

**IMPLEMENTASI DEEP LEARNING MENGGUNAKAN
FRAMEWORK TENSORFLOW DENGAN METODE
CONVOLUTIONAL NEURAL NETWORK UNTUK
MENDETEKSI MASKER WAJAH**

SKRIPSI

PROGRAM STUDI TEKNIK INFORMATIKA



FAKULTAS TEKNIK

UNIVERSITAS SATYA NEGARA INDONESIA

JAKARTA

2021

**IMPLEMENTASI DEEP LEARNING MENGGUNAKAN
FRAMEWORK TENSORFLOW DENGAN METODE
CONVOLUTIONAL NEURAL NETWORK UNTUK
MENDETEKSI MASKER WAJAH**

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar

SARJANA KOMPUTER



OLEH :

NAMA : FAHRI HUSAINI

NIM : 011701503125085

FAKULTAS TEKNIK

UNIVERSITAS SATYA NEGARA INDONESIA

JAKARTA

2021

***IMPLEMENTATION OF DEEP LEARNING USING
TENSORFLOW FRAMEWORK WITH CONVOLUTIONAL
NEURAL NETWORK METHOD TO DETECT FACE MASK***

UNDERGRADUATE THESIS

Presented As One Of Terms To Acquire Degree

BACHELOR OF ENGINEERING

Technical Information



BY:

NAME : FAHRI HUSAINI

NIM : 011701503125085

FACULTY OF ENGINEERING

UNIVERSITAS OF SATYA NEGARA INDONESIA

JAKARTA

2021

SURAT PERNYATAAN KARYA SENDIRI

Yang bertanda tangan dibawah ini :

Nama : Fahri Husaini

Nim : 011701503125085

Program Studi : Teknik Infomatika

Menyatakan bahwa skripsi ini adalah murni hasil karya sendiri dan seluruh isi Skripsi ini menjadi tanggung jawab saya sendiri. Apabila saya mengutip dari karya orang lain maka saya mencantumkan sumbernya sesuai dengan ketentuan yang berlaku. Saya bersedia dikenai sanksi pembatalan Skripsi ini apabila terbukti melakukan Tindakan plagiat (penjiplakan).

Demikian pernyataan ini saya buat dengan sebenarnya.

Bekasi, Agustus 2021



(Fahri Husaini)

011701503125085

LEMBAR PENGESAHAN SKRIPSI / TUGAS AKHIR

Nama : Fahri Husaini

Nim : 011701503125085

Program Studi : Teknik Informatika

Judul Skripsi : Implementasi Deep Learning Menggunakan Framework
TensorFlow Dengan Metode Convolutional Neural Network
Untuk Mendeteksi Masker Wajah

Tanggal Ujian : 17 Agustus 2021
JAKARTA, 17 Agustus 2021

Dosen Pembimbing I

Dosen Pembimbing II

(Hernalom Sitorus S.T., M.Kom) (Wawan Kurniawan S.Kom., M.Kom)



Dekan

Ketua Program Studi

(Ir.Nurhayati M.Si)

(Istiqomah Sumadikarta S.T., M.Kom)

LEMBAR PENGESAHAN PENGUJI**IMPLEMENTASI DEEP LEARNING MENGGUNAKAN FRAMEWORK
TENSORFLOW DENGAN METODE CONVOLUTIONAL NEURAL NETWORK
UNTUK MENDETEKSI MASKER WAJAH**

Anggota Penguji

(Berlin P Sitorus S.Kom., M.Kom)

Anggota Penguji

(Faisal Zuli S.Kom., M.Kom)

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT, karena berkat rahmat dan karunia-Nyalah penulis dapat menyelesaikan skripsi yang berjudul “Implementasi *Deep Learning* Menggunakan *Framework TensorFlow* Dengan Metode *Convolution Neural Network* Untuk Mendeteksi Masker Wajah”. Dalam skripsi ini dibahas mengenai penggunaan *deep learning* dengan model *convolutional neural network* dalam memprediksi pengguna masker wajah yang menggunakan *Framework TensorFlow* sebagai *library* dalam implementasi program dengan Python sebagai Bahasa pemrograman yang digunakan. Adapun maksud dan tujuan dari penulisan skripsi ini adalah untuk memenuhi salah satu syarat untuk mengikuti sidang skripsi, Jurusan Teknik Informatika, Fakultas Teknik Universitas Satya Negara Indonesia, Bekasi.

