusia, wajah manusia memiliki keunggulan keunikan dan tidak dapat direplikasi. Namun, karena variasi yang besar seperti iluminasi yang berbeda, ekspresi dan latar belakang serta ketidakpastian lainnya, deteksi wajah manusia tetap menjadi masalah yang menantang dalam aplikasi dunia nyata. Pada penelitian ini algoritma pendeteksian wajah manusia menggunakan algoritma *haar cascade classifier* yang dikombinasikan dengan tiga pengklasifikasi tambahan. Pengklasifikasi additional

didasarkan pada pencocokan histogram rona kulit, deteksi mata, dan deteksi mulut. Tujuan penelitian ini adalah sistem yang dapat mendeteksi wajah manusia dan dapat membedakan mana yang manusia dan bukan manusia. Hasil penelitian berupa sebuah sistem yang dapat mendeteksi wajah dalam sistem tersebut pengklasifikasian *additional* berdasarkan histogram warna kulit manusia kemudian dilakukan pengklasifikasian tambahan berdasarkan deteksi mulut dan mata.

### **2.1.11 Tinjauan Literatur**

Pada penelitian sebelumnya terdapat beberapa perbedaan dengan penelitian yang akan diteliti. Perbedaan pada literatur 1, 2, 5, 6, 7, 8, dan 9 terdapat pada algoritma yang digunakan pada penelitian yang akan diteliti penulis menggunakan algoritma haar cascade. Sedangkan pada literatur 3, 4, 10 perbedaannya terdapat pada objek yang dideteksi, pada penelitian yang akan diteliti penulis mendeteksi wajah dan hidung untuk mendeteksi apakah seseorang menggunakan masker atau tidak sedangkan pada literatur 3 objek yang dideteksi yaitu wajah dan mulut untuk mendeteksi seseorang menggunakan masker dan pada literatur 4 dan 10 objek yang dideteksi hanya wajah.

## **2.2 Haar Cascade**

Algoritma *Haar Cascade* merupakan salah satu model machine learning yang kerap kali digunakan sebagai pondasi aplikasi *object detection* (terutama *face recognition*), dalam sebuah gambar maupun video (Viola & Jones, 2004).

Konsep pendeteksian wajah memakai masker dengan menggunakan metode adaboost yang sekarang lebih dikenal dengan Algoritma Haar Cascade Classifier. Wajah yang dilakukan untuk perhitungan ini adalah wajah manusia. Ketika wajah berhasil dideteksi maka dilakukan penandaan pada wajah.



Algoritma *Haar Cascade* menerapkan *cascade function* untuk mentraining gambar melalui 4 tahapan utama (Viola & Jones, 2004):

1. Menentukan *Haar features*
2. Integral imege
3. Adabost
4. *Cascade calsifier*

### **2.2.1 Haar Features**

Algoritma ini menggunakan *haar like feature* dimanana perlu malakukan training terlebih dahulu untuk mendapatkan sautu pohon keputusan yaitu *cascade calsifier* sebagai pengklasifikasi apakah ada objek yang dideteksi atau tidak dalam setiap frame yang diproses. *Haar like feature* memproses citra dalam sebuah kotak persegi dengan ukuran tertentu misalnya 24 x 24 pixel (Abidin, 2018). Tiap kotak diproses dan menghasilkan perbedaan nilai yang menandakan daerah gelap dan terang. Nilai-nilai inilah yang nantinya dijadikan dasar dalam pemrosesan gambar.

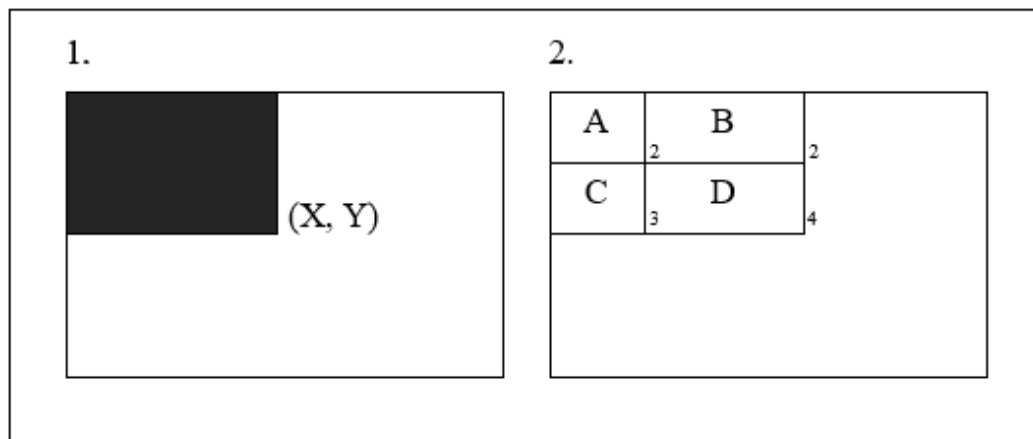
Classifier menggunakan data yang disimpan di file XML untuk menentukan bagaimana mengklasifikasi tiap lokasi gambar, salah satunya bernama *haarcascade\_frontalface\_default.xml*. Secara umum, Haar-Like Feature dipakai dalam mendeteksi objek pada gambar digital. Nama Haar merujuk pada suatu fungsi matematika (Haar Wavelet) yang berbentuk kotak, prinsipnya sama seperti pada fungsi Fourier.

Haar-like feature memproses gambar dalam kotak-kotak, terdapat beberapa pixel di dalam satu kotak. Untuk mencari perbedaan nilai(threshold) yang

menandakan daerah gelap dan terang. Setiap kotak itu diproses. Nilai-nilai inilah yang nantinya dijadikan dasar dalam pemrosesan gambar. Untuk video, perhitungan dan penjumlahan pixel terjadi secara terus menerus dan membutuhkan waktu yang lama. Oleh karena itu, penjumlahan diganti dengan integral sehingga didapatkan perhitungan hasil yang lebih cepat. Untuk hasil yang semakin akurat maka banyak fungsi yang akan dipakai.

### 2.2.2 Integral Image

Untuk menentukan ada tidaknya Haar feature di setiap lokasi gambar, Viola dan Jones memakai teknik yang disebut Integral Image. Integral Image adalah sebuah citra yang nilai tiap pikselnya merupakan penjumlahan dari nilai piksel kiri atas hingga kanan bawah, gambar bisa diintegrasikan sebagai operasi matematika per pixel (Al-Aidid & Pamungkas, 2018).



**Gambar 2.1** Teknik integral image

Seperti yang ditunjukkan oleh gambar di atas setelah pengintegrasian, nilai pada lokasi piksel  $(x,y)$  berisi jumlah dari semua piksel di dalam daerah segiempat dari kiri atas sampai pada lokasi  $(x,y)$  atau daerah yang diarsir. Guna mendapatkan nilai rata-rata piksel pada area segiempat (daerah yang diarsir) ini dapat dilakukan hanya dengan membagi nilai pada  $(x,y)$  oleh area segiempat.

$$ii(x, y) = \sum_{x^1 \leq x, y^1 \leq y} i(x^1, y^1)$$

dimana  $ii(x, y)$  adalah integral image

dan  $i(x^1, y^1)$  adalah original image.

Jumlah pixel dalam kotak D dapat dihitung dengan 4 referensi array. Nilai dari citra integral di lokasi 1 adalah jumlah pixel dalam kotak A. Nilai di lokasi 2 adalah A+B, di lokasi 3 adalah A+C, dan di lokasi 4 adalah A+B+C+D. Jumlah dalam D dapat dihitung sebagai  $4+1-(2+3)$ .

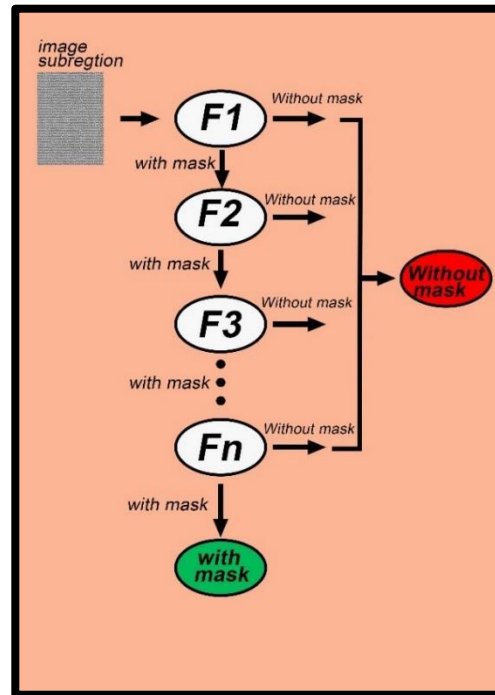
Kemudian untuk menentukan fitur Haar yang spesifik yang akan digunakan dan untuk mengatur nilai ambangnya (threshold) digunakan sebuah metode machine learning yang disebut AdaBoost.

### 2.2.3 AdaBoost

AdaBoost merupakan penggabungan banyak classifier untuk membuat satu classifier yang kuat. Masing-masing classifier menetapkan suatu bobot dan gabungan dari bobot inilah yang akan membentuk satu classifier yang kuat. AdaBoost classifier terpisah yang terdiri classifier lemah atau satu filter Haar (Abidin, 2018). Selama proses pemfilteran, bila ada salah satu filter gagal untuk melewati sebuah daerah gambar, maka daerah itu langsung digolongkan sebagai bukan wajah. Namun ketika filter melewati sebuah daerah gambar dan sampai melewati semua proses filter yang ada dalam rangkaian filter, maka daerah gambar tersebut digolongkan sebagai wajah.

### 2.2.4 Cascade classifier

Bobot yang diberikan AdaBoost menentukan urutan filter pada cascade (Abidin, 2018). Proses pertama ditentukan dari filter dengan bobot paling besar, bertujuan untuk menghapus daerah gambar bukan wajah secepat mungkin.



**Gambar 2. 2** Proses *Cascade Classifier*

Pada tingkat pertama klasifikasi, setiap subcitra diklasifikasi dengan satu fitur haar. Pada klasifikasi pertama hasil berupa True jika gambar memenuhi fitur haar tertentu dan False bila tidak memenuhi fitur haar. Hasil dari klasifikasi kedua berupa True jika gambar memenuhi proses integral image dan False jika tidak. Dan hasil dari klasifikasi berupa True jika gambar memenuhi pada proses AdaBoost dan False jika tidak. Kemudian pada tahapan terakhir yaitu menampilkan gambar yang telah terdeteksi apakah memakai masker atau tidak. Dengan memberi tanda garis kotak di area wajah (region) jika wajah memakai masker maka garis tersebut akan

berwarna hijau dan jika tidak memakai masker maka garis kotak tersebut akan berwarna merah.

