

**RANCANG BANGUN APLIKASI ANDROID
DETEKSI PENGGUNAAN MASKER WAJAH
MENGUNAKAN TENSORFLOW MOBILENET
DI FAKULTAS TEKNOLOGI INDUSTRI UNISSULA**

LAPORAN TUGAS AKHIR

Laporan Ini Disusun Guna Memenuhi Salah Satu Syarat Untuk Menyelesaikan
Program Studi Teknik Informatika S1 Pada Fakultas Teknologi Industri
Universitas Islam Sultan Agung



Disusun Oleh:

Nama : Yusuf Arief Wicaksono

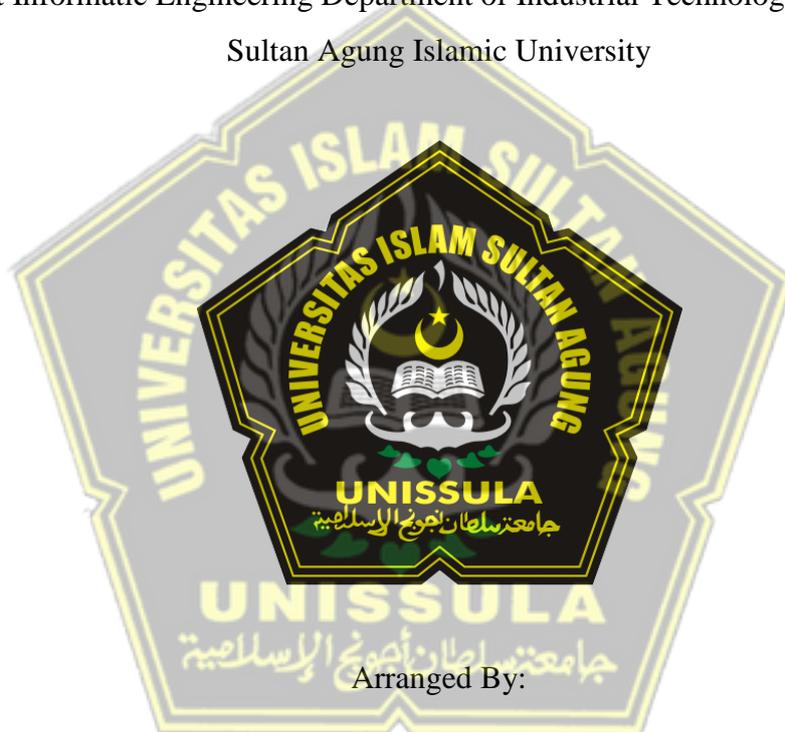
NIM : 32601700028

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG**

2022

FINAL PROJECT
SYSTEM DESIGN FOR
FACE MASK DETECTION ANDROID APPLICATION
WITH TENSORFLOW MOBILENET
AT FACULTY OF INDUSTRIAL TECHNOLOGY UNISSULA

Proposed to complete the requirement to obtain a bachelor's degree (S1)
At Informatic Engineering Department of Industrial Technology Faculty
Sultan Agung Islamic University



Nama : Yusuf Arief Wicaksono
NIM : 32601700028

MAJORING OF INFORMATIC ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG
2022

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul **“Rancang Bangun Aplikasi Android Deteksi Penggunaan Masker Wajah Menggunakan Tensorflow MobileNet Di Fakultas Teknologi Industri UNISSULA”** ini disusun oleh :

Nama : Yusuf Arief Wicaksono

NIM : 32601700028

Program Studi : Teknik Informatika

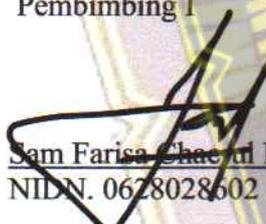
Telah disahkan oleh dosen pembimbing pada :

Hari :

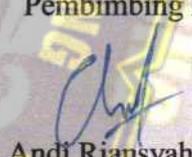
Tanggal :

Mengesahkan,

Pembimbing I


Sam Farisa Chaerul Haviana, ST, M.Kom
NIDN. 0628028602

Pembimbing II


Andi Riensyah, ST, M.Kom
NIDN. 0609108802

Mengetahui,

Ketua Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Islam Sultan Agung


Ir. Sri Mulyono, M.Eng
NIDN. 0626066601

LEMBAR PENGESAHAN PENGUJI

Laporan tugas akhir dengan judul **“Rancang Bangun Aplikasi Android Deteksi Penggunaan Masker Wajah Menggunakan Tensorflow MobileNet di Fakultas Teknologi Industri UNISSULA”** ini telah dipertahankan di depan dosen penguji Tugas Akhir pada :

Hari : Senin

Tanggal : 6 Februari 2023

TIM PENGUJI

Penguji I

Penguji II



Ir. Sri Mulyono, M.Eng
NIDN. 0626066601

09-02-2023


Ghufron, ST, M.Kom
NIDN. 0602079005



SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Yusuf Arief Wicaksono

NIM : 32601700028

Judul Tugas Akhir : Rancang Bangun Aplikasi Android Deteksi Penggunaan Masker Wajah Menggunakan Tensorflow Mobilenet Di Fakultas Teknologi Industri UNISSULA

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 30 Desember 2022

Yang Menyatakan,



Yusuf Arief Wicaksono

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Yusuf Arief Wicaksono

NIM : 32601700028

Program Studi : Teknik Informatika

Fakultas : Teknologi industri

Alamat Asal : Jln. R. Soeprpto no. 25 Kuwaron, Gubug, Grobogan

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : Rancang Bangun Aplikasi Android Deteksi Penggunaan Masker Wajah Menggunakan Tensorflow Mobilenet Di Fakultas Teknologi Industri UNISSULA Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialih mediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 30 Desember 2022

Yang menyatakan,



Yusuf Arief Wicaksono

KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul “Rancang Bangun Aplikasi Android Deteksi Penggunaan Masker Wajah Menggunakan Tensorflow MobileNet Di Fakultas Teknologi Industri UNISSULA” ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang. Tugas Akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, SH., M.Hum yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Novi Marlyana, ST., MT.
3. Dosen pembimbing I penulis Sam Farisa Chaerul Haviana, ST, M.Kom yang telah memberikan banyak nasehat dan saran.
4. Dosen pembimbing II penulis Andi Riansyah, ST, M.Kom yang telah meluangkan waktu dan memberi ilmu.
5. Orang tua penulis yang telah mengizinkan untuk menyelesaikan laporan ini.
6. Dan kepada semua pihak yang tidak dapat saya satu persatu.

Dengan segala kerendahan hati, penulis menyadari masih banyak terdapat banyak kekurangan-kekurangan dari segi kualitas atau kuantitas maupun dari ilmu pengetahuan dalam penyusunan laporan, sehingga penulis mengharapkan adanya saran dan kritikan yang bersifat membangun demi kesempurnaan laporan ini.

Semarang, Desember 2022

Yusuf Arief Wicaksono

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN PEMBIMBING	iii
LEMBAR PENGESAHAN PENGUJI	iv
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR	v
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xi
ABSTRAK.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Pembatasan Masalah.....	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori.....	6
2.2.1 Convolutional Neural Network.....	6
2.2.2 MobileNet	7
2.2.3 Depthwise Separable Convolutions	13
2.2.4 Tensorflow	15

BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM.....	16
3.1 Pengumpulan Data	16
3.2 Pengembangan Model Deteksi Masker Berbasis MobileNet.....	16
3.3 Perancangan Aplikasi Deteksi Masker.....	18
BAB IV HASIL DAN ANALISIS PENELITIAN	22
4.1 Implementasi Sistem	22
4.2 Pengujian Sistem.....	23
4.3 Analisa Hasil Pengujian	38
BAB V KESIMPULAN DAN SARAN	41
5.1 Kesimpulan	41
5.2 Saran.....	41
DAFTAR PUSTAKA	



DAFTAR GAMBAR

Gambar 2. 1 Perbandingan Kinerja Model pada COCO Challenge.....	5
Gambar 2. 2 <i>Filter</i> CNN	6
Gambar 2. 3 Parameter <i>stride</i> menentukan jarak pergerakan <i>filter</i>	7
Gambar 2. 4 Cara kerja CNN.....	7
Gambar 2. 5 Struktur MobileNet V1	8
Gambar 2. 6 <i>Deep Separable Convolution Layer</i> MobileNet V1	8
Gambar 2. 7 ReLU menyeleksi elemen penting gambar	9
Gambar 2. 8 Softmax mengubah <i>output</i> model menjadi <i>probability</i>	10
Gambar 2. 9 Struktur MobileNet V2	10
Gambar 2. 10 <i>Bottleneck Residual Convolution Layer</i> MobileNet V2.....	11
Gambar 2. 11 <i>Skip</i> menghubungkan bagian awal dan akhir <i>network</i>	11
Gambar 2. 12 Cara kerja <i>Bottleneck Residual Block</i>	12
Gambar 2. 13 Jumlah <i>Channel</i> Setiap Bagian <i>Bottleneck Residual Block</i>	12
Gambar 2. 14 Performa MobileNet V2 untuk <i>Object Detection</i> pada ImageNet	13
Gambar 2. 15 <i>Metrics</i> MobileNet V2 dibandingkan V1.....	13
Gambar 2. 16 Cara kerja <i>Depthwise Convolution</i>	14
Gambar 2. 17 Cara kerja <i>Pointwise Convolution</i>	14
Gambar 2. 18 Cara kerja TensorFlow	15
Gambar 3. 1 Langkah-langkah pembuatan dan <i>training</i> model.....	18
Gambar 3. 2 <i>Flowchart</i> sistem.....	19
Gambar 3. 3 Rancangan <i>UI</i> Aplikasi	20
Gambar 4. 1 Tampilan Aplikasi	22
Gambar 4. 5 Hasil Evaluasi <i>Precision vs Recall</i>	40

DAFTAR TABEL

Tabel 4. 1 Hasil pengujian sistem	24
Tabel 4. 2 Evaluasi pengujian sistem.....	39



Abstrak

CoronaVirus Disease 2019 (COVID-19) merupakan sebuah penyakit menular yang disebabkan oleh virus SARS-CoV-2 yang berpotensi menimbulkan kematian. Menanggapi penyebaran COVID-19 banyak kantor memilih untuk melaksanakan kegiatan bekerja secara online atau lebih dikenal dengan istilah *working from home (WFH)*. Setelah melaksanakan pembelajaran berbasis daring selama 21 bulan, Fakultas Teknologi Industri Universitas Islam Sultan Agung (FTI UNISSULA) kini tengah mempersiapkan pelaksanaan kegiatan kependidikan secara luring namun terbatas. Berkaitan dengan ini, Dekan FTI UNISSULA mengeluarkan Keputusan Dekan Fakultas Teknologi Industri UNISSULA Nomor 852/A/SA-TI/X/2021 tentang Panduan Pembelajaran Luar Jaringan (Luring) Terbatas Di Fakultas Teknologi Industri Universitas Islam Sultan Agung (UNISSULA) Tahun Akademik 2021-2022. Panduan ini berisi penjelasan mengenai Prosedur Protokol Kesehatan yang wajib ditaati oleh seluruh warga FTI. Salah satu langkah pencegahan COVID yang disarankan dalam Panduan tersebut yaitu penggunaan masker wajah pada saat melakukan kegiatan. Namun prosedur ini cenderung dilupakan dan seringkali tidak diindahkan. Untuk mengatasi persoalan tersebut melalui penelitian ini akan dibuat sebuah program aplikasi yang dapat secara otomatis mendeteksi penggunaan masker wajah pada civitas FTI UNISSULA dan memberikan peringatan apabila ada yang belum mengenakan masker.

Kata kunci: Deteksi, Masker, Aplikasi *Mobile*, *Smartphone*, *MobileNet*, *TensorFlow*.