Selama penelitian dan penulisan skripsi ini banyak sekali hambatan yang penulis alami, namun berkat bantuan, dorongan serta bimbingan dari berbagai pihak, akhirnya skripsi ini dapat terselesaikan dengan baik.

Penulis beranggapan bahwa skripsi ini merupakan karya terbaik yang dapat penulis persembahkan. Tetapi penulis menyadari bahwa tidak tertutup kemungkinan didalamnya terdapat kekurangan – kekurangan. Oleh karena itu kritik dan saran yang membangun sangat penulis harapkan. Akhir kata, semoga skripsi ini dapat bermanfaat bagi penulis khususnya dan bagi para pembaca pada umumnya.

Bekasi , Agustus 2021

Penulis,



Fahri Husaini

UCAPAN TERIMA KASIH

Dalam Penyusunan skripsi ini, penulis banyak mendapatkan bantuan, bimbingan serta petunjuk dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terimakasih yang sebesar – besarnya kepada yang terhormat:

1. Ir.Nurhayati M.Si selaku Dekan Fakultas Teknik Universitas Satya Negara Indonesia yang telah memberikan izin kepada penulis untuk melakukan penelitian.
2. Istiqomah Sumadikarta S.Kom., M.Kom selaku Ketua Jurusan Program Studi Teknik Informatika Universitas Satya Negara Indonesia atas petunjuk dan nasehatnya kepada penulis.
3. Abdul Kholiq S.Kom., M.Kom selaku dosen Pembimbing Akademik yang telah memberikan bimbingan selama penulis menyelesaikan studi di Jurusan Teknik Informatika di Universitas Satya Negara Indonesia.
4. Hernalom Sitorus S.T., M.Kom, selaku dosen Pembimbing I, yang telah memberikan bimbingan, arahan, dorongan kepada penulis sehingga penyusunan skripsi ini berjalan dengan baik.
5. Wawan Kurniawan S.Kom., M.Kom, selaku dosen Pembimbing II, atas segala bimbingan dan saran kepada penulis sehingga penulis dapat menyelesaikan skripsi ini.
6. Ayahanda Matsani dan Ibunda Ernawati tercinta atas segala semangat, bantuan, dorongan serta doa restu yang diberikan kepada penulis selama penyusunan skripsi.
7. Adik – adik tersayang Fahmi Fadilah dan Fani Septiani atas dorongan yang diberikan kepada penulis.
8. Teman Wanita Amelia Susanti atas segala bantuan, dorongan serta motivasi penulis selama penyusunan skripsi.

9. Vicky Ramadani Eky Saski selaku sahabat penulis atas bantuan dan dorongan yang diberikan kepada penulis.
10. Rekan – Rekan Mahasiswa Jurusan Teknik Informatika Angkatan 2017 atas segala bantuan dan kerjasamanya.
11. Semua yang tidak bisa penulis sebutkan satu per satu. Pastinya Tak henti – henti penulis sampaikan semoga amal baik semua pihak mendapat balasan yang berlipat ganda dari sang pencipta yang pengasih dan penyayang Alla SWT. Amin



ABSTRAK

Universitas Satya Negara Indonesia sudah menerapkan protokol Kesehatan pada lingkungan kampus, Penerapan protokol Kesehatan di lingkungan kampus dengan menyediakan *security* yang berjaga dipintu masuk kampus untuk mengecek kelengkapan protokol Kesehatan termasuk penggunaan masker, dengan menyediakan tenaga manusia untuk menerapkan protokol Kesehatan di lingkungan kampus memungkinkan terjadinya tidak konsistensinya penerapan protokol Kesehatan karena *security* yang bertugas memiliki waktu untuk istirahat dan keluar dari area penjagaan untuk mengerjakan tugas lain. Penelitian ini bertujuan untuk menerapkan penggunaan *Deep Learning* model *Convolutional Neural Network* menggunakan *Framework TensorFlow* untuk dapat memprediksi penggunaan masker wajah dan mengimplementasikan nya dengan membuat sistem pendeteksi masker wajah secara *realtime*.