### **2.2.5 OpenCV**

OpenCV (Open Computer Vision) adalah sebuah API (Application Programming Interface) Library yang sudah sangat familiar pada Pengolahan Citra Computer Vision. Computer Vision itu sendiri adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (Image Processing) yang memungkinkan komputer dapat melihat seperti manusia. Dengan vision tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa pengimplementasian dari Computer Vision adalah Face Detection, Face Recognition, Face/Object Tracking, Road Tracking, dll. OpenCV adalah library Open Source untuk Computer Vision untuk C/C++, OpenCV didesain untuk aplikasi real-time, memiliki fungsi-fungsi akuisisi yang baik untuk gambar atau video.

OpenCV memiliki lebih dari 2500 algoritma pengoptimalan, termasuk satu set lengkap algoritma pembelajaran komputer dan pembelajaran mesin 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 (Aprilian Anarki et al., 2021).

### **2.2.6 Computer Vision**

Computer vision dapat didefinisikan dengan pengertian pengolahan citra yang dikaitkan dengan akuisisi citra, pemrosesan, klasifikasi, penganan, dan

pencakupan keseluruhan, pengambilan keputusan yang diikuti pengidentifikasian citra. Inti dari teknologi Computer Vision adalah untuk menduplikasi kemampuan penglihatan manusia ke dalam benda elektronik sehingga benda elektronik dapat memahami dan mengerti arti dari gambar yang dimasukkan (Prabowo et al., 2018).

### **2.3 Python**

Menurut penelitian (Zulkhaidi et al., 2020) python adalah salah satu bahasa pemrograman tingkat tinggi yang bersifat interpreter, interactive, object-oriented, dan dapat beroperasi hampir di semua platform: Mac, Linux, dan Windows. Python termasuk bahasa pemrograman yang mudah dipelajari karena sintaks yang jelas, dapat dikombinasikan dengan penggunaan modul siap pakai, dan struktur data tingkat tinggi yang efisien. Distribusi Python dilengkapi dengan suatu fasilitas seperti shell di Linux. Lokasi penginstalan Python biasa terletak di “/usr/bin/Python”, dan bisa berbeda.

### **2.4 Masker**

Masker adalah perlindungan pernafasan yang digunakan sebagai metode untuk melindungi individu dari menghirup zat-zat bahaya atau kontaminan yang berada di udara, perlindungan pernafasan atau masker tidak dimaksudkan untuk menggantikan metode pilihan yang dapat menghilangkan penyakit, tetapi digunakan untuk melindungi secara memadai pemakainya (Cohen & Birdner, 2012).

### **2.5 Deteksi wajah**

Menurut penelitian (Syafira, 2017) deteksi wajah merupakan salah satu teknologi yang sering dimanfaatkan dan selalu dikembangkan seiring dengan perkembangan teknologi komputer. Saat ini banyak aplikasi komersial yang

menggunakan algoritma pendeteksian wajah. Penelitian mengenai teknologi deteksi wajah maupun pengenalan wajah perlu dikembangkan lebih lanjut agar memperoleh hasil yang optimal. Kecepatan dan akurasi sistem pendeteksian wajah harus selalu ditingkatkan.

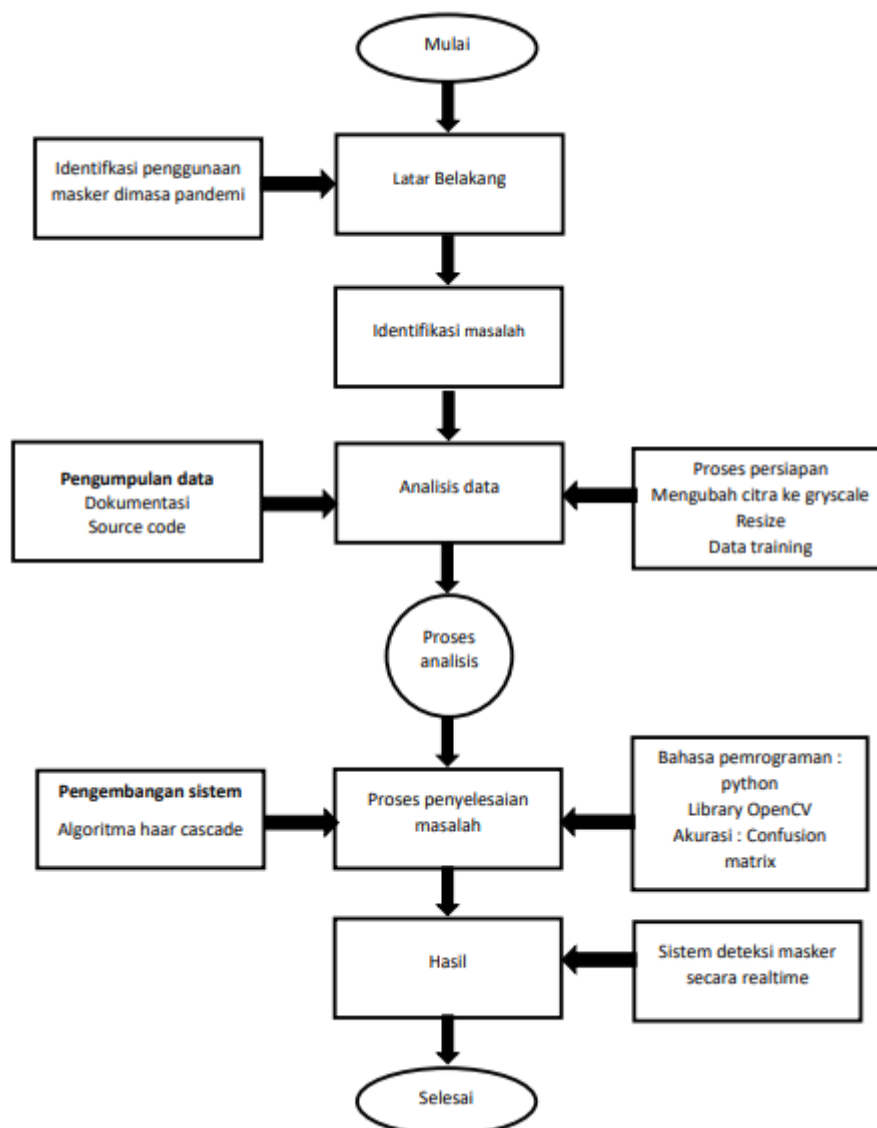
## **2.6 Confusion Matrix**

Confusion matrix adalah sebuah metode yang dapat digunakan untuk mengukur kinerja suatu klasifikasi. Confusion matrix mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh system dengan hasil klasifikasi yang seharusnya, seperti hasil klasifikasi yang seharusnya. terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi Keempat istilah tersebut adalah True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN). Nilai True Negative (TN) merupakan jumlah data negatif yang terdeteksi dengan benar, sedangkan False Positive (FP) merupakan data negatif namun terdeteksi sebagai data positif (Karsito & Susanti, 2019).

## BAB III METODOLOGI PENELITIAN

### 3.1 Kerangka Penelitian

Kerangka penelitian atau garis besar penelitian ini menjelaskan alur penelitian menggunakan gambar sesuai dengan tahapan yang dilakukan dalam penelitian ini. Berikut ini adalah kerangka penelitian yang diajukan oleh peneliti dapat dilihat pada Gambar 3.1:

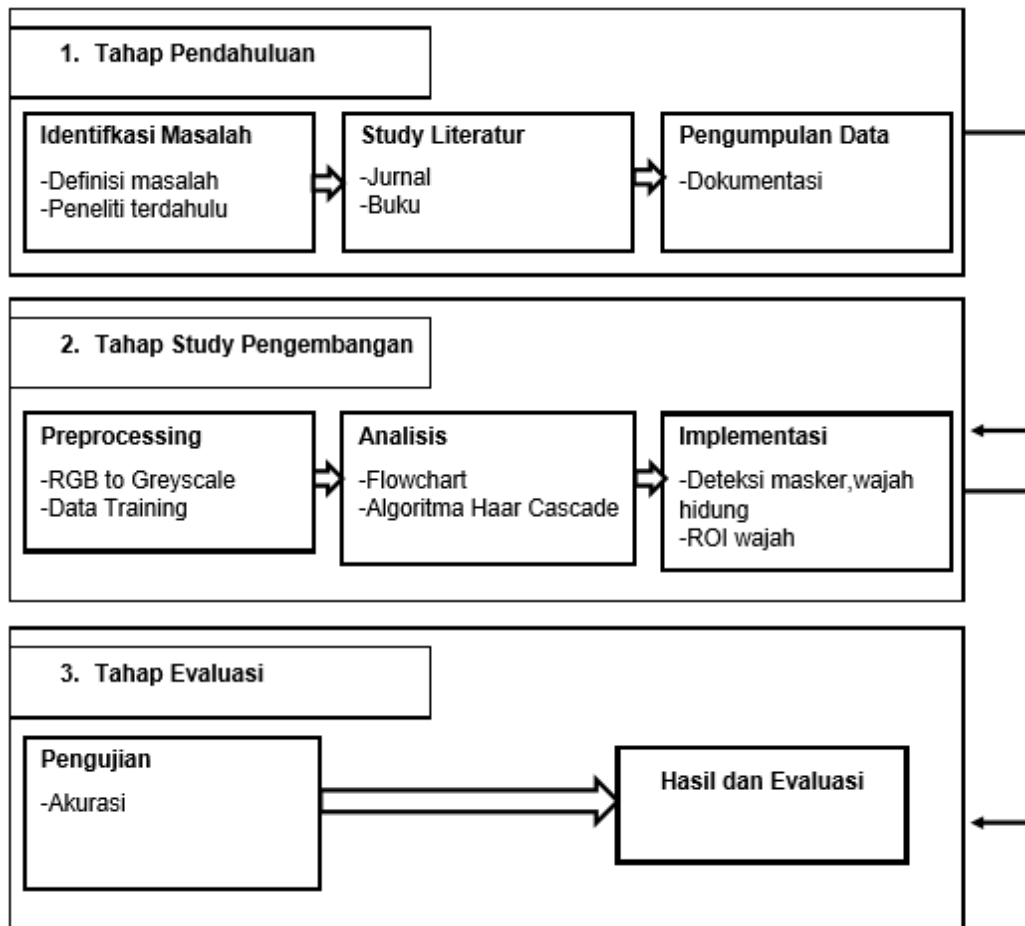


**Gambar 3. 1** Kerangka penelitian



### 3.2 Tahapan Penelitian

Tahapan penelitian yaitu mencakup langkah – langkah dari pelaksanaan sampai dengan akhir penelitian. Pada penelitian ini terdapat beberapa tahapan yang dilakukan, bisa dilihat pada Gambar 3.2 berikut:



**Gambar 3. 2** Tahapan Penelitian

#### 3.2.1 Identifikasi Masalah

Pada tahap ini peneliti akan mencari tahu masalah apa saja yang terjadi pada objek penelitian. Berdasarkan latar belakang yang menjadi masalah pada penelitian ini adalah bagaimana mendeteksi seseorang yang memakai masker atau tidak menggunakan masker dan bagaimana menerapkan algoritma haar cascade untuk mendeteksi apakah seseorang memakai masker atau tidak.