Abstract

CoronaVirus Disease 2019 (COVID-19) is a potentially deadly contagious disease caused by SARS-CoV-2 virus. In response to the spread of COVID-19 many offices chose to do work digitally with the help of computer or better known as working from home (WFH). After 21 months of online classes, Faculty of Industrial Technology of Sultan Agung Islamic University (FTI UNISSULA) is now preparing to hold limited offline classes. Accordingly, Dean of FTI UNISSULA released Decree of Dean of FTI UNISSULA Number 852/A/SA-TI/X/2021 about Guidance for Limited Offline Classes at Faculty of Industrial Technology of Sultan Agung Islamic University (UNISSULA) Academic Year 2021-2022. This document details health protocols which are to be followed by everyone in FTI. One of said protocols concerns the importance of wearing a face mask during activities, but this protocol is often ignored. To help with the aforementioned issue this research will enable the creation of a mobile app able to detect face masks on citizens of FTI and will warn them if they have yet to wear a mask.

Key words: *Detection, Mask, Mobile Application, Smartphone, MobileNet, TensorFlow*

BAB I

PENDAHULUAN

1.1 Latar Belakang

CoronaVirus Disease 2019 (COVID-19) merupakan sebuah penyakit menular yang disebabkan oleh virus SARS-CoV-2. Gejala penyakit ini bermacam-macam, tetapi seringkali berupa demam, batuk, sakit kepala, kecapekan, kesulitan bernapas, dan hilangnya kemampuan mencium bau dan merasakan makanan. Penyakit ini berpotensi menimbulkan kematian. Menanggapi penyebaran COVID-19 banyak kantor dan fasilitas umum pada awal pandemi memilih untuk menghentikan sementara kegiatan aktifitas fisik di tempat, dengan sebagian kantor memilih untuk melaksanakan kegiatannya secara digital dengan memanfaatkan perangkat lunak komputer untuk bekerja secara *online* sehingga memungkinkan proses bisnis untuk dapat berjalan seperti biasa atau lebih dikenal dengan istilah *working from home* (WFH) (Mungkasa, 2020).

Setelah melaksanakan pembelajaran berbasis daring selama 21 bulan, Fakultas Teknologi Industri Universitas Islam Sultan Agung (FTI UNISSULA) berdasarkan Surat Edaran Rektor UNISSULA Nomor 9518/J/SA/X/2021 tentang Aktifitas Kerja Tenaga Kependidikan Secara *Work from Office* kini tengah mempersiapkan pelaksanaan kegiatan kependidikan secara luring namun terbatas. Berkaitan dengan ini, Dekan FTI UNISSULA mengeluarkan Keputusan Dekan Fakultas Teknologi Industri UNISSULA Nomor 852/A/SA-TI/X/2021 tentang Panduan Pembelajaran Luar Jaringan (Luring) Terbatas Di Fakultas Teknologi Industri Universitas Islam Sultan Agung (UNISSULA) Tahun Akademik 2021-2022. Panduan ini berisi penjelasan mengenai Prosedur Protokol Kesehatan yang wajib ditaati oleh seluruh staf karyawan, dosen, dan mahasiswa FTI demi memastikan terlaksananya proses belajar mengajar yang kondusif. Kebijakan ini merupakan salah satu upaya pencegahan untuk mengurangi resiko penyebaran COVID-19 di kalangan civitas kampus.

Salah satu langkah pencegahan COVID yang disarankan dalam Panduan tersebut yaitu penggunaan masker wajah pada saat melakukan kegiatan. Langkah

ini merupakan cara mudah untuk mengurangi resiko penyebaran virus melalui *droplet* air pada saat berbicara. Namun prosedur ini cenderung dilupakan dan seringkali tidak diindahkan sehingga berpotensi mengakibatkan hal yang tidak diinginkan. Untuk mengatasi persoalan tersebut melalui penelitian ini akan dibuat sebuah program aplikasi yang dapat secara otomatis mendeteksi penggunaan masker wajah pada civitas FTI UNISSULA dan memberikan peringatan apabila ada yang belum mengenakan masker. Aplikasi ini berjalan pada perangkat *smartphone* berbasis Android yang terletak di dekat pintu masuk setiap ruangan sehingga semua orang yang masuk dapat secara otomatis terdeteksi. Aplikasi ini diharapkan dapat membantu mengawal pelaksanaan protokol kesehatan di FTI dan mengurangi resiko penyebaran COVID-19.

1.2 Perumusan Masalah

Bagaimana memanfaatkan MobileNet untuk membuat sebuah aplikasi yang dapat mendeteksi penggunaan masker wajah secara visual pada dosen, karyawan, dan mahasiswa FTI UNISSULA dan memberikan peringatan apabila ada yang tidak mengenakan masker.

1.3 Pembatasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Aplikasi berfungsi mendeteksi penggunaan masker wajah pada dosen, karyawan, dan mahasiswa FTI UNISSULA.
2. Aplikasi menggunakan *library* TensorFlow dengan model MobileNet yang sudah dilatih sebelumnya untuk keperluan pendeteksian masker wajah.
3. Aplikasi dibuat menggunakan *framework* Flutter dan berjalan pada perangkat *smartphone* Android.
4. Pendeteksian masker dilakukan dari tampak depan.
5. Pendeteksian masker hanya untuk masker berwarna putih.
6. Penelitian ini tidak membahas segi keamanan.

1.4 Tujuan

Menciptakan sebuah prototipe aplikasi pendeteksi masker wajah yang dapat digunakan untuk membantu menerapkan prosedur pencegahan COVID-19 pada warga FTI UNISSULA dan memberikan peringatan kepada yang belum mengenakan masker.

1.5 Manfaat

Manfaat yang diharapkan dari pembuatan tugas akhir ini antara lain menghasilkan aplikasi yang dapat membantu mencegah penyebaran lebih lanjut COVID-19 melalui deteksi otomatis dan pemberian peringatan apabila staff dan atau mahasiswa tidak mengenakan masker di lingkungan FTI.

1.6 Sistematika Penulisan

Sistematika penulisan pembuatan laporan tugas akhir adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini merupakan penjelasan mengenai latar belakang tugas akhir, rumusan masalah, pembatasan masalah, tujuan pembuatan aplikasi, manfaat penelitian, dan sistematika penulisan laporan.

BAB II TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini berisi penjabaran teori ilmu dan metode yang digunakan dalam penelitian serta rangkuman singkat dari penelitian-penelitian dengan tema serupa yang sudah pernah dilakukan sebelumnya.

BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM

Bab ini mendeskripsikan proses-proses dan tahapan-tahapan yang dilakukan dalam penelitian, diantaranya yaitu pengembangan model deteksi masker berbasis MobileNet dan perancangan aplikasi mobile.

BAB IV HASIL DAN ANALISIS PENELITIAN

Bab ini memuat hasil pengujian model MobileNet dan prototipe aplikasi mobile serta evaluasi kinerja hasil penelitian di lapangan pada FTI UNISSULA.

BAB V KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dari penelitian ini serta saran yang dapat diimplementasikan untuk dapat memberikan manfaat yang lebih besar.



BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pemilihan model MobileNet untuk penggunaan pada aplikasi *face recognition* didasarkan pada paper MobileNet yang membandingkan kemampuan *object detection* MobileNet dengan beberapa model lain yaitu VGG dan Inception. Percobaan *object detection* dilakukan menggunakan dataset COCO Challenge dan ditampilkan sebagai skor *mean Average Precision* (mAP) yang merupakan sebuah metrik populer untuk menguji performa model dalam melakukan operasi *information retrieval* dan *object detection*. Hasil percobaan dapat dilihat pada tabel berikut yang menunjukkan bahwa MobileNet mampu mencapai hasil akurasi yang hampir sama dengan model lain dengan ukuran dan kompleksitas lebih kecil (Howard *et al.*, 2017).

Framework Resolution	Model	mAP	Billion Mult-Adds	Million Parameters
SSD 300	deeplab-VGG	21.1%	34.9	33.1
	Inception V2	22.0%	3.8	13.7
	MobileNet	19.3%	1.2	6.8
Faster-RCNN 300	VGG	22.9%	64.3	138.5
	Inception V2	15.4%	118.2	13.3
	MobileNet	16.4%	25.2	6.1
Faster-RCNN 600	VGG	25.7%	149.6	138.5
	Inception V2	21.9%	129.6	13.3
	MobileNet	19.8%	30.5	6.1

Gambar 2. 1 Perbandingan Kinerja Model pada COCO Challenge

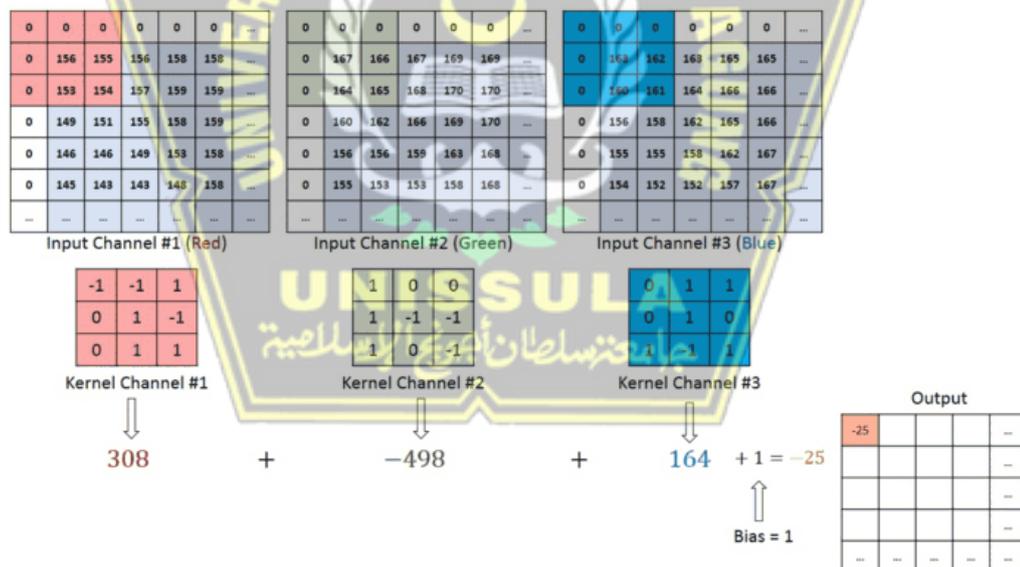
Penelitian deteksi masker berbasis Android sebelumnya sudah pernah dilakukan oleh Nyoman Purnama, et al dari STMIK PRIMAKARA yang menggunakan MobileNet dan TensorFlow untuk melakukan identifikasi jenis masker. Model dilatih menggunakan *dataset* kumpulan gambar masker yang diperoleh melalui pengambilan gambar sampel secara langsung menggunakan *smartphone* maupun citra yang diambil dari internet yang dikelompokkan menjadi 3 kategori yaitu masker bedah/medis, masker N95, dan masker kain/scuba. Sistem yang dihasilkan mampu mengidentifikasi secara realtime jenis masker yang digunakan dengan akurasi mencapai 90% pada jarak ideal 40 cm dengan kamera dan pencahayaan yang baik (Nyoman and Putu Kusuma Negara, 2021).