Deep learning sebagai metode pembelajaran mesin yang dapat merepresentasikan data ke mesin untuk memprediksi suatu hal yang masih abstrak. *Deep Learning* memiliki model pembelajaran salah satunya *Convolution Neural Network* yang dapat digunakan untuk *computer vision*, *face recognition* dan *object detection* yang bisa menggunakan objek berupa gambar array video dengan didukung menggunakan *Framework TensorFlow* dapat memprediksi penggunaan masker wajah dengan nilai akurasi tertinggi sebesar 99,7 %, nilai loss sebesar 0,017 % , validasi akurasi sebesar 99,7% dan validasi loss sebesar 0,016%. Pada pengujian kemampuan sistem dalam mendeteksi penggunaan masker dari berbagai kondisi, sistem berhasil mendeteksi penggunaan masker secara *realtime* dengan nilai akurasi 90%.

Kata Kunci: *Deep Learning, Convolutional Neural Network, TensorFlow, Sistem deteksi masker*

ABSTRACT

University Of Satya Negara Indonesia, has applied health protocols to the university's environment, implementing campus health protocols by providing security guards at the university entrance to check for complete health protocols, including the use of masks, by providing human beings to implement health protocols in the campus environment, the application of health protocols is possible, as the security on duty has time for rest and escape from the security area for another task. The research aims to know the use of deep learning models convolutional neural network using the framework TensorFlow neural network to predict the use of facial masks and implement them by creating a fact-mask system.

Deep learning as a machine learning method that can represent data to machines to predict something that is still abstract. Deep learning has a learning model, one of which is a convolution neural network that can be used for computer vision, face recognition and object detection that can use objects in the form of video array images supported using the TensorFlow framework to predict the use of face masks with the highest accuracy value of 99.7 %, the loss value is 0.017 %, the validation accuracy is 99.7% and the loss validation is 0.016%. In testing the system's ability to detect the use of masks from various conditions, the system succeeded in detecting the use of masks in real time with an accuracy value of 90%.

Keywords: *Deep Learning, Convolutional Neural Network, TensorFlow, Mask detection system*

DAFTAR ISI

Halaman

HALAMAN JUDUL.....	
SURAT PERNYATAAN KARYA SENDIRI.....	i
LEMBAR PENGESAHAN SKRIPSI / TUGAS AKHIR.....	ii
LEMBAR PENGESAHAN PENGUJI.....	iii
KATA PENGANTAR.....	iv
UCAPAN TERIMA KASIH.....	v
ABSTRAK.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN.....	1
A. Latar Belakang.....	1
B. Rumusan Masalah.....	2
C. Batasan Masalah.....	2
D. Tujuan dan Manfaat.....	3
D.1 Tujuan.....	3
D.2 Manfaat.....	3
E. Sistematis Penulisan.....	4
BAB II LANDASAN TEORI.....	6
A. Tinjauan Pustaka.....	6
B. Pengolahan Citra Digital.....	7
B.1 Resolusi Citra.....	7
B.2 Jenis Citra.....	8

B.3	Konversi Citra <i>Grayscale</i>	11
C.	<i>Face Recognition</i>	13
D.	Deep Learning	13
D.1	Arstitektur Deep Learning	14
D.2	Model Deep Learning	15
E.	Convolution Neural Network	17
E.1	Neural Network.....	18
E.2	Convolution Layer.....	19
F.	Tensorflow.....	24
G.	OpenCV (Open Source Computer Vision Library)	25
H.	Python.....	27
I.1	Tipe Data.....	27
I.2	Operasi Dasar.....	28
I.	Webcam Eksternal.....	31
BAB III	METODE PENELITIAN	33
A.	Waktu Dan Tempat	33
A.1	Waktu Penelitian.....	33
A.2	Tempat Penelitian.....	33
B.	Analisa Kebutuhan Sistem	33
B.1	Perangkat Keras (<i>Hardware</i>).....	33
B.2	Perangkat Lunak (<i>Software</i>)	34
C.	Pengumpulan Data (<i>Data Gathering</i>)	34
D.	Prosedur Perancangan Sistem	35
E.	Konsep Umum Sistem.....	40
F.	Kerangka Berfikir.....	40