### **3.2.2 Studi Literatur**

Studi literatur merupakan proses pengumpulan teori, informasi, dan data pendukung yang dibutuhkan untuk melakukan penilitan ini. Dilakukan dengan mempelajari jurnal, karya ilmiah, atau buku yang berkaitan dengan penelitian ini seperti informasi tentang algoritma haar cascade.

### **3.2.3 Pengumpulan Data**

Sistem yang dibangun menggunakan bahasa pemrograman Python dengan dataset berupa gambar wajah yang memakai masker yang digunakan sebagai data training dan testing.

#### **3.2.3.1 Dokumentasi**

Pengambilan gambar didapatkan dari gambar yang tersedia di internet. Gambar yang di ambil yaitu gambar orang yang memakai masker dan yang tidak memakai masker.

### **3.2.4 Preprocessing**

Pada tahap preprocessing gambar dilakukan proses scaling dan grayscaling untuk menjadi inputan dalam Haar Cascade.

#### **3.2.4.1 Tahap Greyscaling**

Gambar yang sudah melalui proses scaling kemudian diubah menjadi citra dua warna dengan malakukan proses grayscaling. Proses perubahan gambar RGB menjadi grayscale bisa dengan cara membagikan total nilai pada piksel di bagi 3.

$$\text{Gray} = (R + G + B)/3$$

### **3.2.4.2 Data Training**

Data training adalah data yang digunakan untuk melakukan ekstraksi fitur kemudian dibentuk menjadi classifier untuk melakukan klasifikasi pada gambar. Data training yaitu dataset berisi preprocessed gambar grayscale wajah memakai masker dan wajah tidak memakai masker yang memiliki resolusi 24 x 24 piksel dengan format jpg. Pada penelitian ini training data menggunakan cascade trainer GUI. Data yang digunakan dibagi menjadi 2 yaitu data negatif dan data positif, data negatif berupa gambar wajah yang tidak menggunakan masker sedangkan data positif berupa gambar wajah yang menggunakan masker. Data negatif sebanyak 300 gambar dan data positif sebanyak 300 gambar.

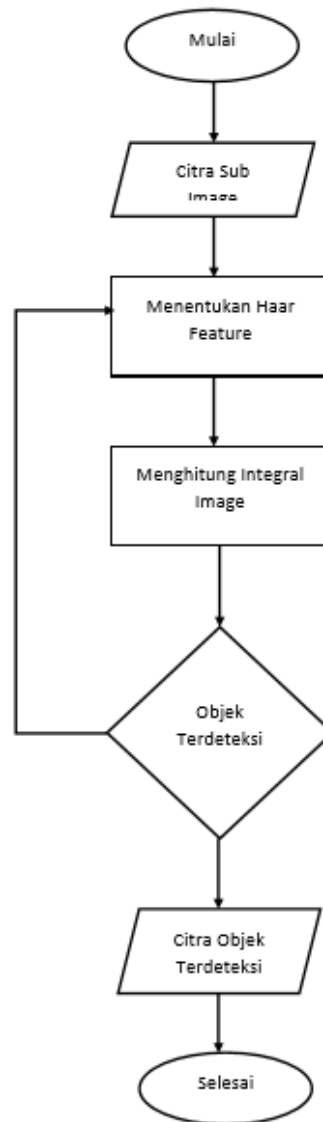
### **3.2.5 Analisis sistem**

Pada sistem yang akan di buat menggunakan bahasa pemrograman Python dan menggunakan library OpenCV, citra yang di inputkan berupa video dan gambar, video di ambil langsung dari web cam atau kamera laptop sedangkan gambar diambil dari citra latih. Citra latih berupa gambar yang telah di siapkan citra dengan objek memakai masker. Jumlah citra yang menggunakan masker 150 kemudian dimasukan ke dalam klasifikasi, disini klasifikasi menggunakan software cascade training GUI kemudian akan menghasilkan file berupa XML berisi fitur dari citra masker. Kemudian nilai fitur dilakukan pencocokan untuk membedakan apakah objek menggunakan masker atau tidak.

#### **3.2.5.1 Analisa Prinsip Kerja Haar Cascade**

Alur proses sistem deteksi wajah dapat dilihat pada Gambar 3.3. Pertama yang harus dilakukan dalam metode ini adalah mengubah gambar masukan menjadi representasi gambar baru berupa gambar integral. Gambar integral terdiri dari sejumlah rangkaian fitur yang cukup banyak. Fitur-fitur tersebut kemudian

diseleksi oleh AdaBoost untuk dijadikan sebagai komponen classifier yang nantinya digunakan untuk mengklasifikasi gambar. Setelah itu AdaBoost akan melakukan training pada classifier yang telah dibentuk. Pada akhirnya, gambar akan diklasifikasi secara bertahap (cascaded) oleh classifier tersebut.

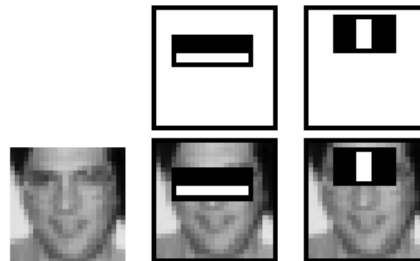


**Gambar 3. 3** Flowchart Metode Haar Cascade

#### **A. Haar Feature**

Fitur Haar merupakan fitur yang digunakan pada metode Viola Jones. Fitur ini terdiri dari satu nilai interval tinggi dan satu nilai interval rendah; untuk gambar

dua dimensi disebut sebagai daerah terang dan daerah gelap. Fitur Haar ditentukan dengan cara mengurangi rata-rata piksel pada daerah gelap dari rata-rata piksel pada daerah terang.

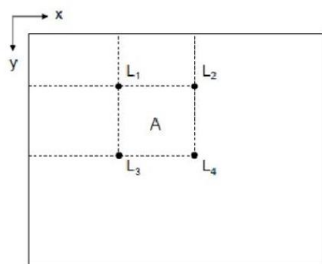


**Gambar 3. 4** Fitur Haar

Untuk memperoleh perbedaan kondisi tingkat kecerahan, maka seluruh citra harus dalam bentuk nilai rata-rata yang telah dinormalisasikan dari variasi sebelumnya.

### B. Integral Image

Gambar integral digunakan untuk menghitung jumlah semua piksel di dalam suatu persegi panjang dengan hanya menggunakan empat nilai secara efisien. Nilai-nilai tersebut adalah piksel pada gambar integral yang bertepatan dengan sudut-sudut persegi panjang pada gambar masukan.



**Gambar 3. 5** Representasi Integral Image

Rumus untuk menghitung piksel A adalah:

$$A = L_4 - (L_2 + L_3) + L_1$$

### C. AdaBoost

Rangkaian filter pada AdaBoost cukup efisien untuk menggolongkan daerah pada suatu gambar. Rangkaian filter tersebut terdiri dari AdaBoost classifier yang terbentuk dari gabungan classifier lemah. Classifier lemah tersebut menetapkan suatu bobot sehingga apabila digabungkan akan menjadi satu classifier yang kuat. Untuk memaksimalkan AdaBoost pada sistem dengan cara menentukan classifier lemah dengan mengevaluasi setiap fitur pada semua data training untuk menemukan fitur dengan kinerja terbaik.

#### **D. Cascade Classifier**

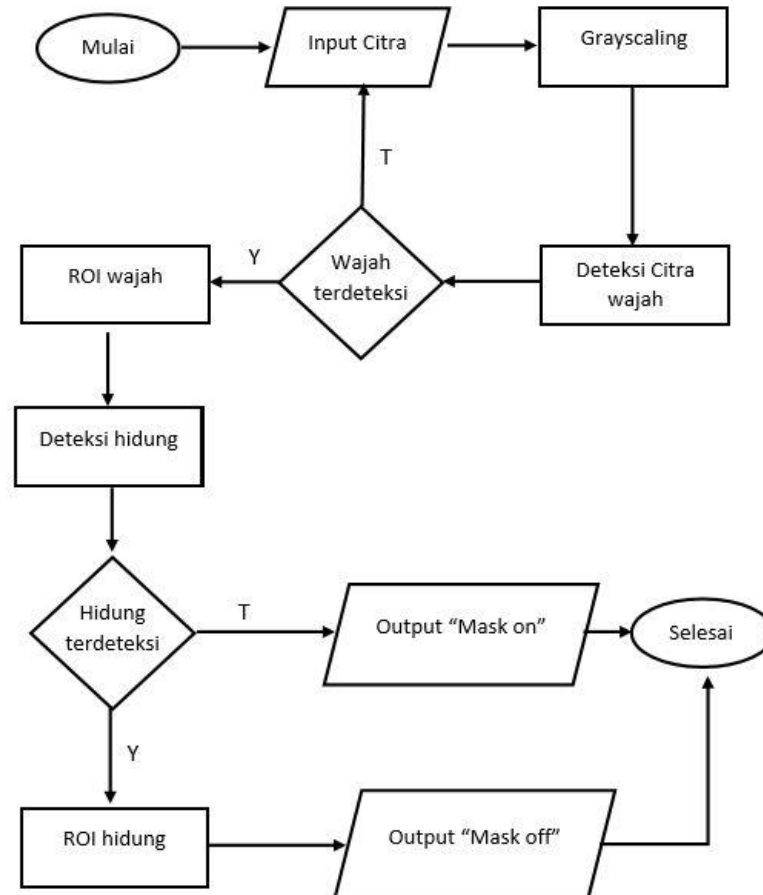
Cascade Classifier adalah metode klasifikasi bertingkat yang bertugas untuk menolak area gambar yang tidak terdeteksi wajah dengan menggunakan classifier yang telah dilatih oleh algoritma AdaBoost pada tiap tingkatan klasifikasinya. Klasifikasi tingkat pertama, setiap inputan berupa subwindow yang akan diklasifikasi secara sederhana. Seiring dengan bertambahnya tingkatan klasifikasi, maka diperlukan syarat yang lebih spesifik sehingga classifier / filter yang digunakan menjadi lebih kompleks. Hal ini dapat mengurangi kasus wajah terdeteksi sebagai bukan wajah.