2.2 Dasar Teori

2.2.1 Convolutional Neural Network

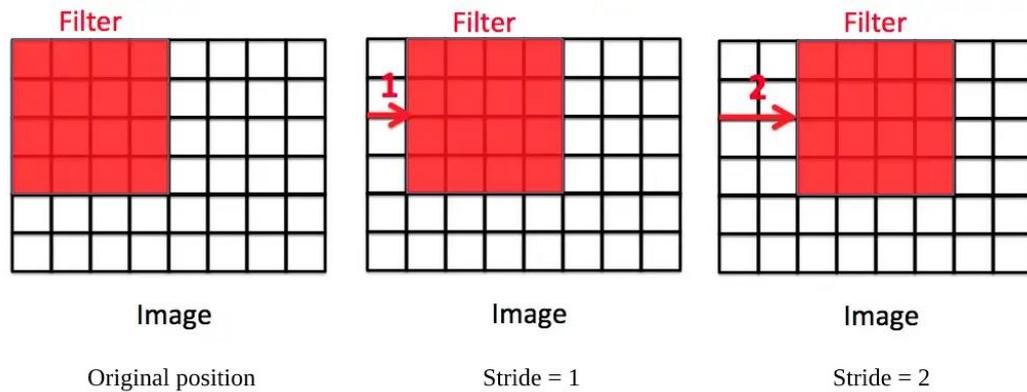
Convolutional Neural Network (ConvNet/CNN) adalah sebuah algoritma yang dapat memberikan *importance* (berupa weight dan bias) ke *features* dalam gambar sehingga dapat dibedakan satu objek dari yang lain (Sumit Saha, 2018). CNN bekerja menggunakan *convolution*, yaitu sebuah operasi matematika pada dua fungsi (f dan g) yang menghasilkan fungsi baru ($f * g$) yang merepresentasikan bagaimana bentuk satu fungsi dipengaruhi fungsi yang lain. CNN bertujuan mengubah gambar ke bentuk yang lebih mudah diproses komputer tanpa menghilangkan aspek-aspek penting yang dapat digunakan untuk melakukan identifikasi objek.

Sebuah *regular convolutional layer* menerapkan sebuah *convolution kernel* (“*filter*”) dengan ukuran tertentu pada semua *channels input image*, menghasilkan *weighted sum* untuk setiap *pixels* yang dilewati *kernel*.



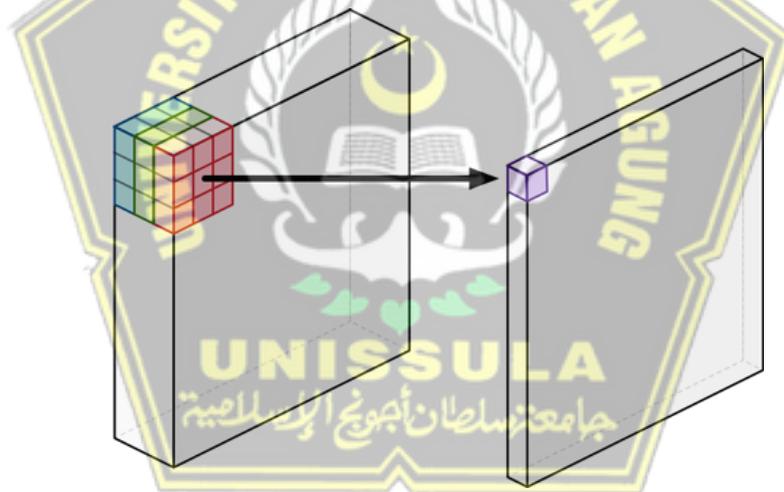
Gambar 2. 2 Filter CNN

Besar jarak pergerakan *filter* dapat diatur menggunakan parameter bernama *stride* (Ting-Hao Chen, 2017). *Stride* = 1 berarti *filter* akan bergerak satu *pixel* ke kanan. Parameter ini dapat digunakan untuk mengatur ukuran *output* supaya menjadi lebih kecil.



Gambar 2. 3 Parameter *stride* menentukan jarak pergerakan *filter*

Operasi *convolution* ini mengkombinasikan semua *values* dari *input channels*. Jika sebuah input gambar memiliki 3 *channel*, sebuah *convolution kernel* akan menghasilkan *output* yang hanya memiliki 1 *channel* per *pixel*.



Gambar 2. 4 Cara kerja CNN

2.2.2 MobileNet

MobileNet adalah sebuah model *Convolutional Neural Network* (CNN) yang didesain untuk dijalankan pada platform mobile dan embedded applications. Model ini bersifat *low-latency* dan *low-power* sehingga dapat digunakan dengan *resource* minimal untuk menangani berbagai macam skenario penggunaan. MobileNet dapat dilatih untuk menangani klasifikasi, deteksi, embeddings, dan lainnya. MobileNet dapat dijalankan di perangkat seluler menggunakan TensorFlow Lite.

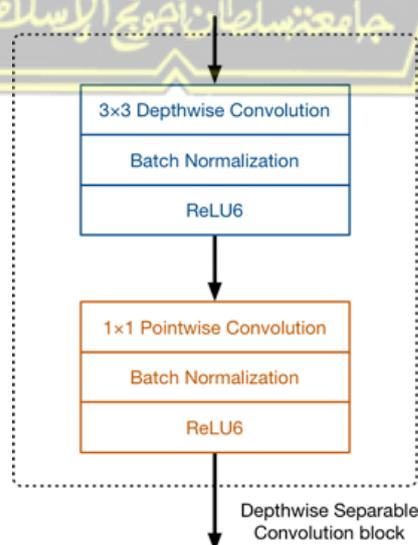
A. MobileNet V1

MobileNet V1 terdiri dari 30 *layers*. *Layer* pertama merupakan sebuah *regular convolution*, kemudian diikuti dengan *depthwise* dan *pointwise layers* secara bergantian. Beberapa *depthwise layer* memiliki parameter *stride=2* untuk memperkecil ukuran data yang diproses, dan *pointwise layer* yang mengikuti *layer* tersebut akan menggandakan jumlah *output channel*. Semua *convolutional layers* diikuti oleh ReLU *activation function*.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Gambar 2. 5 Struktur MobileNet V1



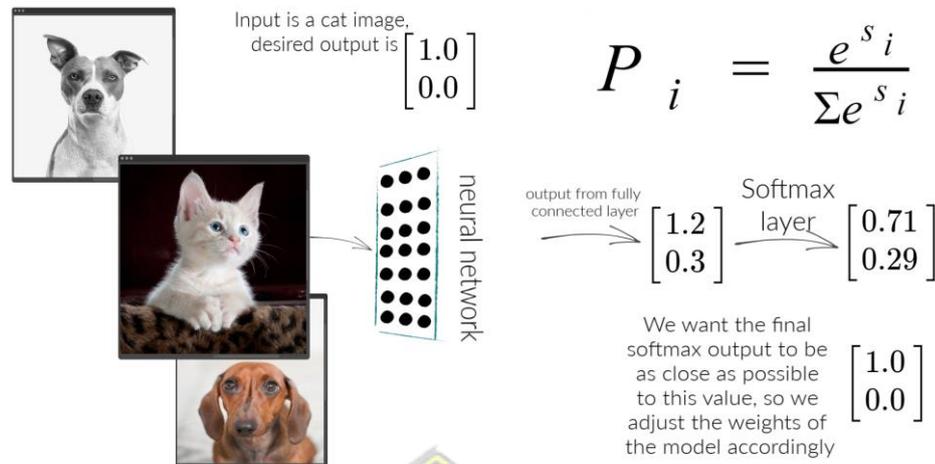
Gambar 2. 6 Deep Separable Convolution Layer MobileNet V1

Rectified linear activation unit (ReLU) function merupakan *activation function* yang berfungsi menentukan apakah sebuah elemen bersifat relevan dan perlu diproses lebih lanjut (SuperDataScience Team, 2018). Dalam kasus ini ReLU digunakan untuk menyeleksi apakah *output* dari *convolution* merupakan *pixel* yang mengandung informasi yang dapat digunakan untuk membantu proses klasifikasi dan perlu disimpan atau dapat diabaikan dan dihapus dari *matrix pixel*.



Gambar 2. 7 ReLU menyeleksi elemen penting gambar

Kombinasi ini berlangsung sampai image dengan ukuran 224×224 *pixels* berubah menjadi 7×7 *pixels* dengan 1024 *channels*. Kemudian dilanjutkan oleh sebuah *average-pooling layer* yang melakukan operasi pada semua *pixel*, menghasilkan sebuah *vector image* berukuran $1 \times 1 \times 1024$. Proses di atas menghasilkan *output* berupa vektor yang dapat digunakan untuk melakukan identifikasi apakah *object* dalam gambar merupakan *object* dari sebuah *class*. Untuk dapat menggunakan MobileNet sebagai *classifier* dan mendapatkan hasil klasifikasi sebagai probabilitas perlu ditambahkan sebuah *final layer* berupa *fully-connected layer* yaitu sebuah *softmax function*.



Gambar 2. 8 *Softmax* mengubah *output* model menjadi *probability*

Softmax function merupakan *activation function* yang berfungsi untuk mengubah *real values* menjadi *probabilities* (Bhardwaj, 2020). Hasil keluaran dari fungsi ini berupa 1000 *outputs* berisi probabilitas *input gambar* berisi sebuah objek dari kategori terkait. Melalui proses ini MobileNet dapat memberikan prediksi identitas kelas objek pada gambar.

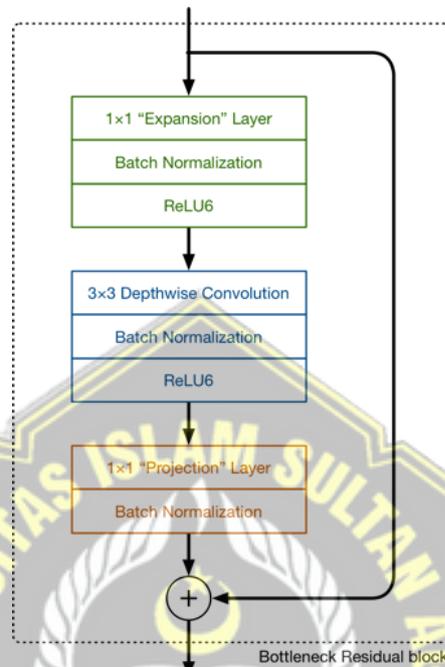
B. MobileNet V2

MobileNet V2 merupakan peningkatan dari MobileNet V1. MobileNet V2 memiliki arsitektur yang hampir sama dengan V1 dengan tambahan dua fitur baru: *Inverted Residuals* dan *Linear Bottlenecks* (Sandler *et al.*, 2018).

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

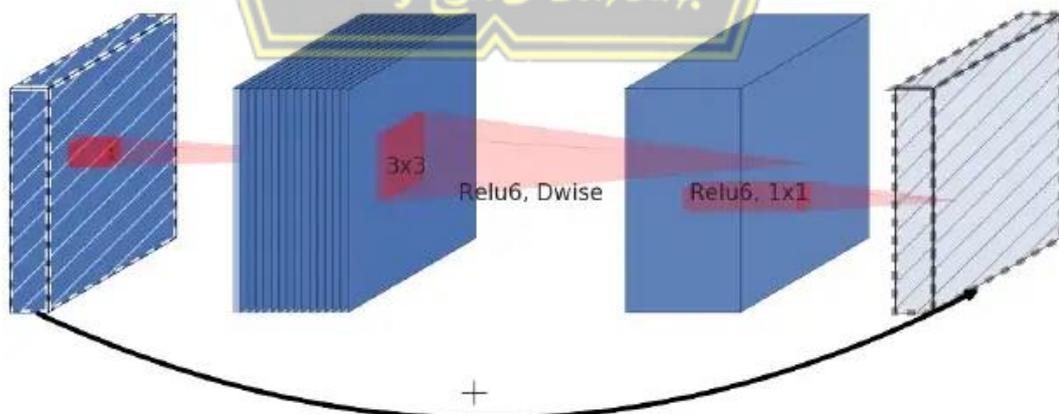
Gambar 2. 9 Struktur MobileNet V2

MobileNet V2 masih menggunakan teknik *depth separable convolution*, tetapi dalam bentuk *bottleneck residual block* yang berbeda dari *depth separable convolution* pada V1.