F. Timeline Penelitian	43
BAB IV IMPLEMENTASI DAN EVALUASI SISTEM	44
A. Umum.....	44
B. Distribusi Gambar	44
C. Augmentasi Gambar Sample.....	45
D. Pengujian Hasil Training Dataset.....	47
E. Uji Coba dan Evaluasi.....	50
F. Pengujian Kemampuan Sistem Mendeteksi Masker.....	52
G. Pengujian Quality Of Service (QoS).....	61
BAB V KESIMPULAN DAN SARAN	64
A. KESIMPULAN.....	64
B. SARAN	65
DAFTAR PUSTAKA.....	67
LAMPIRAN.....	68

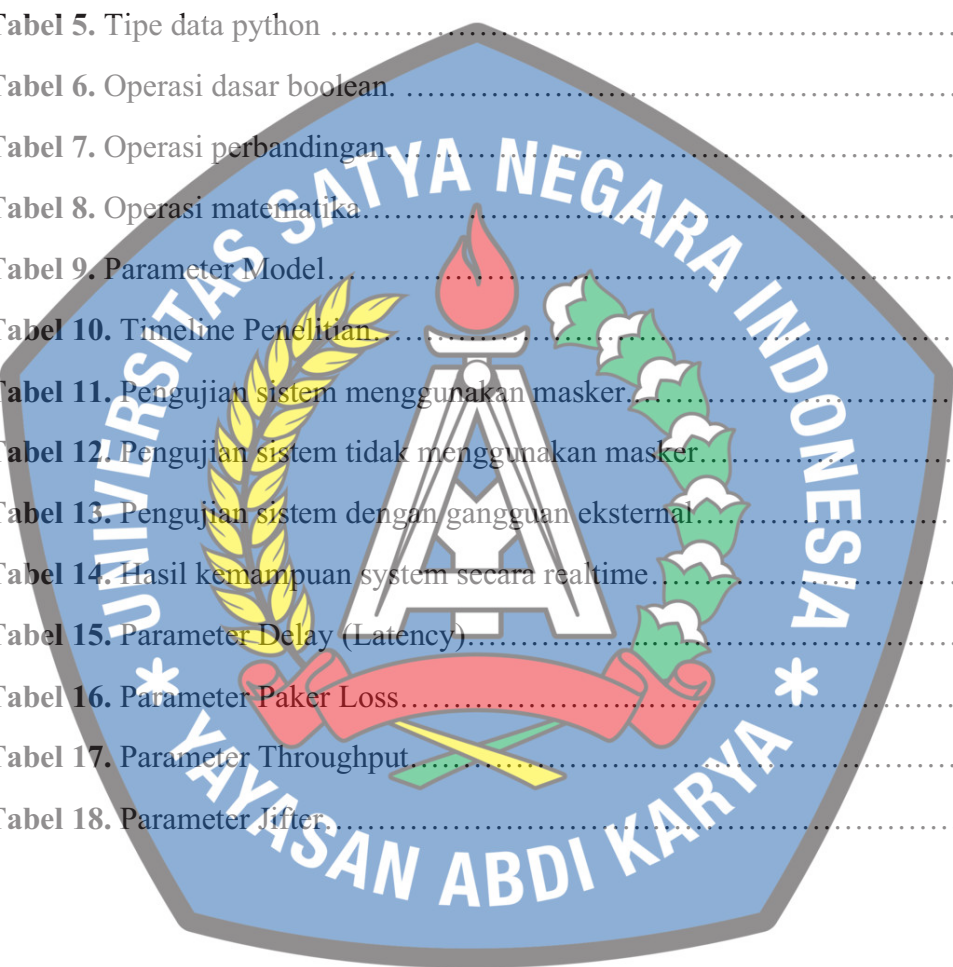


DAFTAR GAMBAR

Gambar 1. Citra biner.....	9
Gambar 2. Citra grayscale.....	9
Gambar 3. 8 bit truecolor.....	10
Gambar 4. Arsitektur <i>Deep Learning</i>	15
Gambar 5. CNN tiga layer.....	18
Gambar 6. Contoh Neural Network.....	18
Gambar 7. Sample gambar data masukan.....	20
Gambar 8. Proses konvolusi dengan karnel/feature 3x3.....	21
Gambar 9. Sample pergeseran dengan nilai stride =1 dan karnel 3x3.....	22
Gambar 10. Proses konvolusi pergeseran pertama.....	22
Gambar 11. Proses konvolusi pergeseran terakhir.....	22
Gambar 12. Ilustrasi <i>Convolution layer</i>	24
Gambar 13. Bagian – Bagian pada OpenCV.....	27
Gambar 14. Webcam eksternal.....	32
Gambar 15. Diagram alir (flowchart) penelitian.....	36
Gambar 16. *Konsep umum sistem.....	40
Gambar 17. Kerangka Berfikir.....	41
Gambar 18. Distribusi gambar.....	45
Gambar 19. Augmentasi citra sample.....	46
Gambar 20. Hasil training dataset.....	47
Gambar 21. Train dan Validation loss... ..	48
Gambar 22. Train dan Validasi Accuracy.. ..	49
Gambar 23. Rumus <i>Confusion Matrix</i>	50
Gambar 24. <i>Confusion Matrix</i>	50
Gambar 25. Hasil prediksi <i>with_mask</i>	52

DAFTAR TABEL

Tabel 1. Contoh nilai piksel citra berwarna.....	11
Tabel 2. Contoh nilai piksel citra grayscale.....	12
Tabel 3. Model <i>Deep Learning</i>	15
Tabel 4. Contoh input dan output Nerual Network.....	19
Tabel 5. Tipe data python	28
Tabel 6. Operasi dasar boolean.	28
Tabel 7. Operasi perbandingan.....	29
Tabel 8. Operasi matematika.....	30
Tabel 9. Parameter Model.....	37
Tabel 10. Timeline Penelitian.....	43
Tabel 11. Pengujian sistem menggunakan masker.....	53
Tabel 12. Pengujian sistem tidak menggunakan masker.....	54
Tabel 13. Pengujian sistem dengan gangguan eksternal.....	57
Tabel 14. Hasil kemampuan system secara realtime.....	60
Tabel 15. Parameter Delay (Latency).....	61
Tabel 16. Parameter Paker Loss.....	62
Tabel 17. Parameter Throughput.....	63