#### **3.2.6 Perancangan Sistem**

Program ini dikembangkan dengan menggunakan bahasa pemrograman Python dengan menggunakan Library OpenCV, pada program ini pengambilan citra secara *Realtime* atau secara langsung menggunakan web Cam yang kemudian dilakukan pengolahan pada tiap framenya.

Pada tahap akhir dilakukan pendeteksian apakah wajah menggunakan masker atau tidak, jika wajah terdeteksi tidak menggunakan masker maka program

akan mengeluarkan teks “Mask off” dan wajah akan di tandai dengan ROI, jika wajah terdeteksi menggunakan masker maka akan mengeluarkan teks “Mask on”.



**Gambar 3. 6** Diagram alir Program

### 3.2.6.1 Input Citra

Tahap pertama dalam proses deteksi wajah menggunakan masker ini adalah memasukan citra berupa citra video secara langsung. Pada citra masukan awal ini berupa citra RGB. Citra RGB merupakan citra berwarna dari kumpulan piksel yang terdiri dari 3 warna yaitu merah, hijau, dan biru. Citra RGB mempunyai nilai dari 0 (hitam) sampai 255 (putih).

### **3.2.6.2 Grayscale**

Pada proses merubah citra awal yang tadinya RGB kemudian diubah menjadi citra Grayscale. Citra Greyscale atau yang biasa disebut citra keabuan dari kumpulan piksel yang terdiri dari 2 warna yaitu hitam dan putih. Citra Greyscale mempunyai nilai dari 0 (hitam) sampai 255 (putih).

### **3.2.6.3 Deteksi Wajah**

Program akan menjalankan fungsi untuk mendeteksi wajah. Pada proses ini program mendeteksi apakah didalam masukan citra terdapat objek wajah atau tidak. Untuk dapat mendeteksi wajah secara realtime disini penulis menggunakan program yang sudah di sediakan berformat XML yang bersumber dari: [https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade\\_frontalface\\_default.xml](https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml).

### **3.2.6.4 ROI**

ROI (Region of Interest) berupa bujursangkar yang menandai wajah jika terdapat wajah yang terdeteksi. Dalam penelitian ini jika masker terdeteksi maka akan ditandai dengan ROI berwarna hijau dan jika masker tidak terdeteksi maka akan ditandai berwarna merah.

### **3.2.6.1 Deteksi Hidung**

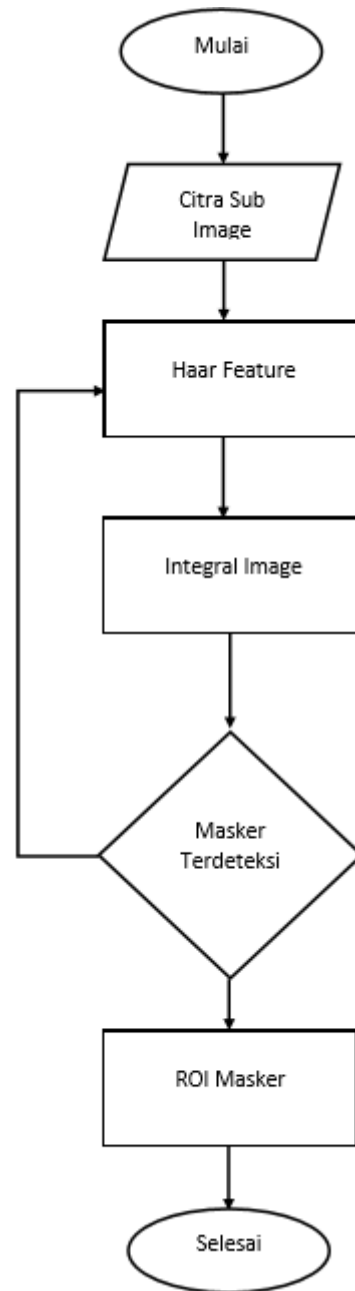
Pada proses deteksi hidung menggunakan program yang sudah di kembangkan yang bersumber dari <https://github.com/sightmachine/SimpleCV/blob/master/SimpleCV/Features/HaarCascades/nose.xml>. Untuk dapat mendeteksi hidung, fungsi deteksi hidung pada penelitian ini adalah untuk menentukan apakah seseorang menggunakan masker



atau tidak, jika hidung terdeteksi dan wajah terdeteksi maka akan ditandai dengan ROI berwarna merah dan muncul peringatan teks “Masker Off”, jika masker terdeteksi dan hidung terdeteksi maka dianggap tidak menggunakan masker.

### **3.2.6.5 Deteksi Masker**

Proses dimana menentukan apakah terdapat masker atau tidak. Pada OpenCV banyak pengklasifikasi terlatih yang dapat digunakan kita dapat menggunakan pengklasifikasi terlatih wajah frontal untuk dapat mendeteksi wajah. Pada pengklasifikasi ini, terdapat modul detectMultiScale. Pada detectMultiScale fungsi ini dapat mengembalikan persegi panjang dengan koordinat (x, y, w, h) mengelilingi wajah yang diinginkan.



**Gambar 3. 7** Flowchart Deteksi Masker

### 3.2.7 Skema Pengujian

Pengujian merupakan tahap uji coba pada sistem yang telah dibuat apakah sesuai yang di harapkan atau tidak. Pada tahapan ini dilakukan beberapa pengujian

untuk mengetahui akurasi pada sistem yang dibuat dan untuk mengetahui apakah sistem berjalan dengan baik

### 3.2.7.1 Pengujian Fungsional

Pada pengujian fungsional melakukan uji pada fitur yang terdapat pada aplikasi sesuai dengan kebutuhan dan rancangan pada aplikasi seperti pada table 3.1.

**Tabel 2.1** Pengujian Fungsional

No	Fitur	Hasil
1	Deteksi Masker	
2	Deteksi Wajah	
3	Deteksi Hidung	
4	Membuat Tanda Bungkai Persegi pada Objek	
5	Memunculkan Peringatan berupa teks	

### 3.2.7.2 Pengujian Berdasarkan Sudut dan Jarak

Pada pengujian ini dilakukan berdasarkan kondisi yaitu jarak dan sudut. Pertama untuk pengujian berdasarkan sudut pengambilan citra wajah yaitu tegak lurus, rotasi 30° kekanan, rotasi 30° kekiri, mengangkat kepala 15° dan menunduk 15°. Kemudian pengujian berdasarkan jarak wajah terhadap kamera yaitu 30-50 cm dan 100 cm.

**Tabel 3.2** Pengujian berdasarkan Sudut dan Jarak

No	Jarak	Sudut	Keterangan
1	50-100 cm	Tegak lurus	

2	50-100 cm	30° kekanan	
3	50-100 cm	30° kekiri	
4	50-100 cm	Menghadap keatas 15°	
5	50-100 cm	Menghadap kebawah 15°	
6	100-200 cm	Tegak lurus	
7	100-200 cm	30° kekanan	
8	100-200 cm	30° kekiri	
9	100-200 cm	Menghadap kebawah 15°	
10	100-200 cm	Menghadap keatas 15°	

### 3.2.7.3 Pengujian Berdasarkan Citra Gambar

Pada pengujian ini dilakukan deteksi menggunakan citra gambar yang sudah tersedia, citra tersebut berupa gambar seseorang menggunakan masker. Pada tahapan ini data diuji berdasarkan jumlah gambar, jenis kelamin, warna masker, jenis masker dan jumlah gambar yang akan terdeteksi.

### 3.2.7.4 Pengujian Menggunakan *Confusion Matrix*

*Confusion Matrix* adalah pengukuran performa untuk masalah klasifikasi *machine learning* dimana keluaran dapat berupa dua kelas atau lebih. *Confusion matrix* merupakan salah satu *tools* analitik prediktif yang menampilkan dan membandingkan nilai aktual dengan nilai hasil prediksi model yang dapat digunakan untuk menghasilkan metrik evaluasi seperti akurasi, *Precision*, *Recal.* Terdapat 4 kombinasi pada *confusion matrix* dari nilai prediksi dan nilai aktual yang berbeda. Dapat dilihat pada gambar berikut:

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	<b>TP</b> (True Positive)	<b>FP</b> (False Positive) <small>Type I Error</small>
	0 (Negative)	<b>FN</b> (False Negative) <small>Type II Error</small>	<b>TN</b> (True Negative)

**Gambar 3. 8** *Confusion Matrix*

Ada 4 istilah sebagai representasi hasil dari proses klasifikasi pada *confusion matrix*. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN).