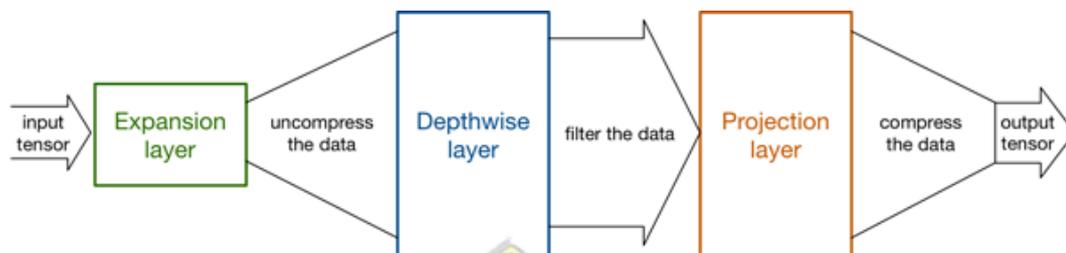
Gambar 2. 10 Bottleneck Residual Convolution Layer MobileNet V2

Residual blocks menghubungkan bagian awal dan akhir dari *convolution* melalui sebuah jalur terkoneksi yang disebut '*skip*'. Jalur ini memungkinkan network untuk mengakses data yang terdapat pada tingkat atas *layer* yang belum tersentuh *convolution* (Sandler *et al.*, 2018).



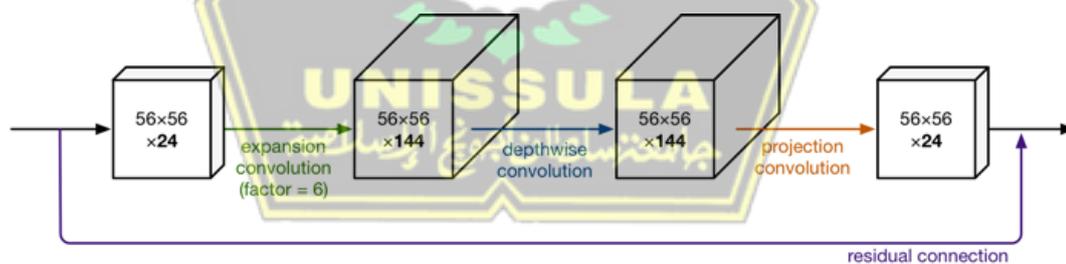
Gambar 2. 11 Skip menghubungkan bagian awal dan akhir *network*

Residual blocks biasa memiliki *channels* dengan struktur besar -> kecil -> besar. MobileNet V2 menggunakan struktur kecil -> besar -> kecil yang disebut *inverted residual blocks* karena merupakan kebalikan dari struktur normal (Paul-Louis Pröve, 2018).



Gambar 2. 12 Cara kerja *Bottleneck Residual Block*

Bottleneck Residual Block terdiri dari tiga *convolution layers*. *Layer* pertama merupakan sebuah 1×1 *convolution* yang memperbesar ukuran *input* dan disebut *Expansion Layer*. *Layer* ini diikuti oleh sebuah *depthwise convolution layer* yang sama seperti terdapat pada MobileNet V1. *Layer* terakhir merupakan sebuah 1×1 *pointwise convolution* yang disebut *Projection Layer*, tetapi berbeda dengan pada V1 dimana *layer* dapat menghasilkan *output* dengan jumlah *channel* sama atau lebih besar dari *input*, pada V2 *layer* ini menghasilkan *output* dengan jumlah *channel* identik dengan *input* awal (Matthijs Hollemans, 2018).



Gambar 2. 13 Jumlah *Channel* Setiap Bagian *Bottleneck Residual Block*

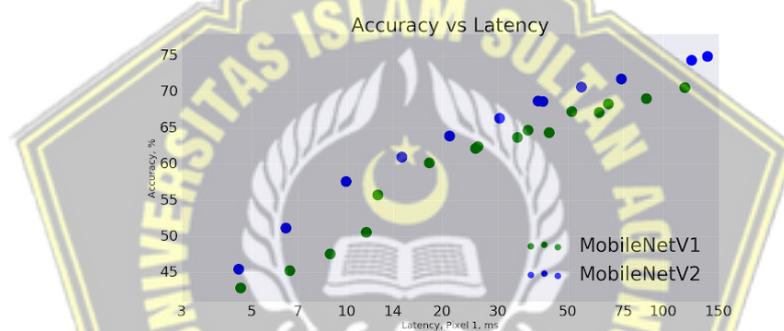
MobileNet V2 merupakan pengembangan dari desain V1 dengan menggunakan tensor berdimensi rendah untuk meminimalisir operasi yang harus dilakukan oleh *convolution layers* dan mengurangi beban komputasi yang diperlukan. Tetapi jika hanya menggunakan tensor berdimensi rendah *convolution layers* tidak akan dapat bekerja dengan baik dalam mengambil informasi yang diperlukan, untuk dapat melakukan pengambilan informasi dengan baik *convolution layer* memerlukan tensor dengan dimensi tinggi. MobileNet V2

dikembangkan dengan desain yang memperhatikan keseimbangan kedua aspek tersebut.

MobileNet V2 menunjukkan peningkatan performa yang signifikan dari V1. Model V2 dapat mencapai 30-40% lebih cepat dari V1 dengan menggunakan parameter 30% lebih kecil dan menghasilkan prediksi yang lebih akurat.

Network	Top 1	Params	MAdds	CPU
MobileNetV1	70.6	4.2M	575M	113ms
ShuffleNet (1.5)	71.5	3.4M	292M	-
ShuffleNet (x2)	73.7	5.4M	524M	-
NasNet-A	74.0	5.3M	564M	183ms
MobileNetV2	72.0	3.4M	300M	75ms
MobileNetV2 (1.4)	74.7	6.9M	585M	143ms

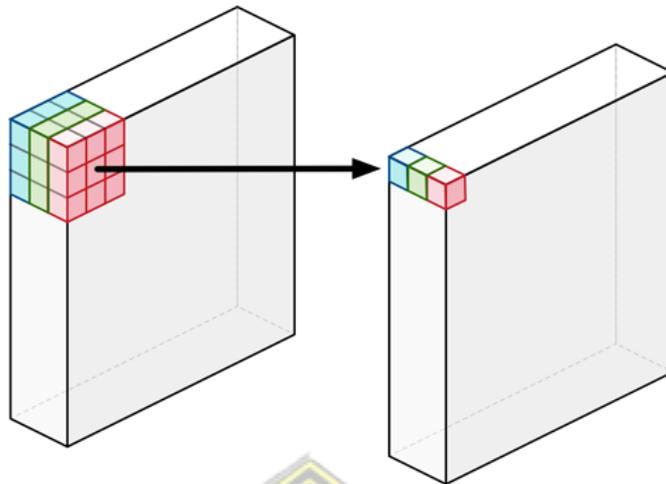
Gambar 2. 14 Performa MobileNet V2 untuk *Object Detection* pada ImageNet



Gambar 2. 15 Metrics MobileNet V2 dibandingkan V1

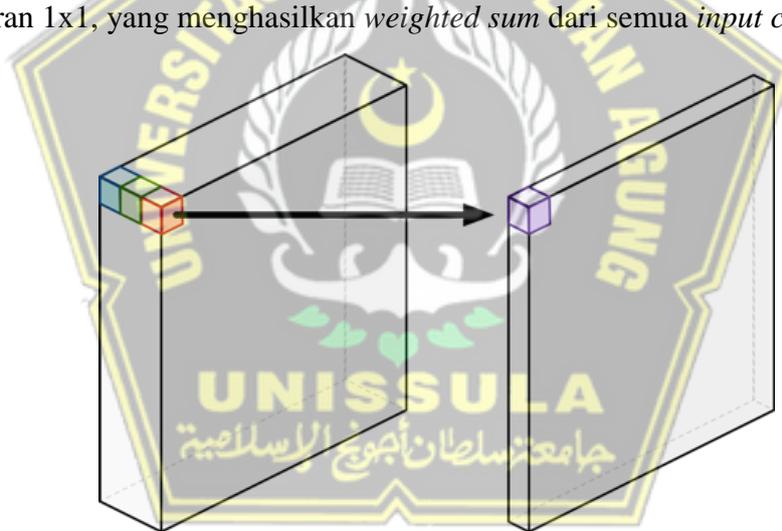
2.2.3 Depthwise Separable Convolutions

Depthwise Separable Convolution merupakan kombinasi dari 2 *convolutions*: *depthwise convolution* dan *pointwise convolution*. *Depthwise convolution* melakukan operasi *convolution* pada setiap *channel* secara terpisah. Jika sebuah *input* gambar memiliki 3 *channel*, *depthwise convolution* akan menghasilkan *output* yang juga memiliki 3 *channel*, dengan masing-masing *channel* memiliki weights nya sendiri.



Gambar 2. 16 Cara kerja *Depthwise Convolution*

Depthwise convolution diikuti oleh *pointwise convolution*, sebuah proses yang kurang lebih sama dengan *regular convolution* menggunakan *kernel* berukuran 1×1 , yang menghasilkan *weighted sum* dari semua *input channels*.

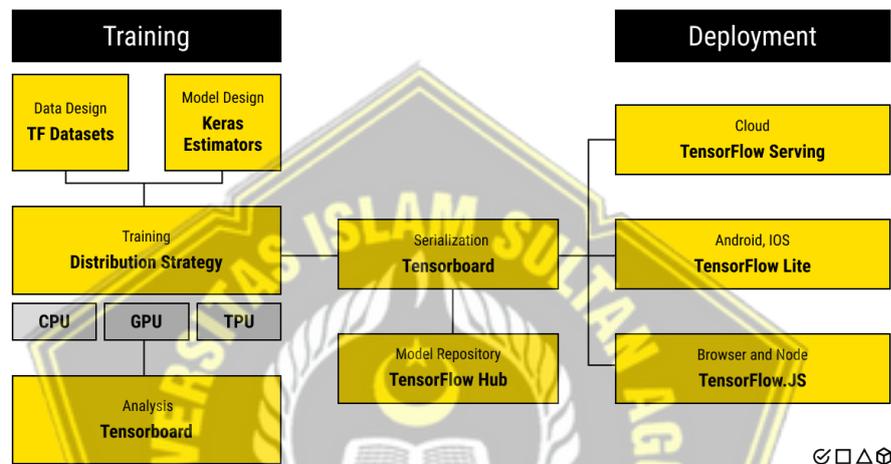


Gambar 2. 17 Cara kerja *Pointwise Convolution*

Regular convolution melakukan kedua proses ini sekaligus, sedangkan pada *depthwise separable convolution* kedua operasi ini dijalankan secara terpisah. Penggunaan *depthwise separable convolutions* mengurangi secara signifikan jumlah parameter yang dibutuhkan jika dibandingkan dengan *regular convolutions* pada *depth* yang sama (Matthijs Hollemans, 2017), menghasilkan sebuah *neural network* yang memerlukan komputasi lebih sedikit dan daya lebih kecil dengan hanya sedikit pengurangan pada akurasi sehingga dapat dijalankan pada platform mobile / embedded.

2.2.4 Tensorflow

TensorFlow merupakan sebuah platform *open source end-to-end* untuk *machine learning* yang memiliki ekosistem alat, library, dan sumber daya komunitas yang komprehensif dan fleksibel yang memungkinkan *developer* untuk membangun dan menerapkan aplikasi yang didukung ML dengan mudah. Tensorflow bekerja menggunakan Tensor. Tensor adalah sebuah n-dimensional *vector matrix* yang merepresentasikan *input data*.



Gambar 2. 18 Cara kerja TensorFlow

TensorFlow dikembangkan untuk dapat bekerja secara umum dan tidak memiliki bidang fokus spesifik tertentu. Hampir semua jenis model dapat dibuat, tetapi penggunaan paling umum TensorFlow adalah untuk membuat *neural networks*.

TensorFlow dapat menghasilkan berbagai jenis model untuk beragam keperluan, seperti:

1. *Image classification (e.g., handwritten digit classification)*
2. *Image recognition (e.g., facial recognition, image search)*
3. *Recurrent Neural Network (e.g., LSTM models untuk time series forecasting)*
4. *Natural Language Processing (e.g., seq2seq models untuk machine translation)*

BAB III

METODE PENELITIAN DAN PERANCANGAN SISTEM

3.1 Pengumpulan Data

Tahapan pengumpulan data yang dilakukan dalam penelitian ini adalah :

1. Studi Literatur

Studi literatur dilakukan untuk mempelajari berbagai sumber referensi seperti berbagai buku, jurnal, website, maupun YouTube yang berkaitan dengan penelitian demi memperdalam pengetahuan mengenai subjek yang diambil.