Tabel 18. Parameter Jitter.....	63



BAB I

PENDAHULUAN

A. Latar Belakang

Dimasa Pandemi Covid 19 saat ini, pemerintah Indonesia meminta untuk seluruh elemen masyarakat maupun organisasi negara dan swasta untuk menerapkan *Social Distancing* atau menjaga jarak dalam rangka menekan angka penularan Virus Covid 19. Salah satunya dengan menghindari kontak langsung antar seseorang dilingkungan masyarakat maupun lingkungan kerja dan mewajibkan penggunaan masker pada masyarakat yang ingin berpergian keluar rumah, Rajin untuk mencuci tangan menggunakan sabun dan *hand sanitaizer*. Pemerintah juga membatasi kunjungan ke tempat – tempat umum yang dapat menimbulkan kerumunan dengan syarat mematuhi protokol kesehatan dengan tetap menggunakan masker. Tempat – tempat seperti sekolah, mall dan kampus merupakan tempat yang berpotensi menimbulkan keramaian.

Universitas Satya Negara Indonesia khususnya kampus B pada proses pelaksanaan kepatuhan ketentuan protokol kesehatan sudah menyediakan 1 orang *security* yang ditugaskan untuk memeriksa kelengkapan dan prosedur protokol kesehatan sebelum masuk kedalam kampus. Dengan digunakan nya sistem pengawasan yang dilakukan oleh manusia memungkinkan kurang konsistensi nya dalam pengawasan kepatuhan protokol kesehatan, karena tenaga security yang bertugas mengawasi protokol kesehatan memiliki waktu untuk istirahat dan keluar dari area pintu masuk kampus di sebabkan ada nya tugas lain yang menyebabkan cepat. Kelemahan dari sistem yang saat ini dipakai adalah kurangnya efisiensi tidak

terkontrolnya pemeriksaan protokol kesehatan khusus nya dalam pengawasan penggunaan masker secara *realtime* ketika *security* tersebut tidak berada pada tempat pemeriksaan kelengkapan dan prosedur protokol kesehatan. Dari permasalahan tersebut penulis mengusulkan sebuah sistem otomatis yang berfungsi untuk mendeteksi penggunaan masker di Universitas Satya Negara Indonesia kampus B secara *Realtime* untuk memonitoring penggunaan masker wajah pada mahasiswa, staf dan dosen yang hendak masuk ke area Kampus Universitas Satya Negara Indonesia Kampus B. Sistem yang akan dibangun menggunakan *Deep Learning* yang dan *Framework* Tensorflow, dengan metode *Convolutional Neural Network* (CNN) yang di integrasikan ke usb webcam untuk dapat mendeteksi penggunaan masker wajah pada Kampus Universitas Satya Negara Indonesia Kampus B