Untuk mengetahui *performance metrics* kita dapat menggunakan beberapa *performance metrics* yang sering digunakan yaitu Accuracy, Precision dan Recall. Untuk mendapatkan nilai Accuracy, Precision dan Recall dapat di peroleh dengan persamaan berikut:

*Accuracy* adalah tingkat kedekatan nilai prediksi dengan nilai sebenarnya (actual). Nilai *accuracy* dapat diperoleh dengan persamaan berikut.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

*Precision* adalah rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Dari seluruh kelas positif yang telah di prediksi dengan benar, berapa banyak data yang benar-benar positif. Nilai *precision* dapat diperoleh menggunakan persamaan berikut.

$$precision = \frac{TP}{TP + FP}$$

*Recall* atau *sensitivity* mendeskripsikan keberhasilan model dalam menemukan kembali sebuah informasi. Maka, *recall* adalah rasio prediksi benar

positif dibandingkan dengan keseluruhan data yang benar positif. Nilai *recall* bisa diperoleh menggunakan persamaan berikut.

$$recall = \frac{TP}{TP + FN}$$

## BAB IV HASIL DAN PEMBAHASAN

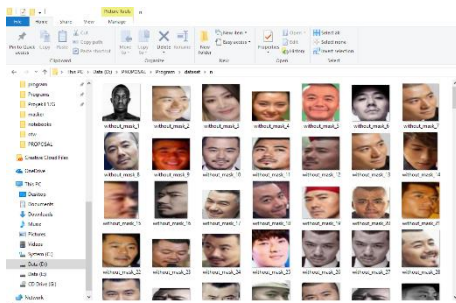
### 4.1 Preprocessing

Pada tahapan preprocessing terdapat 2 tahapan yaitu tahap *greyscaling* dan training. Pada tahap *greyscaling* dilakukan perubahan warna citra yang awalnya RGB menjadi hitam putih kemudian pada tahapan training dilakukan klasifikasi pada dataset citra menggunakan software Cascade Trainer GUI.

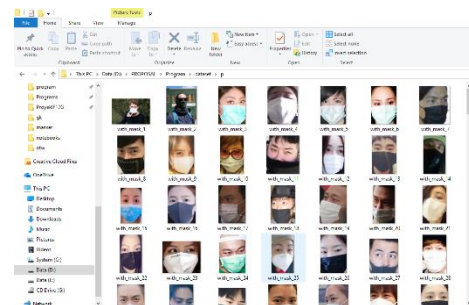
#### 4.1.1 Training

Pada tahapan ini, akan dilakukan proses training menggunakan Cascade Trainer GUI. Dataset yang digunakan diambil dari <https://www.kaggle.com/omkargurav/face-mask-dataset>. Data yang digunakan di bagi menjadi tipe data negatif (n) dan data positif (p).

Data negatif berupa gambar wajah yang tidak menggunakan masker sedangkan data positif berupa gambar wajah yang menggunakan masker. Data negatif sebanyak 291 gambar dan data gambar positif sebanyak 253 gambar. Kumpulan data positif dan negatif dapat dilihat pada gambar 4.2 dan 4.3.

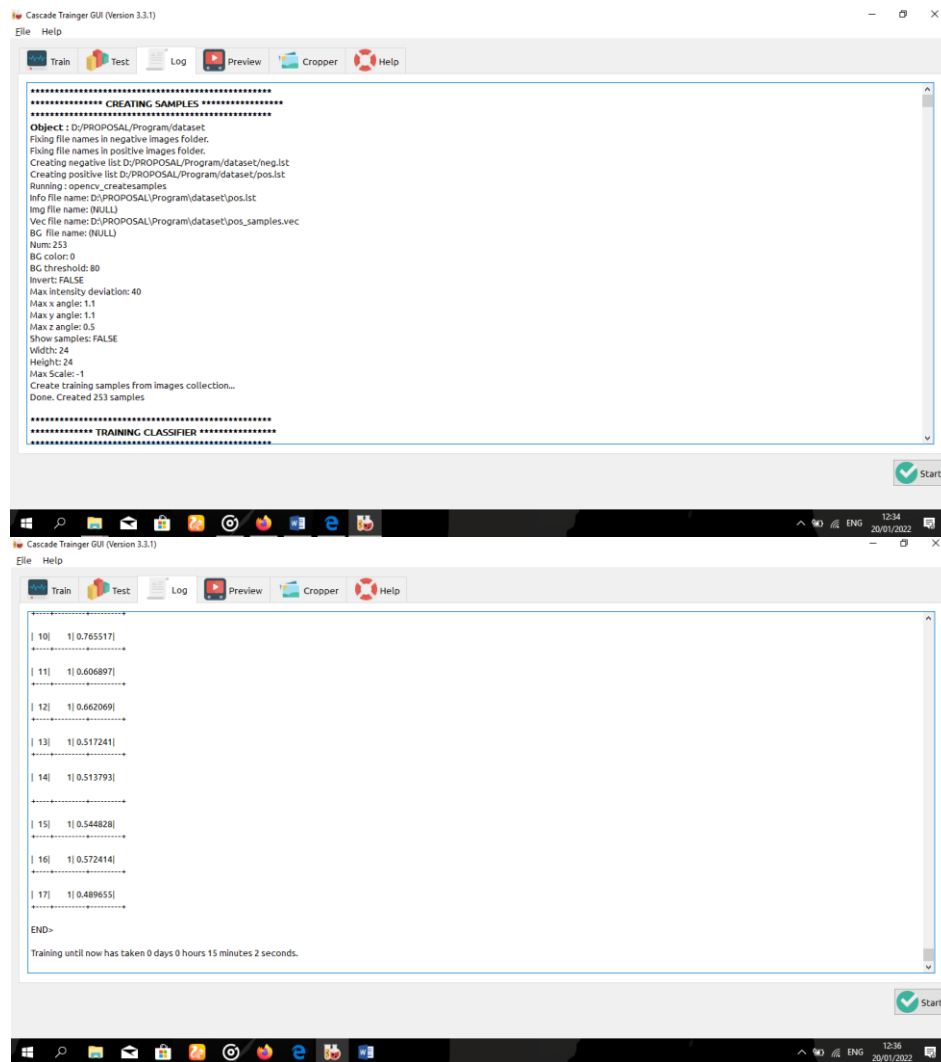


**Gambar 4. 1** Data negatif



**Gambar 4. 2** Data positif

Proses training data menggunakan Cascade Trainer GUI membutuhkan waktu yang cukup lama, tergantung dari seberapa baik perangkat yang digunakan dan seberapa banyak dataset yang digunakan. Tampilan proses training dapat dilihat pada gambar 4.3.



**Gambar 4. 3** Proses training

Pada proses training menghasilkan file cascade berformat .XML yang berisi fitur citra dari masker.



## 4.2 Implementasi Program

Pada tahapan ini menampilkan kode program yang telah dibuat untuk mendeteksi wajah memakai masker.

### 4.2.1 Input Citra

Untuk dapat mendeteksi wajah memakai masker diperlukan citra masukan terlebih dahulu, pada penelitian ini citra masukan di ambil menggunakan kamera secara langsung (*Realtime*) untuk dapat mengambil gambar secara langsung di perlukan kode sebagai berikut.

```
[3]:  
cap = cv2.VideoCapture(1)  
mask_on = False
```

**Gambar 4. 4** Kode input citra

### 4.2.2 Grayscale

Masukan citra yang di peroleh perlu di ubah terlebih dahulu menjadi citra grayscale terlebih dahulu menggunakan kode berikut.

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

**Gambar 4. 5** Kode merubah citra grayscale

### 4.2.3 Deteksi Wajah

Pada tahapan ini dilakukan perulangan untuk mendeteksi apakah terdapat wajah atau tidak. Untuk dapat mendeteksi wajah atau hidung di perlukan kode program yang sudah di sediakan berformat XML, kemudian dilakukan pemanggilan dari kode program tersebut.

```

face_mask = cv2.CascadeClassifier("haarcascade/cascade1.xml")
faceCascade = cv2.CascadeClassifier("haarcascade/haarcascade_frontalface_default.xml")
noseCascade = cv2.CascadeClassifier("haarcascade/Nose.xml")

```

**Gambar 4. 6** Kode program cascade

```

-
wajah = faceCascade.detectMultiScale(gray, 1.1, 5)
for(x, y, w, h) in wajah:
    →roi_gray = gray[y:y+h, x:x+w]
    →roi_color = img[y:y+h, x:x+w]
    →if mask_on:(
    →→cv2.rectangle(img, (x,y), (x+w, y+h), (0, 255, 0), 1),
    →→cv2.putText(img, 'Mask On', (x,y), 1, 2, (0,255,0), 2),
    -----
    )
    →else:(
    →→cv2.rectangle(img, (x,y), (x+w, y+h), (0,0,255),1),
    →→cv2.putText(img, 'Mask Off', (x,y), 1, 2, (0,0,255), 2),
    -----
    |
    )

```

**Gambar 4. 7** Kode deteksi wajah dan ROI

Setelah dilakukan deteksi jika wajah terdeteksi maka akan di tandai dengan ROI pada wajah. Pada ROI wajah jika terdeteksi wajah tidak menggunakan masker maka akan ditandai dengan garis berwarna merah dan terdapat tulisan Mask off diatas, kemudian jika wajah terdeteksi menggunakan masker maka akan ditandai dengan garis berwarna hijau dan terdapat tulisan Mask On.

#### 4.2.4 Deteksi Hidung

Setelah wajah dideteksi untuk menentukan apakah seseorang menggunakan masker dengan cara mendeteksi hidung pada wajah, jika hidung terdeteksi maka dianggap tidak menggunakan masker jika hidung pada wajah tidak terdeteksi maka dianggap menggunakan masker. Untuk dapat mendeteksi hidung sama seperti mendeteksi wajah yaitu di perlukan kode program yang sudah di sediakan berformat XML untuk mendeteksi hidung.

```

hidung = noseCascade.detectMultiScale(gray, 1.18, 35,)
for(sx, sy, sw, sh) in hidung:
    cv2.rectangle(img, (sx,sy), (sx+sw, sy+sh), (255,0,0),2),
    cv2.putText(img, 'Hidung', (sx,sy), 1, 1, (255,0,0), 1)

```

**Gambar 4. 8** Kode deteksi hidung, ROI

Setelah hidung terdeteksi maka akan di tandai dengan ROI berwarna biru dan terdapat tulisan hidung.

#### 4.2.5 Deteksi masker

Pada penelitian ini juga dilakukan pendeteksian masker yang dikenakan untuk memastikan apakah seseorang menggunakan masker.

```

face = face_mask.detectMultiScale(gray,1.1,4)
for(x,y,w,h) in face:
    roi_gray = gray[y:y+h, x:x+w]
    roi_color=img[y:y+h, x:x+w]

    cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),1),
    cv2.putText(img, 'masker', (x,y), 1, 1, (255,0,0), 1),

```

**Gambar 4. 9** Kode deteksi masker, ROI

Jika masker terdeteksi maka akan ditandai dengan ROI berwarna biru dan terdapat tulisan masker.