2. Rancang Bangun dan Pengembangan Aplikasi Mobile

Membangun aplikasi mobile yang dapat menggunakan kamera untuk menangkap gambar wajah pelanggan secara *real-time* untuk kemudian diproses.

3. Pengembangan Subsistem Pendeteksi Masker Wajah

Membangun subsistem aplikasi yang dapat menggunakan data gambar wajah dari kamera untuk menentukan apakah seseorang sedang mengenakan masker atau tidak secara *real-time* berbasis TensorFlow Lite.

4. Pengujian

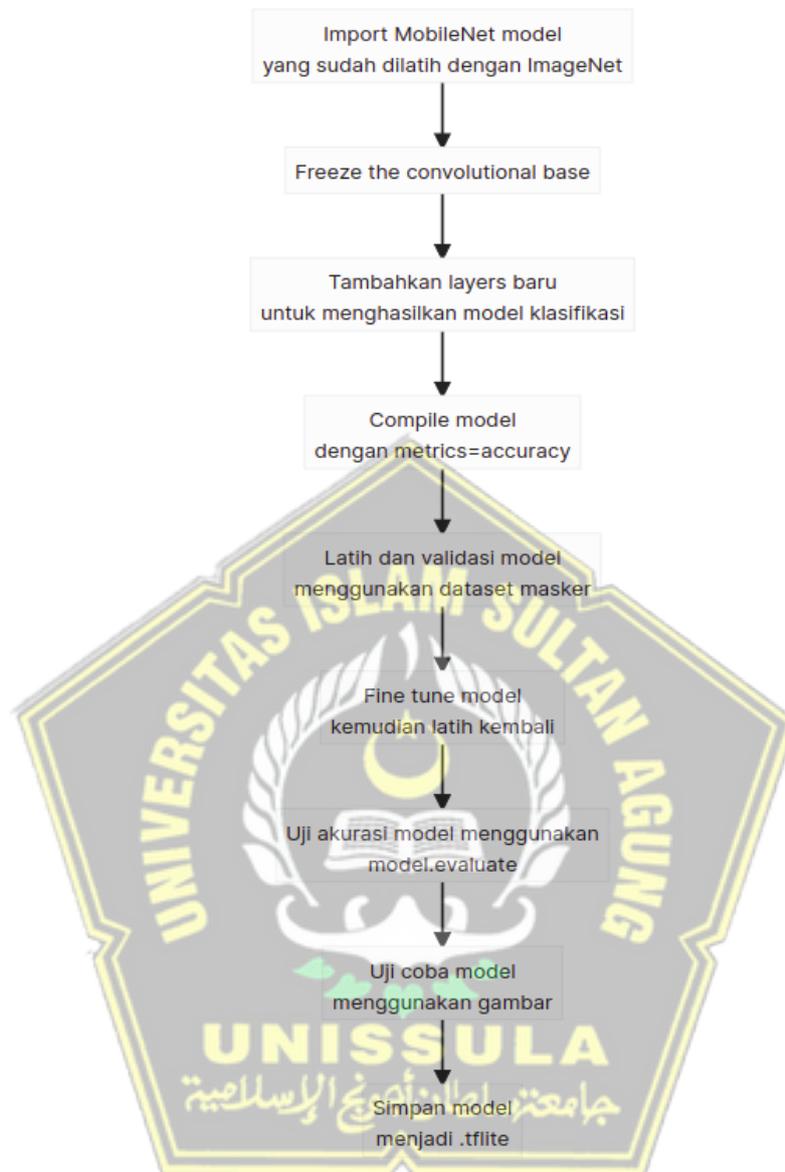
Pengujian dilakukan menggunakan metode *Black Box testing*. *Black Box Testing* adalah metode pengujian dimana sebuah program dievaluasi hanya dengan melihat kinerjanya dari luar saja tanpa mengetahui proses dibalik layar bagaimana program tersebut mendapatkan hasil yang ditampilkan dan tanpa melihat sumber kode program. Metode ini dipilih karena kemudahannya.

3.2 Pengembangan Model Deteksi Masker Berbasis MobileNet

Pemilihan model MobileNet didasarkan pada *paper* MobileNet yang menunjukkan bahwa MobileNet mampu mencapai hasil akurasi yang hampir sama dengan model lain dengan ukuran dan kompleksitas lebih kecil. Pembuatan model dilakukan dengan bantuan Google Colab menggunakan metode *transfer learning* yaitu melatih ulang model yang sudah terlatih menggunakan *dataset* baru. Model dibuat menggunakan *library* Keras.

Langkah-langkah yang dilakukan dalam proses *training* model deteksi masker adalah sebagai berikut:

1. Langkah pertama yaitu *import* model MobileNet yang sudah di-*training* menggunakan ImageNet sebagai dasar model baru. Kemudian bekukan *convolution base layers* model sehingga kemampuan *image recognition* hasil *training* sebelumnya tidak berubah. Tambahkan *layers* baru yang berfungsi mengubah *output recognition* MobileNet ke dalam bentuk *array* yang berisi *probability* sebuah *object* merupakan *item* dari sebuah *class*. Kemudian lakukan kompilasi model menggunakan *metrics=accuracy*.
2. Setelah model baru selesai dibuat, lakukan *training* dengan *dataset* yang diinginkan yaitu *dataset* masker dan validasi hasil *training*. Jika hasil validasi dirasa belum memuaskan, lakukan *fine tuning* lebih lanjut pada model untuk meningkatkan akurasi kemudian ulangi proses *training* dan validasi hasil *training*. Hasil akhir dari tahap ini adalah model yang dapat melakukan prediksi *object* pada gambar *input*.
3. Setelah proses *training* model kemudian lakukan *testing* untuk mengetahui performa model. Pertama-tama lakukan *testing* menggunakan fungsi *model.evaluate* untuk mengetahui akurasi model dalam memprediksi data baru yang belum pernah dilihat. Hasil yang diharapkan adalah skor akurasi yang hampir sama dengan skor akurasi model pada saat memprediksi data menggunakan data *training*. Kemudian lakukan *testing* menggunakan fungsi *model.predict* pada gambar. Hasil dari proses ini berupa *probability* penggunaan masker pada setiap gambar yang diharapkan sesuai dengan kenyataan.
4. Setelah semua proses selesai, simpan model menggunakan *format* tflite sehingga dapat digunakan pada TensorFlow mobile.



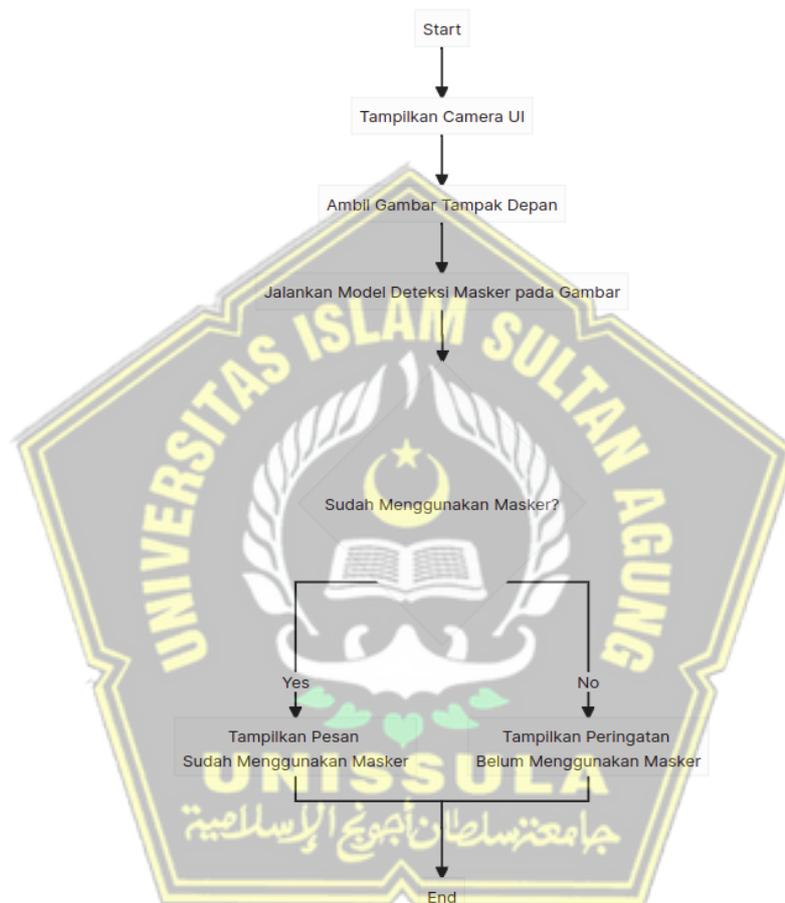
Gambar 3. 1 Langkah-langkah pembuatan dan *training* model

3.3 Perancangan Aplikasi Deteksi Masker

Penelitian ini bertujuan membuat sebuah program aplikasi yang secara otomatis dapat mendeteksi penggunaan masker wajah pada dosen, staf karyawan, dan mahasiswa FTI UNISSULA dan memberikan peringatan kepada yang belum mengenakan masker. Aplikasi yang dibuat merupakan aplikasi berbasis Android dan dalam pembuatannya menggunakan *framework* Flutter. Aplikasi ini berjalan pada perangkat *smartphone* yang terletak di dekat pintu masuk setiap ruangan

sehingga semua orang yang masuk dapat secara otomatis terdeteksi. Aplikasi menggunakan kamera untuk mengambil gambar wajah kemudian menjalankan model MobileNet yang sudah dibuat untuk menganalisa apakah seseorang menggunakan masker kemudian menampilkan hasilnya di layar.

Berikut merupakan *flowchart* cara kerja aplikasi deteksi masker wajah:



Gambar 3. 2 *Flowchart* sistem

Pada gambar 3.2 merupakan *flowchart* dari sistem yang dimulai ketika masuk ke aplikasi kemudian membuka kamera depan untuk mengambil gambar dari wajah, kemudian sistem melakukan proses deteksi apakah sedang menggunakan masker atau tidak. Hasil analisa akan ditampilkan langsung setelah gambar wajah masuk di kamera.

Berikut merupakan rancangan *user interface* aplikasi deteksi masker wajah:



Gambar 3. 3 Rancangan *UI* Aplikasi

Pada gambar 3.2 merupakan tampilan rancangan *UI* aplikasi yang berisi gambar wajah yang tertangkap kamera dan hasil dari proses deteksi berupa *result*, *score*, *pre processing time*, *inference time*, dan *total prediction time*. *Result* menampilkan kesimpulan hasil deteksi model. *Score* berisi hasil prediksi model kemungkinan pengguna sedang menggunakan masker, jika skor mendekati 1 maka kemungkinan pengguna menggunakan masker dan sebaliknya jika skor semakin mendekati 0 maka kemungkinan pengguna tidak menggunakan masker. *Pre processing time* menampilkan waktu yang dibutuhkan untuk menyesuaikan gambar agar bisa diterima model. *Inference time* merupakan waktu yang dibutuhkan model untuk menghitung probabilitas penggunaan masker. *Total prediction time*

merupakan keseluruhan waktu proses untuk melakukan deteksi penggunaan masker.