B. Rumusan Masalah

Berdasarkan uraian latar belakang masalah diatas, dapat ditarik rumusan masalah pada penelitian ini sebagai berikut : Bagaimana cara menerapkan *Deep Learning* menggunakan *framework* TensorFlow dengan metode *Convolutional Neural Networks* untuk mendeteksi masker wajah ?

C. Batasan Masalah

Dalam mengatasi permasalahan diatas, Maka cakupan masalah akan dibatasi, Yaitu sebagai berikut :

- a. Tidak membahas dasar dari *framework* TensorFlow diluar penggunaannya untuk *face recognition*.

- b. Tidak membahas fungsi – fungsi dari *library* pendukung lainnya pada perancangan *system* seperti proses pembacaan citra, proses komputasi citra di dalam *system* dan lain – lain.
- c. Tidak membahas dasar – dasar *deep learning* selain penggunaan nya untuk mendeteksi masker

D. Tujuan dan Manfaat

Adapun tujuan dan manfaat dalam penelitian ini adalah sebagai berikut :

D.1 Tujuan

1. Menerapkan *deep learning* dengan *framework tensorflow* untuk mendeteksi penggunaan masker wajah.
2. Menerapkan metode *Convolutional Neural Networks* untuk training dataset dan menghasilkan model.
3. Menerapkan teknologi pengenalan wajah (*face recognition*) dalam penggunaan protokol kesehatan covid-19

D.2 Manfaat

1. Dapat Membantu Kampus Universitas Satya Negara Indonesia Kampus B dalam monitoring penggunaan masker wajah.
2. Dapat memaksimalkan teknologi *face recognition* yang di implementasikan untuk system mendeteksi masker dan menghasilkan *prototype* yang nantinya dapat dikembangkan serta diterapkan secara langsung.

3. Dapat Mengatasi permasalahan tidak konsisten nya pemeriksaan penggunaan masker.

E. Sistematis Penulisan

Untuk mempermudah pemahaman, skripsi ini ditulis dengan sistematika penulisan sebagai berikut :

BAB I: PENDAHULUAN

Bab ini terdiri dari latar belakang masalah, pembatasan masalah, perumusan masalah, tujuan, manfaat penelitian dan sistematika penulisan.

BAB II: LANDASAN TEORI

Bab ini membahas tentang tinjauan pustaka, teori-teori mengenai kerangka berpikir, serta teori-teori lain yang berkaitan dengan pembahasan ini.

BAB III: METODE PENELITIAN

Bab ini menjelaskan tentang penjabaran metode penelitian, jenis penelitian, variable penelitian, tahapan penelitian, teknik pengumpulan data serta rencana analisis data.

BAB IV: PEMBAHASAN

Bab ini menjelaskan tentang diskripsi obyektif penelitian berdasarkan metode penelitian yang dilakukan untuk melakukan pembahasan dalam penelitian ini.

BAB V: PENUTUP

Bab ini menjelaskan mengenai kesimpulan dari hasil penelitian dan saran yang sekiranya dapat dipertimbangkan lebih lanjut.



BAB II

LANDASAN TEORI

A. Tinjauan Pustaka

Pada penelitian Pulung Adi Nugroho, dkk (2020) dengan judul “Implementasi *Deep Learning* Menggunakan *Convolutional Neural Network* (CNN) Pada Ekspresi Manusia”. Penelitian ini bertujuan untuk mengetahui ekspresi manusia dengan Epoch 100 batch size 128 dan akurasi yang didapatkan 90% dan validation sebesar 65% dari total 35 ekspresi yang berhasil ditebak dengan benar yaitu 28 ekspresi.