### 4.3 Hasil Pengujian

Pada tahapan ini dilakukan pengujian untuk mengetahui apakah sistem berjalan sesuai dengan fungsinya dan untuk mengetahui akurasi dari algoritma yang digunakan untuk melakukan deteksi wajah memakai masker. Pada penelitian ini terdapat 4 pengujian yang dilakukan.

#### 4.3.1 Pengujian Fungsional

Pada pengujian fungsional dilakukan proses uji pada fitur yang terdapat pada aplikasi sesuai dengan kebutuhan dan rancangan.

**Tabel 4.1** Pengujian Fungsional

No	Fitur	Hasil	Keterangan
1	Deteksi Masker		Terdeteksi
2	Deteksi Wajah		Terdeteksi
3	Deteksi Hidung		Terdeteksi
4	Membuat Tanda Bungkai Persegi pada Objek		Terdeteksi

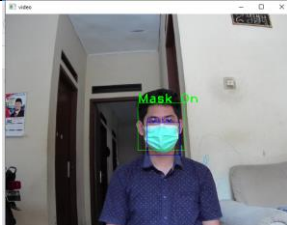
5	Memunculkan Peringatan berupa teks		Terdeteksi
---	------------------------------------	--	------------

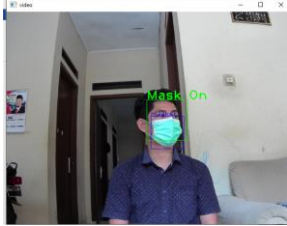
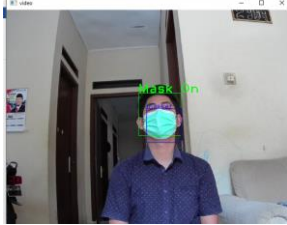
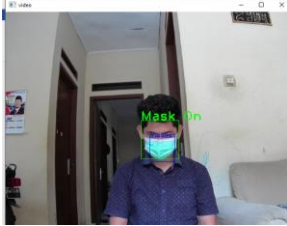


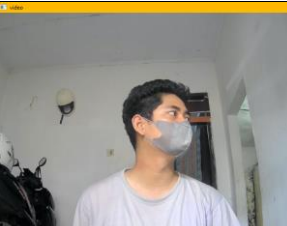
Berdasarkan tabel 4.2 dapat dilihat bahwa keseluruhan fungsional pada sistem deteksi wajah memakai masker dapat bekerja dengan baik.

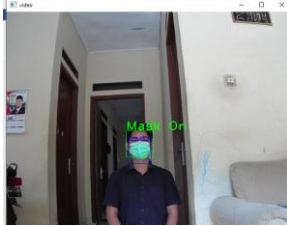
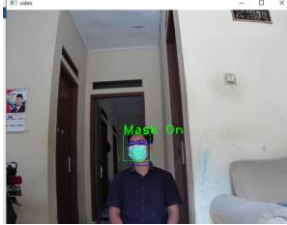
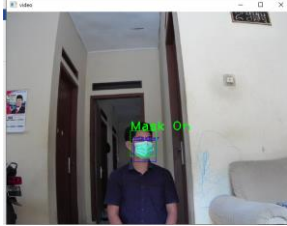
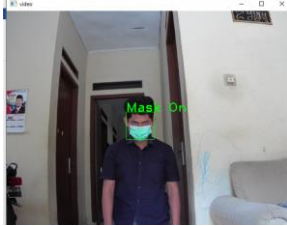

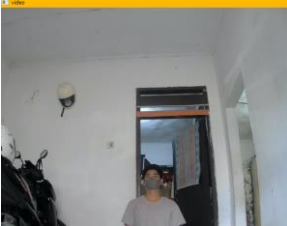
#### 4.3.2 Pengujian Berdasarkan Sudut dan Jarak

Pada pengujian ini dilakukan berdasarkan kondisi yaitu sudut dan jarak. Pertama untuk pengujian berdasarkan sudut pengambilan citra wajah yaitu tegak lurus, rotasi 30° kekanan, rotasi 30° kekiri, mengangkat kepala 15° dan menunduk 15°. Kemudian pengujian berdasarkan jarak wajah terhadap kamera yaitu 30-50 cm dan 100 cm.

**Tabel 4.2** Pengujian Berdasarkan Sudut dan jarak

No	Jarak	Sudut	Hasil	Keterangan
1	20-100 cm	Tegak lurus		Terdeteksi
2	20-100 cm	30° kekanan		Terdeteksi

3	20-100 cm	30° kekiri		Terdeteksi
4	20-100 cm	Menghadap keatas 15°		Terdeteksi
5	20-100 cm	Menghadap kebawah 15°		Terdeteksi
6	<20 cm	Tegak lurus		Tidak Terdeteksi
7	20-100 cm	Menghadap kanan >30°		Tidak Terdeteksi
8	20-100 cm	Manghadap kiri >30°		Tidak Terdeteksi

9	100-200 cm	Tegak lurus		Terdeteksi
10	100-200 cm	30° kekanan		Terdeteksi
11	100-200 cm	30° kekiri		Terdeteksi
12	100-200 cm	Menghadap kebawah 15°		Terdeteksi
13	100-200 cm	Menghadap keatas 15°		Terdeteksi
14	>200 cm	Tegak lurus		Tidak Terdeteksi

Berdasarkan Tabel 5.2 maka dapat di simpulkan sistem deteksi wajah memakai masker ini dapat mendeteksi wajah memakai masker berdasarkan jarak dan posisi objek. Dapat disimpulkan pula batas wajah menghadap kekanan dan kekiri yaitu 30°, untuk jarak maksimum posisi objek yaitu sejauh 200 cm dan jarak minimum objek yaitu sejauh 20 cm.

### 4.3.3 Pengujian Dengan Citra Gambar

Pada pengujian ini proses program akan mendeteksi gambar citra, diantaranya

a. warna masker

Dengan melihat warna masker apa yang sedang digunakan.

b. jenis masker yang dikenakan.

Dengan melihat jenis masker yang digunakan seperti jenis masker KN95, Kain dan Bedah.

Berikut hasil dari pengujian citra gambar dapat dilihat pada Tabel 6.3.

**Tabel 4.3** Pengujian dengan citra gambar

No	Jumlah Gambar	Warna Masker	Jenis Masker	Jumlah Terdeteksi
1	40	Putih	KN95	35
2	28	Hijau	Bedah	22
3	28	Biru	Bedah	23
4	16	Hitam	Kain	7

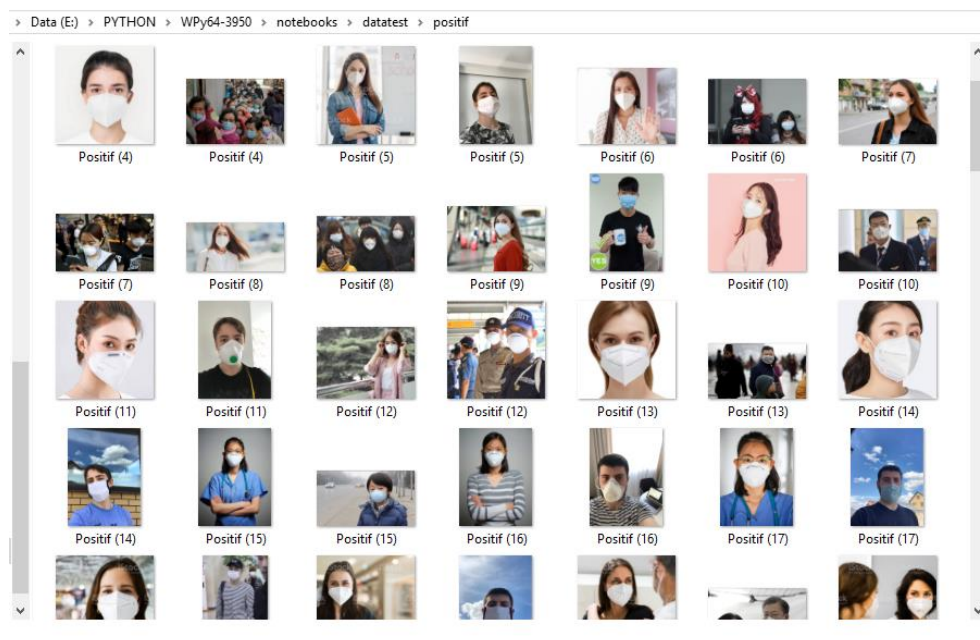


Pada tabel 5 diatas dapat dilihat bahwa total data data uji sebanyak 112 citra dan citra masker yang terdeteksi sebanyak 87 citra (positif).

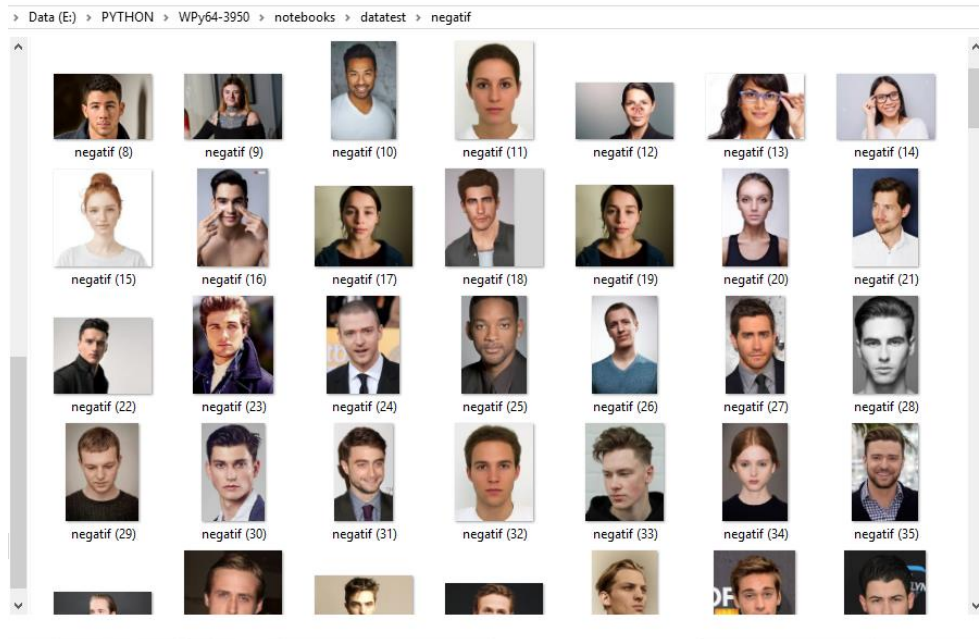
#### 4.3.4 Pengujian Menggunakan *Confusion Matrix*

Untuk mendapatkan model Machine Learning (ML) agar dapat memberikan hasil yang diinginkan maka, diperlukan tahapan evaluasi model yang telah ditraining. Salah satu cara mengetahui kinerja model adalah dengan mengukur tingkat akurasi. Kemudian dalam mengevaluasi performance algoritma, digunakan acuan Confusion Matrix yaitu dengan merepresentasikan prediksi dan kondisi sebenarnya (aktual) dari data yang dihasilkan oleh algoritma ML.