Aplikasi didesain untuk dijalankan pada perangkat dengan spesifikasi minimum sebagai berikut:

- Android 7.1.2 Nougat
- Qualcomm Snapdragon 435
- 2GB RAM
- Front Camera



BAB IV HASIL DAN ANALISIS PENELITIAN

4.1 Implementasi Sistem

Berikut merupakan tampilan hasil implementasi *user interface* aplikasi:



Gambar 4. 1 Tampilan Aplikasi

Pada gambar 4.1 merupakan tampilan dari aplikasi sudah yang dibuat sesuai dengan rancangan *UI*. Aplikasi menampilkan gambar wajah yang tertangkap kamera beserta hasil dari proses deteksi berupa *result*, *score*, *inference time*, *pre processing time*, dan *total prediction time*. *Result* berisi hasil deteksi penggunaan masker. *Score* berisi tingkat kemungkinan menggunakan masker, semakin mendekati 1 maka kemungkinan menggunakan masker semakin tinggi dan semakin mendekati 0 maka kemungkinan tidak menggunakan masker. *Pre-processing time*

merupakan waktu yang digunakan untuk melakukan *crop* gambar dan proses penyesuaian gambar agar bisa dilakukan proses selanjutnya. *Inference time* merupakan waktu yang dibutuhkan *machine learning* untuk memproses gambar setelah dilakukan *preprocessing*. *Total Prediction time* merupakan waktu keseluruhan dari sistem untuk proses mendeteksi penggunaan masker. Setelah model bekerja dan menghasilkan prediksi akan ditampilkan layar berisi hasil prediksi dan peringatan apabila pengguna belum mengenakan masker.

4.2 Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode *black box testing* yaitu pengujian tanpa melihat cara kerja internal aplikasi dengan hanya memperhatikan *input* dan *output* sehingga dapat memberikan gambaran kinerja aplikasi di lapangan. Pengujian dilakukan sesuai cara kerja sistem yaitu dengan menganalisa hasil tangkapan tampak depan wajah dan prediksi yang dihasilkan model.

Hasil prediksi sistem memiliki 4 kemungkinan keluaran (Sumeet Kumar Agrawal, 2021) yaitu sebagai berikut:

1. *True Positive*, model mendeteksi masker pada pengguna dengan masker (*system: with_mask, actual: with_mask*)
2. *False Positive*, model mendeteksi masker pada pengguna tanpa masker (*system: with_mask, actual: without_mask*)
3. *True Negative*, model tidak mendeteksi masker pada pengguna tanpa masker (*system: without_mask, actual: without_mask*)
4. *False Negative*, model tidak mendeteksi masker pada pengguna dengan masker (*system: without_mask, actual: with_mask*)

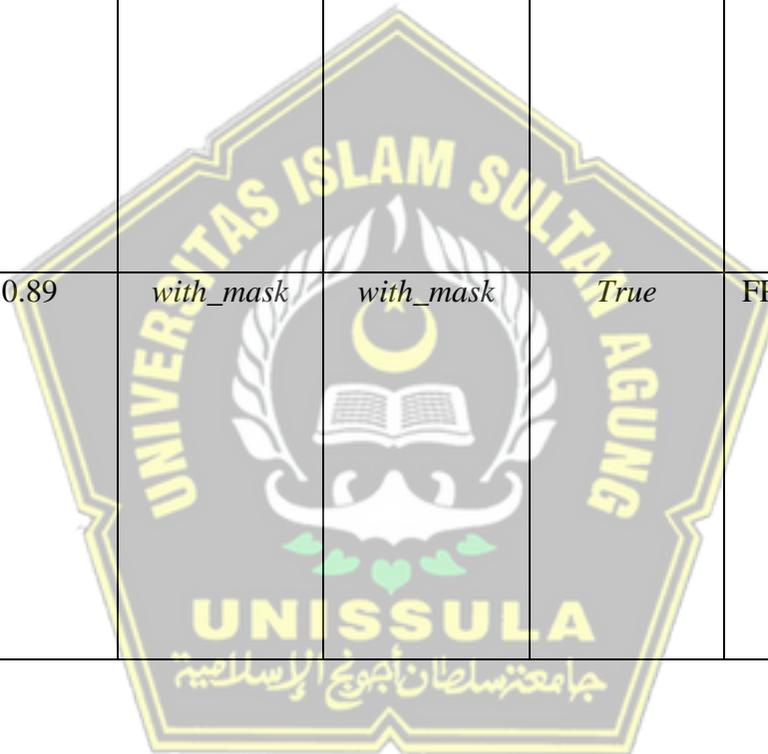
Pengujian model dilakukan pada ruangan dengan pencahayaan yang cukup dan menggunakan beberapa *Confidence Threshold* yang menentukan apakah sebuah keluaran *Positive* dianggap sebagai *True Positive* jika *confidence* lebih tinggi dari *threshold* atau *False Positive* jika *confidence* lebih rendah dari *threshold*.

Berikut merupakan tabel hasil pengujian model pada perangkat Android:

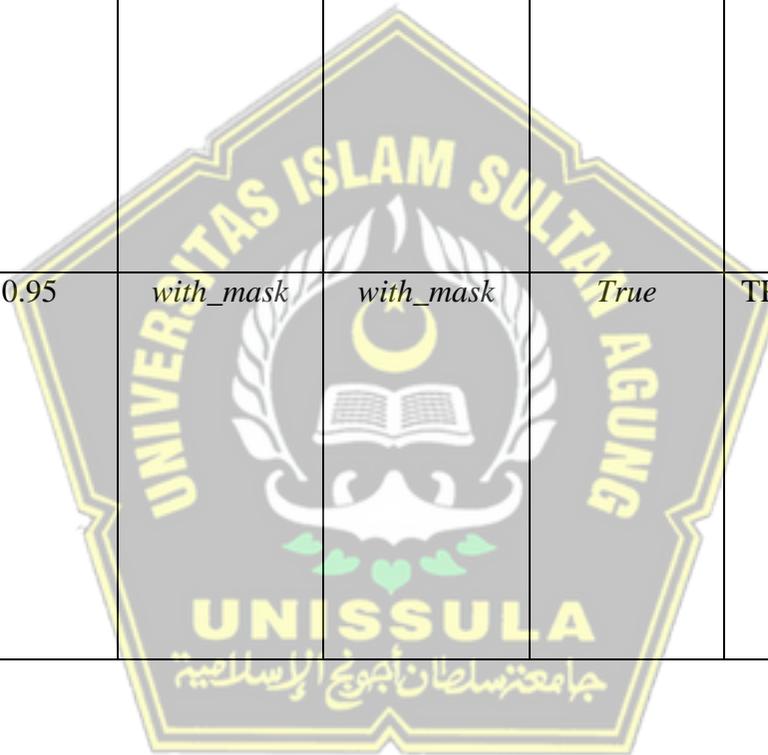
Tabel 4. 1 Hasil pengujian sistem

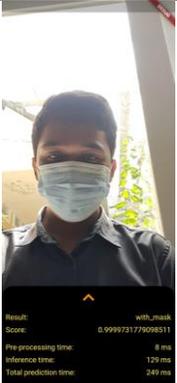
No	Hasil Tangkapan	Confidence Score	System	Actual	Is Prediction Correct	Threshold				
						0.9	0.8	0.7	0.6	0.5
1		0.99	<i>with_mask</i>	<i>with_mask</i>	<i>True</i>	TP	TP	TP	TP	TP
2		0.99	<i>with_mask</i>	<i>with_mask</i>	<i>True</i>	TP	TP	TP	TP	TP

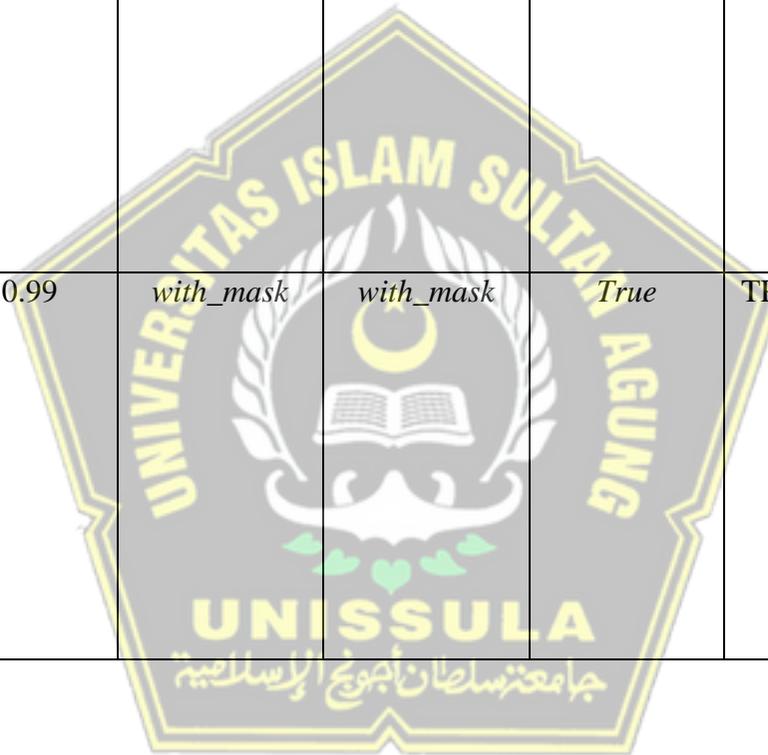
3		0.96	<i>with_mask</i>	<i>with_mask</i>	<i>True</i>	TP	TP	TP	TP	TP
4		0.89	<i>with_mask</i>	<i>with_mask</i>	<i>True</i>	FP	TP	TP	TP	TP



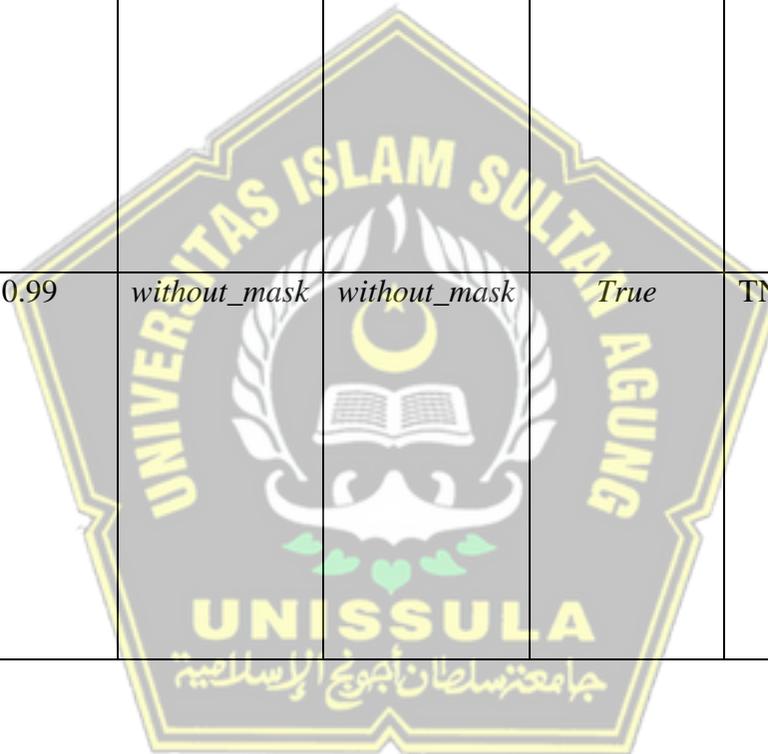
5		0.64	<i>with_mask</i>	<i>with_mask</i>	<i>True</i>	FP	FP	FP	TP	TP
6		0.95	<i>with_mask</i>	<i>with_mask</i>	<i>True</i>	TP	TP	TP	TP	TP