Pada penelitian Dufan J.P Manajang, dkk (2020) dengan judul “Implementasi *Framework Tensorflow Object Detection* Dalam Mengklasifikasi Jenis Kendaraan Bermotor”. Penelitian ini bertujuan untuk mengklasifikasi kendaraan yang lewat disuatu lintas jalan dengan tingkat akurasi deteksi objek rata – rata adalah 90,8%.

Pada penelitian Arham Rahim, Dkk (2020) dengan judul *Convolutional Neural Network Untuk Klasifikasi Penggunaan Masker*. Hasil penelitian ini menunjukkan skenario kedua yang menggunakan epoch 50 dan rasio dataset 90% data latih dan 10% data uji mendapatkan akurasi terbaik mencapai 96%. Pengujian pada gambar wajah yang menggunakan masker memperoleh nilai precision 98%, recall 94% dan gambar wajah yang tidak menggunakan masker memperoleh nilai precision 94%, recall 98. Skenario satu dan tiga memperoleh nilai akurasi terendah yaitu 94%.

Pada Penelitian Ahmad Kurniadi, Dkk (2020) yang berjudul “ Implementasi Convolutional Neural Network Untuk Klasifikasi Varietas Pada Citra Daun Sawi Menggunakan Keras”. Penelitian ini menghasilkan data uji coba dari klasifikasi citra pada sayuran varietas sawi yaitu sawi pakcoy, sawi putih, dan sawi caisim menghasilkan nilai akurasi sebesar 83%, recall 80% dan presisi 89%. Ketika data training dilakukan perbedaan dengan jumlah perbandingannya maka hasil yang didapatkan adalah sama yaitu 83%

B. Pengolahan Citra Digital

Menurut Darma Putra (2010:19) Pengolahan citra digital mengacu pada penggunaan komputer untuk memproses gambar dua dimensi. Dalam konteks yang lebih luas, pengolahan citra digital mengacu pada pengolahan data dua dimensi. Citra digital adalah larik yang berisi nilai nyata atau kompleks yang diwakili oleh deretan bit tertentu.

Suatu citra dapat didefinisikan sebagai suatu fungsi (x, y) dengan ukuran M baris dan N kolom, dimana x dan y adalah koordinat spasial, dan amplitudo f pada titik koordinat (x, y) disebut intensitas atau abu-abu gambar pada tingkat derajat saat itu. Jika nilai x dan y serta besar f diskrit berhingga, maka bayangan tersebut dapat dikatakan sebagai citra digital. adalah citra digital. Darma Putra (2010:19)

B.1 Resolusi Citra

Menurut Darma Putra (2010:38) Resolusi gambar adalah tingkat detail dalam gambar. Semakin tinggi resolusi gambar, semakin besar tingkat detail gambar. Satuan untuk mengukur resolusi gambar dapat berupa ukuran fisik

(garis per milimeter/garis per inci) atau ukuran keseluruhan gambar (garis per tinggi gambar). Resolusi sebuah citra dapat di ukur dengan berbagai cara sebagai berikut :

1. Resolusi pixel
2. Resolusi spasial
3. Resolusi spectral
4. Resolusi temporal
5. Resolusi radiometric

B.2 Jenis Citra

Menurut Darma Putra (2010:44) terdapat 5 Jenis citra yang berdasarkan nilai pixel yaitu : Citra biner , Citra Grayscale, Citra warna 8 bit, Citra warna 16 bit, citra warna 24 bit.

1. Citra biner merupakan Sebuah citra digital dengan hanya dua kemungkinan nilai piksel, yaitu hitam dan putih. Gambar biner disebut juga gambar hitam putih (black and white) atau gambar monokrom. Hanya 1 bit yang dibutuhkan untuk merepresentasikan nilai setiap piksel dalam citra biner. Citra biner sering muncul sebagai hasil dari proses pemrosesan seperti segmentasi, ekspansi, morfologi, atau pengaburan.



Gambar 1. Citra biner

2. Citra Grayscale Untuk citra digital, setiap piksel hanya memiliki satu nilai kanal, yaitu nilai $RED = GREEN = BLUE$. Nilai ini digunakan untuk menunjukkan tingkat intensitas. Warna mereka hitam, abu-abu dan putih. Skala abu-abu di sini adalah
3. warna abu-abu dari berbagai tingkatan mulai dari hitam hingga mendekati putih.