Data test yang digunakan sebanyak 500 data yang terbagi menjadi 217 data positif dan 283 data negatif, kemudian data tersebut dilakukan pengujian.



**Gambar 4. 10** Data Positif



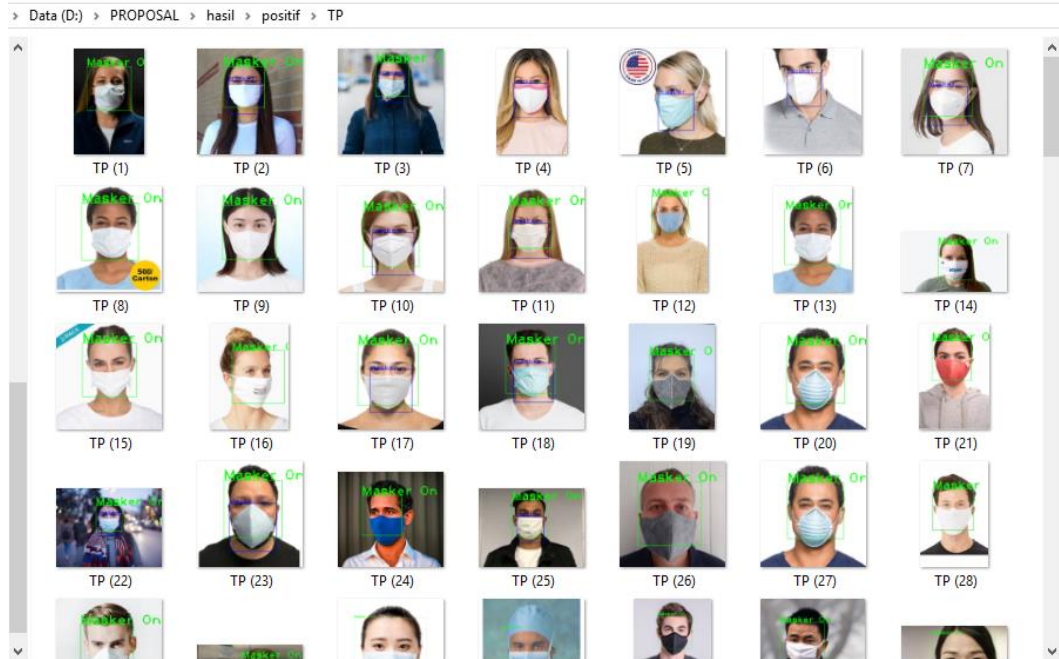
**Gambar 4. 11** Data Negatif

Dari data test tersebut menghasilkan nilai prediksi dan nilai aktual yang dapat dilihat pada Tabel berikut.

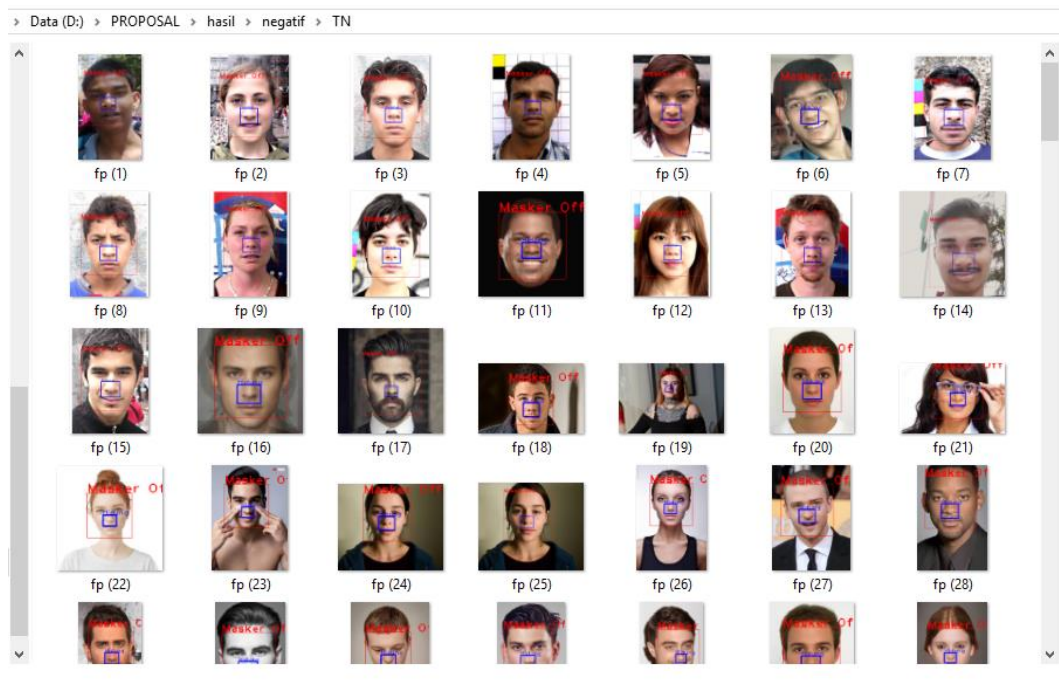
**Tabel 4.4** Confusion matrix

n=500	1(Positif)	0(negatif)
1(Positif)	177	53
0(negatif)	40	230

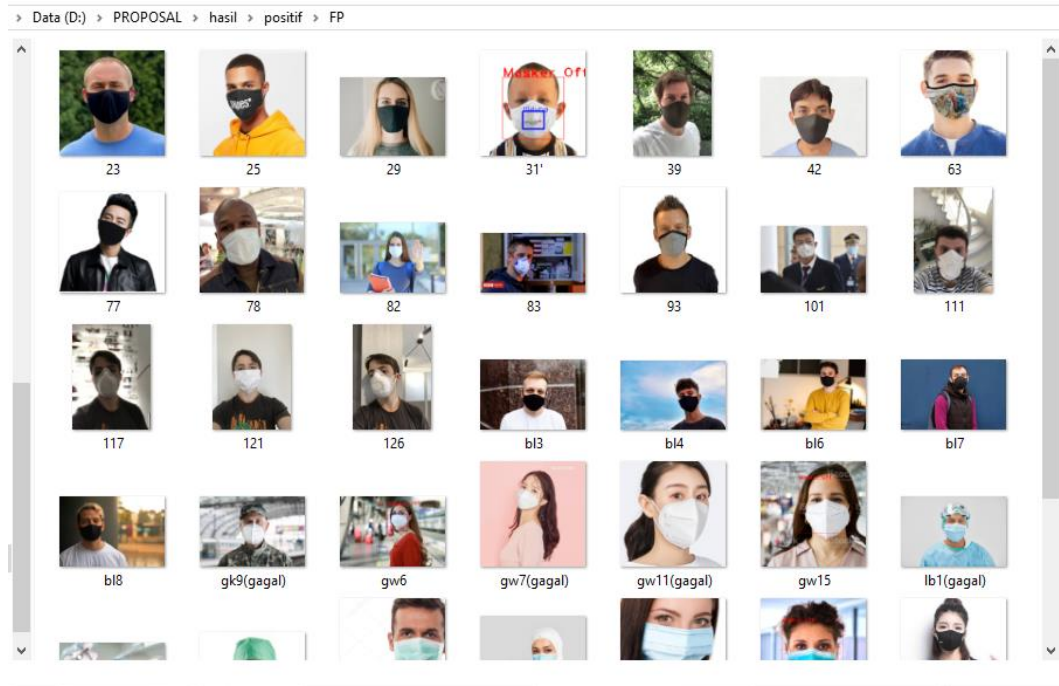
Dapat di lihat pada Tabel Dari data test yang digunakan menghasilkan nilai actual dan predicted, yang terdapat nilai TP = 177, TN = 230, FP = 40 dan FN = 53. Hasil gambar dapat dilihat pada gambar dibawah ini.



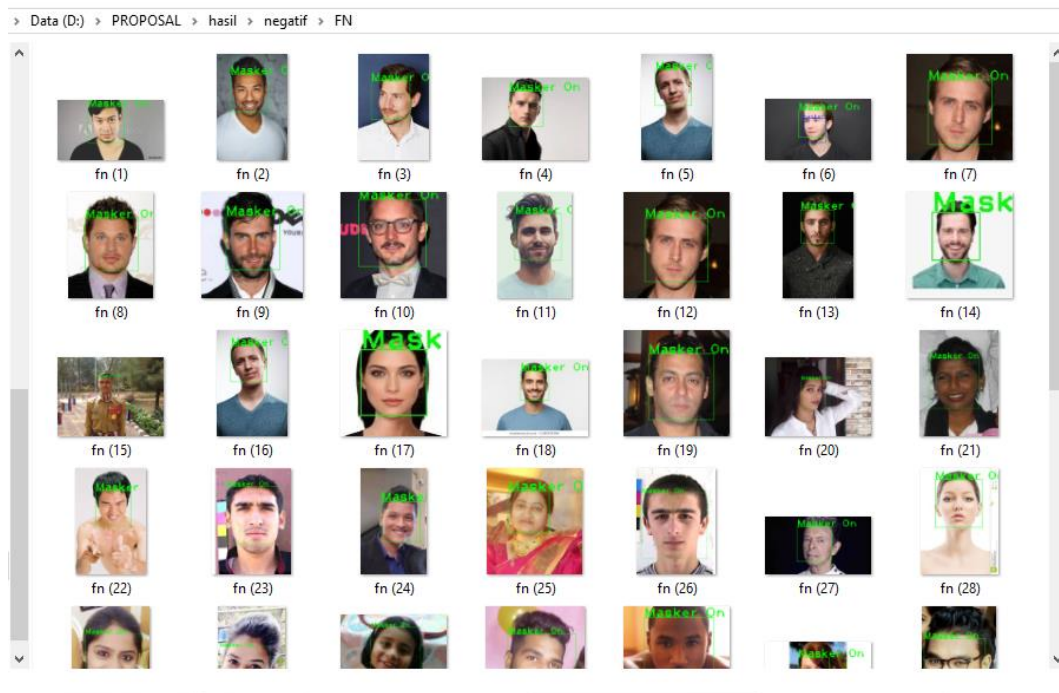
**Gambar 4. 12** *Data True positive*



**Gambar 4. 13** *Data True negative*



**Gambar 4. 14** Data False positive



**Gambar 4. 15** Data False negative

Kemudian dari nilai actual dan predicted dapat menentukan nilai dari Accuracy, Precision dan Recall. Untuk mendapatkan nilai dari Accuracy, Precision dan Recall bisa di dapatkan dengan menggunakan persamaan berikut.