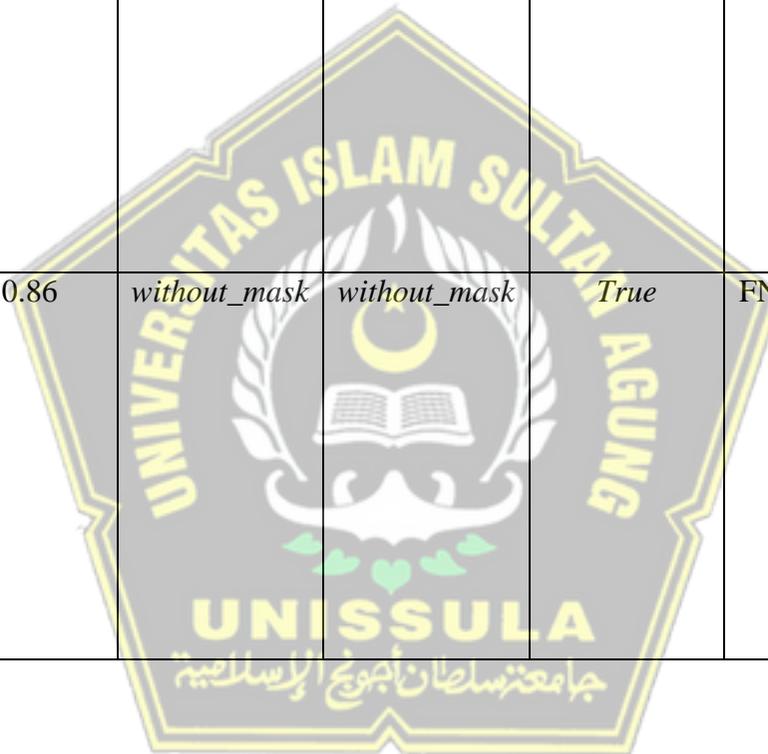
7	 <p>Result: without_mask Score: 0.9657276916502906 Pre-processing time: 10 ms Inference time: 389 ms Total prediction time: 637 ms</p>	0.96	<i>without_mask</i>	<i>with_mask</i>	<i>False</i>	FN	FN	FN	FN	FN
8	 <p>Result: with_mask Score: 0.9999731779098311 Pre-processing time: 8 ms Inference time: 129 ms Total prediction time: 249 ms</p>	0.99	<i>with_mask</i>	<i>with_mask</i>	<i>True</i>	TP	TP	TP	TP	TP



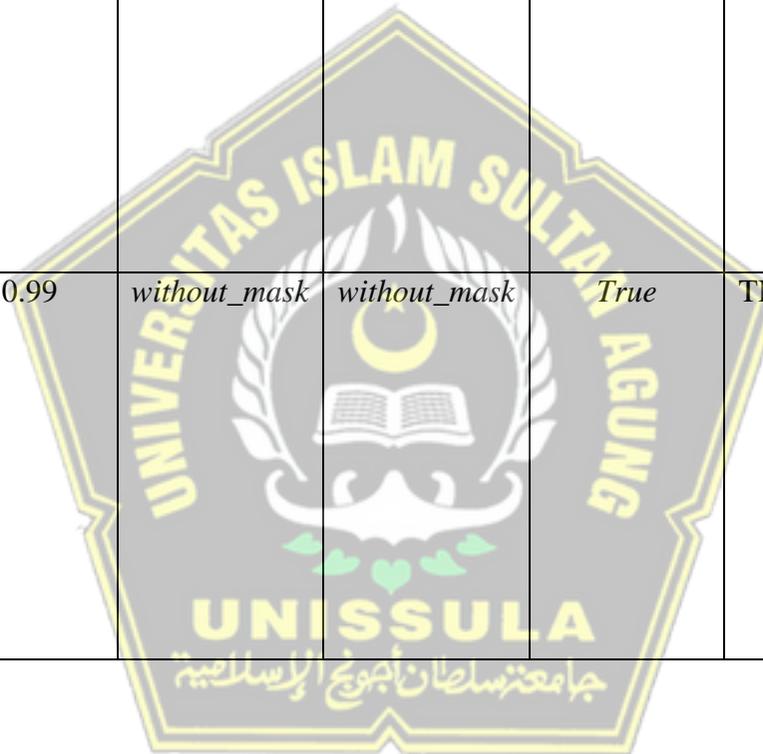
9	 <pre> Result: without_mask Score: 0.964253170013428 Pre-processing time: 29 ms Inference time: 258 ms Total prediction time: 440 ms </pre>	0.96	<i>without_mask</i>	<i>without_mask</i>	<i>True</i>	TN	TN	TN	TN	TN
10	 <pre> Result: without_mask Score: 0.9912509918212891 Pre-processing time: 8 ms Inference time: 119 ms Total prediction time: 174 ms </pre>	0.99	<i>without_mask</i>	<i>without_mask</i>	<i>True</i>	TN	TN	TN	TN	TN

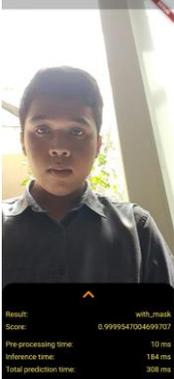


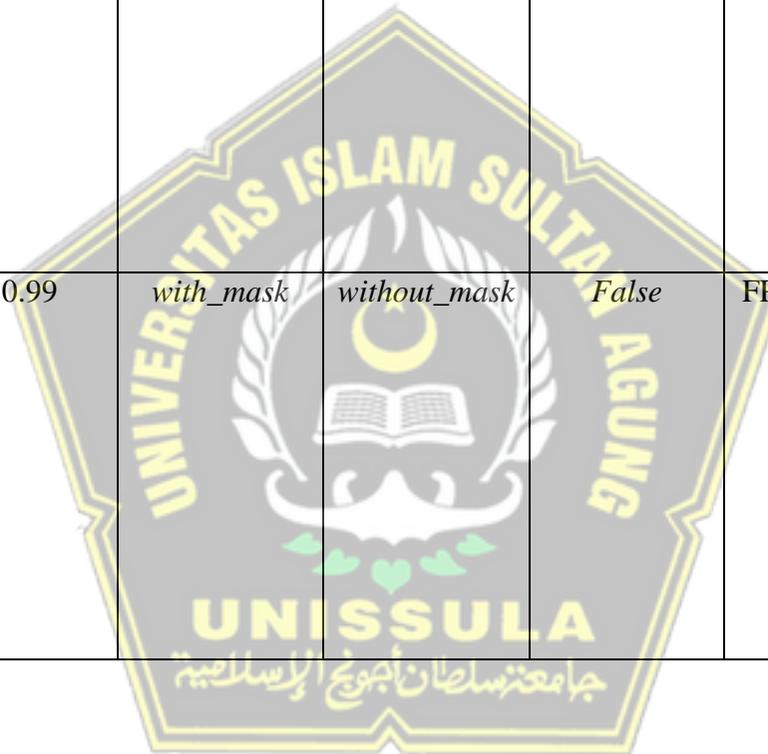
11	  <p>Result: without_mask Score: 0.9776947264289856 Pre-processing time: 8 ms Inference time: 97 ms Total prediction time: 153 ms</p>	0.97	<i>without_mask</i>	<i>without_mask</i>	<i>True</i>	TN	TN	TN	TN	TN
12	  <p>Result: without_mask Score: 0.8651860952377319 Pre-processing time: 8 ms Inference time: 155 ms Total prediction time: 209 ms</p>	0.86	<i>without_mask</i>	<i>without_mask</i>	<i>True</i>	FN	TN	TN	TN	TN



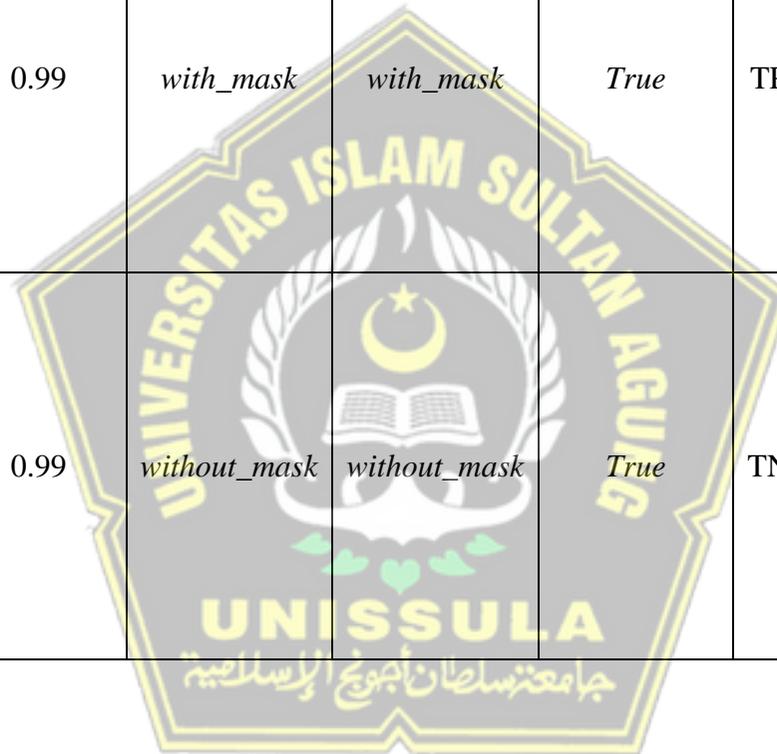
13	 <pre> Result: without_mask Score: 0.9849675046245056 Pre-processing time: 8 ms Inference time: 97 ms Total prediction time: 151 ms </pre>	0.98	<i>without_mask</i>	<i>without_mask</i>	<i>True</i>	TN	TN	TN	TN	TN
14	 <pre> Result: without_mask Score: 0.999922755432129 Pre-processing time: 8 ms Inference time: 95 ms Total prediction time: 153 ms </pre>	0.99	<i>without_mask</i>	<i>without_mask</i>	<i>True</i>	TN	TN	TN	TN	TN



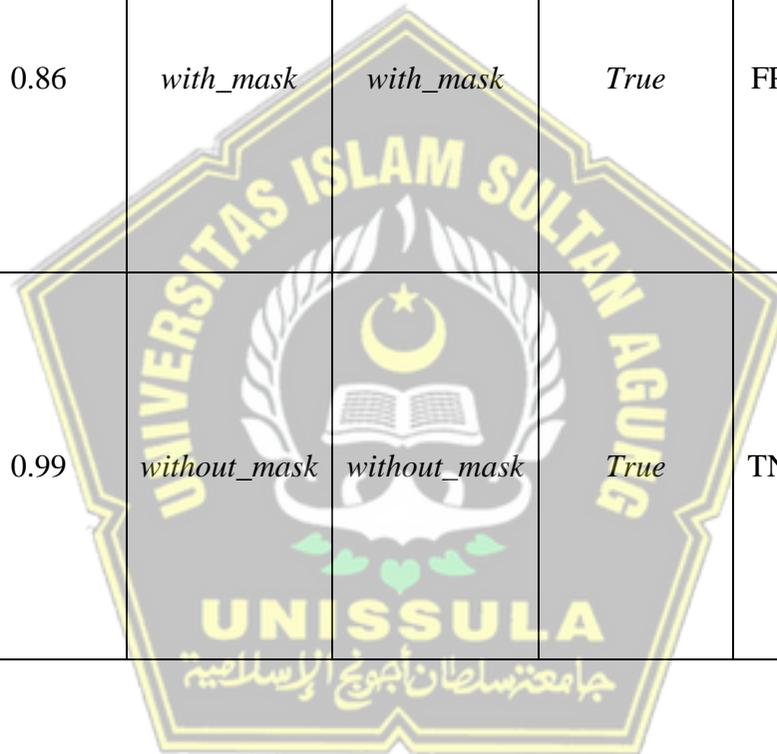
15		0.98	<i>without_mask</i>	<i>without_mask</i>	<i>True</i>	TN	TN	TN	TN	TN
16		0.99	<i>with_mask</i>	<i>without_mask</i>	<i>False</i>	FP	FP	FP	FP	FP



17		0.99	<i>with_mask</i>	<i>with_mask</i>	<i>True</i>	TP	TP	TP	TP	TP
18		0.99	<i>without_mask</i>	<i>without_mask</i>	<i>True</i>	TN	TN	TN	TN	TN



19	 <pre data-bbox="459 592 633 687"> Result: with_mask Score: 0.86062367635498 Pre-processing time: 9 ms Inference time: 92 ms Total prediction time: 153 ms </pre>	0.86	<i>with_mask</i>	<i>with_mask</i>	<i>True</i>	FP	TP	TP	TP	TP
20	 <pre data-bbox="459 975 633 1072"> Result: without_mask Score: 0.99271446464592 Pre-processing time: 8 ms Inference time: 126 ms Total prediction time: 182 ms </pre>	0.99	<i>without_mask</i>	<i>without_mask</i>	<i>True</i>	TN	TN	TN	TN	TN