$$accuracy = \frac{177 + 230}{177 + 230 + 40 + 53} = 0.814$$

$$recall = \frac{177}{177 + 40} = 0.815$$

$$precision = \frac{177}{177 + 53} = 0.76$$

Dari perhitungan menggunakan persamaan convu. diatas untuk mencari nilai Racall, Accuracy dan Precision, maka mendapatkan ketiga nilai tersebut dan di simpulkan dalam tabael di bawah ini :

**Tabel 4.5** Hasil Confusion Matrix

Accuracy (%)	Precision (%)	Recall(%)
81.4	76	81.5

## **BAB V PENUTUP**

### **5.1 Kesimpulan**

Berdasarkan pengujian yang telah dilakukan sebelumnya maka dapat diambil kesimpulan diantaranya:

1. Hasil dari penelitian ini adalah dapat mendeteksi wajah memakai masker dengan menggunakan algoritma haar cascade Dan melakukan pengujian dengan menggunakan confusion matrix
2. Algoritma haar cascade terbukti memiliki tingkat akurasi yang baik. Tingkat akurasinya terbukti dengan melakukan pengujian confusion matrix dengan nilai akurasi 81% recall 81.5% dan presisi 76%
3. Dengan menerapkan algoritma haar cascade terbukti dengan baik, dapat digunakan untuk mendeteksi objek khususnya pendeteksian wajah memakai masker seperti pada penelitian ini.

### **5.2 Saran**

Pada penelitian ini masih memiliki kekurangan yang kedepannya bias di kembangkan lebih baik lagi guna mencapai hasil yang optimal. Beberapa saran yang diberikan untuk penelitian selanjutnya, daiataranya:

1. Penelitian ini hanya mengggunakan algoritma haar cascade agar lebih tinggi tingat akurasinya bisa di padukan dengan algoritma atau metode lainnya.
2. Dapat memberikan notifikasi jika ada yang terdeteksi tidak menggunakan masker.

3. Dalam penelitian ini hanya berupa program untuk pendeteksian, diharapkan penelitian selanjutnya dapat berupa aplikasi.
4. Pada penelitian ini deteksi dapat di jalankan secara realtime menggunakan web cam, diharapkan penelitian selanjutnya dapat mendeteksi menggunakan CCTV.
5. Penelitian ini masih pada jarak deteksi masih terbatas, di aharpkan pad penelitian selanjutnya dapat mendeteksi lebih jauh lagi.

## DAFTAR PUSTAKA

- Abidin, S. (2018). Deteksi Wajah Menggunakan Metode Haar Cascade Classifier Berbasis Webcam Pada Matlab. *Jurnal Teknologi Elekterika*, 15(1), 21. <https://doi.org/10.31963/elekterika.v15i1.2102>
- Al-Aidid, S., & Pamungkas, D. (2018). Sistem Pengenalan Wajah dengan Algoritma Haar Cascade dan Local Binary Pattern Histogram. *Jurnal Rekayasa Elekrika*, 14(1), 62–67. <https://doi.org/10.17529/jre.v14i1.9799>
- 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>
- Giancini, D., Puspaningrum, E. Y., & Via, Y. V. (2020). Identifikasi Penggunaan Masker Menggunakan Algoritma CNN YOLOv3-Tiny. *Seminar Nasional Informatika Bela Negara (SANTIKA)*, 1, 153–159.
- Jignesh Chowdary, G., Punn, N. S., Sonbhadra, S. K., & Agarwal, S. (2020). Face Mask Detection Using Transfer Learning of InceptionV3. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12581 LNCS, 81–90. [https://doi.org/10.1007/978-3-030-66665-1\\_6](https://doi.org/10.1007/978-3-030-66665-1_6)
- Karsito, & Susanti, S. (2019). Klasifikasi Kelayakan Peserta Pengajuan Kredit Rumah Dengan Algoritma Naïve Bayes Di Perumahan Azzura Residencia. *Jurnal Teknologi Pelita Bangsa*, 9, 43–48.
- 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>
- Li, C., Qi, Z., Jia, N., & Wu, J. (2017). Human face detection algorithm via Haar cascade classifier combined with three additional classifiers. *ICEMI 2017 - Proceedings of IEEE 13th International Conference on Electronic Measurement and Instruments, 2018-Janua*, 483–487. <https://doi.org/10.1109/ICEMI.2017.8265863>
- Prabowo, D. A., Abdullah, D., & Manik, A. (2018). *BERDASARKAN WARNA MENGGUNAKAN*. V(September), 85–91.
- Septiana, T., Puspita, N., Fikih, M. Al, & Setyawan, N. (2020). Face Mask Detection Covid-19 Using Convolutional Neural Network ( Cnn ). *Seminar Nasional Teknologi Dan Rekayasa (SENTRA) 2020*, 27–32.
- Suhaimin, M. S. M., Hijazi, M. H. A., Kheau, C. S., & On, C. K. (2021). Real-time mask detection and face recognition using eigenfaces and local binary pattern histogram for attendance system. *Bulletin of Electrical Engineering*



*and Informatics*, 10(2), 1105–1113.  
<https://doi.org/10.11591/EEL.V10I2.2859>

Syafira, A. R. (2017). Sistem Deteksi Wajah Dengan Modifikasi Metode Viola Jones. *Emitor: Jurnal Teknik Elektro*, 17(1), 26–33.  
<https://doi.org/10.23917/emitor.v17i1.5964>

Vinh, T. Q., & Anh, N. T. N. (2020). Real-Time Face Mask Detector Using YOLOv3 Algorithm and Haar Cascade Classifier. *Proceedings - 2020 International Conference on Advanced Computing and Applications, ACOMP 2020*, 146–149. <https://doi.org/10.1109/ACOMP50827.2020.00029>

Viola, P., & Jones, M. (2004a). *Rapid Object Detection Using a Boosted Cascade of Simple Features*.

Zulkhaidi, T. C. A.-S., Maria, E., & Yulianto, Y. (2020). Pengenalan Pola Bentuk Wajah dengan OpenCV. *Jurnal Rekayasa Teknologi Informasi (JURTI)*, 3(2), 181. <https://doi.org/10.30872/jurtti.v3i2.4033>

## LAMPIRAN

### Lampiran 1 Kode Program Deteksi Menggunakan Image

```

scale_factor = 1.1
min_neighbors = 3
min_size = (30, 30)

mask_on = False

def detect(path):
    img = cv2.imread(path)
    cascade = cv2.CascadeClassifier("haarcascade/haarcascade_frontalface_alt.xml")
    rects = cascade.detectMultiScale(img, scaleFactor=scale_factor, minNeighbors=min_neighbors,
                                     minSize=min_size)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    if len(rects) >= 0:
        # create the bounding box around the detected face
        wajah = faceCascade.detectMultiScale(gray, 1.1, 5)
        for(x, y, w, h) in wajah:
            roi_gray = gray[y:y+h, x:x+w]
            roi_color=img[y:y+h, x:x+w]

        hidung = noseCascade.detectMultiScale(gray, 1.18, 35,)
        for(x, y, w, h) in hidung:
            roi_gray = gray[y:y+h, x:x+w]
            roi_color=img[y:y+h, x:x+w]

        masker = face_mask.detectMultiScale(gray, 1.1, 4)
        for(x, y, w, h) in masker:
            roi_gray = gray[y:y+h, x:x+w]
            roi_color=img[y:y+h, x:x+w]

        if(len(wajah) > 0 and len(hidung) == 0 and len(masker)>0 ):
            mask_on = True
        elif(len(hidung) == 0):
            mask_on = True
        else:
            mask_on = False

        wajah = faceCascade.detectMultiScale(gray, 1.1, 5)
        for(x, y, w, h) in wajah:
            roi_gray = gray[y:y+h, x:x+w]
            roi_color=img[y:y+h, x:x+w]
            if mask_on:
                cv2.rectangle(img, (x,y), (x+w, y+h), (0, 255, 0), 1),
                cv2.putText(img, 'Masker On', (x,y), 1, 2, (0, 255, 0), 2),
            )
            else:
                cv2.rectangle(img, (x,y), (x+w, y+h), (0, 0, 255), 1),
                cv2.putText(img, 'Masker Off', (x,y), 1, 2, (0, 0, 255), 2),
            )

        hidung = noseCascade.detectMultiScale(gray, 1.18, 35,)
        for(ax, ay, aw, ah) in hidung:
            cv2.rectangle(img, (ax,ay), (ax+aw, ay+ah), (255, 0, 0), 2),
            cv2.putText(img, 'Hidung', (ax,ay), 1, 1, (255, 0, 0), 1)

        masker = face_mask.detectMultiScale(gray, 1.1, 4)
        for(bx,by,bw,bh) in masker:
            cv2.rectangle(img, (bx,by), (bx+bw,by+bh), (255, 0, 0), 1),
            cv2.putText(img, 'masker', (bx,bv), 1, 1, (255, 0, 0), 1)

        cv2.imshow("Face Detection", img)
        cv2.imwrite('detected.jpg', img)
        cv2.waitKey(0) # wait for user input to display the detected face
        cv2.destroyAllWindows()

    else:
        return [], img

def main():
    #detect("qw2.jpg")
    detect("datatest/negatif/negatif (50).jpg")

if __name__ == "__main__":
    main()

```