21		0.98	<i>with_mask</i>	<i>with_mask</i>	<i>True</i>	TP	TP	TP	TP	TP
22		0.63	<i>without_mask</i>	<i>without_mask</i>	<i>True</i>	FN	FN	FN	TN	TN

23	 <pre data-bbox="461 592 633 687"> Result: with_mask Score: 0.9475193619728989 Pre-processing time: 8 ms Inference time: 97 ms Total prediction time: 164 ms </pre>	0.94	<i>with_mask</i>	<i>with_mask</i>	<i>True</i>	TP	TP	TP	TP	TP
24	 <pre data-bbox="461 975 633 1074"> Result: without_mask Score: 0.9418989208770735 Pre-processing time: 8 ms Inference time: 124 ms Total prediction time: 179 ms </pre>	0.94	<i>without_mask</i>	<i>without_mask</i>	<i>True</i>	TN	TN	TN	TN	TN

25	 <pre data-bbox="459 582 633 687"> Result: with_mask Score: 0.9999964237213135 Pre-processing time: 8 ms Inference time: 51 ms Total prediction time: 108 ms </pre>	0.99	<i>with_mask</i>	<i>with_mask</i>	<i>True</i>	TP	TP	TP	TP	TP
26	 <pre data-bbox="459 965 633 1074"> Result: with_mask Score: 0.9290328025817871 Pre-processing time: 8 ms Inference time: 66 ms Total prediction time: 151 ms </pre>	0.92	<i>with_mask</i>	<i>without_mask</i>	<i>False</i>	FP	FP	FP	FP	FP

27	 <pre data-bbox="461 582 631 687"> Result: with_mask Score: 0.98512380599756 Pre-processing time: 8 ms Inference time: 77 ms Total prediction time: 166 ms </pre>	0.98	<i>with_mask</i>	<i>with_mask</i>	<i>True</i>	TP	TP	TP	TP	TP
28	 <pre data-bbox="461 965 631 1074"> Result: without_mask Score: 0.9851962327957153 Pre-processing time: 7 ms Inference time: 125 ms Total prediction time: 178 ms </pre>	0.98	<i>without_mask</i>	<i>without_mask</i>	<i>True</i>	TN	TN	TN	TN	TN

4.3 Analisa Hasil Pengujian

Dalam melakukan evaluasi model *machine learning* terdapat beberapa *metrics* yang dapat digunakan untuk menilai performa dan kebenaran model dalam melaksanakan fungsinya.

Untuk model berupa *classifier* seperti model deteksi masker, *metrics* yang digunakan untuk melakukan evaluasi (Jonathan Grandperrin, 2021) yaitu:

1. *Precision*, yaitu *metric* yang menggambarkan berapa persen prediksi *Positive* yang merupakan *True Positive*. *Precision* berguna untuk kasus dimana *False Positive* merupakan hasil yang tidak diinginkan.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

2. *Recall*, yaitu *metric* yang menggambarkan berapa persen kasus *Positive* yang berhasil diprediksi. *Recall* berguna untuk kasus dimana *False Negative* merupakan hasil yang tidak diinginkan.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

3. *Accuracy*, yaitu *metric* yang menggambarkan berapa persen kemungkinan model untuk menghasilkan prediksi benar.

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

Evaluasi akan menampilkan *metrics* yang beragam tergantung pada *confidence threshold*. Untuk mendapatkan *metrics* yang akurat dilakukan beberapa kali pengujian pada *threshold* yang berbeda:

1. Pengujian pada *confidence threshold* 0.9 menghasilkan keluaran 10 *True Positive* dan 10 *True Negative*, dengan nilai *Precision* 67%, *Recall* 77%, dan *Accuracy* 71%.
2. Pengujian pada *confidence threshold* 0.8 menghasilkan keluaran 12 *True Positive* dan 11 *True Negative*, dengan nilai *Precision* 80%, *Recall* 86%, dan *Accuracy* 82%.

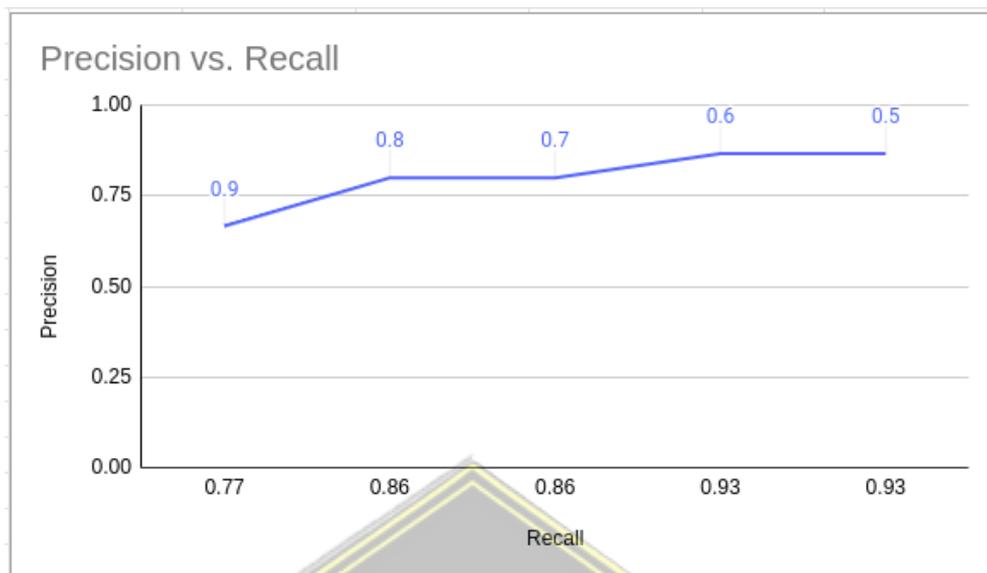
3. Pengujian pada *confidence threshold* 0.7 menghasilkan keluaran 12 *True Positive* dan 11 *True Negative*, dengan nilai *Precision* 80%, *Recall* 86%, dan *Accuracy* 82%.
4. Pengujian pada *confidence threshold* 0.6 menghasilkan keluaran 13 *True Positive* dan 12 *True Negative*, dengan nilai *Precision* 87%, *Recall* 93%, dan *Accuracy* 89%.
5. Pengujian pada *confidence threshold* 0.5 menghasilkan keluaran 13 *True Positive* dan 12 *True Negative*, dengan nilai *Precision* 87%, *Recall* 93%, dan *Accuracy* 89%.

Berikut merupakan tabel evaluasi pengujian model pada perangkat Android:

Tabel 4. 2 Evaluasi pengujian sistem

No	<i>Prediction</i>	<i>Threshold</i>				
		0.9	0.8	0.7	0.6	0.5
1	<i>True Positive</i>	10	12	12	13	13
2	<i>True Negative</i>	10	11	11	12	12
3	<i>False Positive</i>	5	3	3	2	2
4	<i>False Negative</i>	3	2	2	1	1
5	<i>Precision</i> $TP/(TP+FP)$	67%	80%	80%	87%	87%
6	<i>Recall</i> $TP/(TP+FN)$	77%	86%	86%	93%	93%
7	<i>Accuracy</i> $TP+TN/(TP+TN+FP+FN)$	71%	82%	82%	89%	89%

Evaluasi menunjukkan bahwa *threshold* lebih rendah menghasilkan prediksi yang lebih akurat. Karena model dikembangkan untuk membantu proses deteksi masker dan *False Positive* bukan merupakan resiko serius, maka aplikasi dapat secara aman menggunakan *threshold* rendah demi mendapatkan akurasi hasil prediksi yang lebih tinggi.



Gambar 4. 2 Hasil Evaluasi *Precision vs Recall*

Berdasarkan evaluasi hasil pengujian di atas dapat disimpulkan bahwa model berjalan dengan baik dan aplikasi dapat menjalankan modul MobileNet dan dapat melaksanakan fungsi deteksi penggunaan masker wajah berdasarkan hasil tangkapan foto tampak depan dengan cukup baik dengan akurasi mencapai 89%.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Melalui penelitian ini dapat kita lihat sistem deteksi masker wajah menggunakan MobileNet pada perangkat Android dapat bekerja dengan baik dengan tingkat akurasi mencapai 89% pada ruangan dengan pencahayaan yang baik sehingga dapat disimpulkan bahwa sistem bekerja sesuai yang diharapkan.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, peneliti memiliki beberapa saran pengembangan model yang telah dibuat:

1. Model yang dikembangkan tidak dapat mendeteksi masker berwarna. Penelitian lebih lanjut dapat dilakukan untuk menghasilkan model yang dapat melakukan deteksi masker berwarna.
2. Pengujian pada ruangan dengan pencahayaan terlalu rendah atau terlalu tinggi mengakibatkan hasil prediksi menjadi tidak akurat dan cenderung menghasilkan prediksi tidak sesuai dengan kenyataan. Penelitian lebih lanjut dapat dilakukan untuk mengatasi masalah ini.

DAFTAR PUSTAKA

- Bhardwaj, A. (2020) *What is the Softmax Function? — Teenager Explains*. Available at: <https://towardsdatascience.com/what-is-the-softmax-function-teenager-explains-65495eb64338> (Accessed: 20 November 2022).
- Howard, A.G. *et al.* (2017) 'MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications'. Available at: <http://arxiv.org/abs/1704.04861>.
- Jonathan Grandperrin (2021) *How to use confidence scores in machine learning models*. Available at: <https://mindee.com/blog/how-use-confidence-scores-ml-models/> (Accessed: 13 December 2022).
- Matthijs Hollemans (2017) *Google's MobileNets on the iPhone*. Available at: <https://machinethink.net/blog/googles-mobile-net-architecture-on-iphone/> (Accessed: 12 June 2022).
- Matthijs Hollemans (2018) *MobileNet version 2*. Available at: <https://machinethink.net/blog/mobilenet-v2/> (Accessed: 17 October 2022).
- Mungkasa, O. (2020) 'Working from Home (WFH): Towards a New Order in the Era of the COVID-19 Pandemic', *Jurnal Perencanaan Pembangunan: The Indonesian Journal of Development Planning*, 4(2), pp. 126–150.
- Nyoman, P. and Putu Kusuma Negara (2021) 'Deteksi Masker Pencegahan Covid19 Menggunakan Convolutional Neural Network Berbasis Android', *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 5(3), pp. 576–583. Available at: <https://doi.org/10.29207/resti.v5i3.3103>.
- Paul-Louis Pröve (2018) *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. Available at: <https://towardsdatascience.com/mobilenetv2-inverted-residuals-and-linear-bottlenecks-8a4362f4ffd5> (Accessed: 17 October 2022).
- Sandler, M. *et al.* (2018) 'MobileNetV2: Inverted Residuals and Linear Bottlenecks', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520. Available at: <https://doi.org/10.1109/CVPR.2018.00474>.
- Sumeet Kumar Agrawal (2021) *Metrics to Evaluate your Classification Model to*

take the right decisions. Available at: <https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/> (Accessed: 13 December 2022).

Sumit Saha (2018) *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.* Available at: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> (Accessed: 20 November 2022).

SuperDataScience Team (2018) *Convolutional Neural Networks (CNN): Step 1(b) - ReLU Layer.* Available at: <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-1b-relu-layer> (Accessed: 20 November 2022).

Ting-Hao Chen (2017) *What is “stride” in Convolutional Neural Network?* Available at: <https://medium.com/machine-learning-algorithms/what-is-stride-in-convolutional-neural-network-e3b4ae9baedb> (Accessed: 18 October 2022).

Hongkun Yu, Chen Chen, Xianzhi Du, Yeqing Li, Abdullah Rashwan, Le Hou, Pengchong Jin, Fan Yang, Frederick Liu, Jaeyoun Kim, J. L. (2020) *TensorFlow Model Garden.* Available at: <https://github.com/tensorflow/models> (Accessed: 22 November 2021).