Gambar 2. Citra *grayscale*

4. Citra warna (8 bit) hanya diwakili oleh 8 bit dengan jumlah warna maksimum yang dapat digunakan adalah 256 warna. Ada dua jenis citra warna 8 bit. Pertama, citra warna 8 bit dengan menggunakan palet warna 256 dengan setiap paletnya memiliki pemetaan nilai (*colormap*) RGB tertentu. Model ini

lebih sering digunakan. Kedua, setiap pixel memiliki format 8 bit sebagai berikut :

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
R	R	R	G	G	G	B	B

Bentuk kedua ini dinamakan 8 bit truecolor . berikut ini adalah warnanya :



Gambar 3. 8 bit truecolor

5. Citra Warna (16 bit) biasanya disebut sebagai citra high color dengan setiap pixelnya diwakili dengan 2 byte memory (16 bit). Warna 16 Bit memiliki 65.536 warna. Dalam formasi bitnya , nilai merah dan biru mengambil tempat di 5 bit dikanan dan kiri. Komponen hijau memiliki 5 bit ditambah 1 bit ekstra. Pemilihan komponen hijau dengan deret 6 bit dikarenakan penglihatan manusia lebih sensitive terhadap warna hijau.
6. Citra Warna (24 bit) setiap pixel dari citra warna 24 bit diwakili dengan 24 bit sehingga total 16.777.216 variasi warna. Variasi ini sudah lebih dari cukup untuk memvisualisasikan seluruh warna yang dapat dilihat penglihatan manusia. Penglihatan manusia dipercaya hanya dapat membedakan hingga 10 juta

warna saja. Setiap point informasi pixel (RGB) disimpan ke dalam 1 *byte* data. 8 bit pertama menyimpan nilai biru, kemudian di ikuti dengan nilai hijau pada 8 bit kedua dan 8 bit terakhir merupakan warna merah.

B.3 Konversi Citra *Grayscale*

Pada proses pengolahan citra digital terdapat beberapa proses konversi citra salah satunya proses konversi dari citra berwarna ke citra *grayscale* dimana pada proses konversi tersebut agar dapat merubah citra berwarna menjadi citra *grayscale* diperlukan rumus sebagai berikut :
sebelum nya sebagai contoh citra dengan ukuran piksel 3x3 dengan nilai pada masing – masing citra sebagai berikut :

Tabel 1. Contoh Citra ukuran berwarna

(i,j)	0	1	2
0	31 33 22	42 44 33	46 48 37
1	42 44 33	44 46 35	64 66 55
2	188 189 181	186 187 179	193 194 186

Kemudian dikalikan dengan rumus *grayscale* pada masing – masing kordinat sehingga menghasilkan nilai kordinat baru atau nilai piksel baru berikut contoh perhitungan dari citra piksel 8x8 sebagai berikut :

$$Gray [0,0] = \frac{31 + 33 + 22}{3} = 28,7$$

$$Gray [0,1] = \frac{42 + 44 + 33}{3} = 39,6$$

$$Gray [0,2] = \frac{46 + 48 + 37}{3} = 43,6$$

$$Gray [1,0] = \frac{42 + 44 + 33}{3} = 39,6$$

$$Gray [1,1] = \frac{44 + 46 + 35}{3} = 41,6$$

$$Gray [1,2] = \frac{64 + 66 + 55}{3} = 61,6$$

$$Gray [2,0] = \frac{188 + 189 + 181}{3} = 186$$

$$Gray [2,1] = \frac{186 + 187 + 179}{3} = 184$$

$$Gray [2,2] = \frac{193 + 194 + 186}{3} = 191$$

Dari perhitungan tersebut didapati nilai matriks dengan titik kordinat baru yaitu sebagai berikut :

Tabel 2. Contoh citra hasil konversi grayscale

(i,j)	0	1	2
0	29	40	44
1	40	42	62

2	188	184	191

Perhitungan tersebut dilakukan untuk semua kordinat, sehingga matriks citra hasil konversi akan berubah nilai dari matriks citra sebelum nya.

C. *Face Recognition*

Menurut Dodit Suprianto,dkk (2018:180) dalam Jurnal “ Sistem penegenalan wajah secara Real Time dengan Adaboost , Eigenface PCA & MySql” Face Recognition atau pengenalan wajah merupakan mengutip informasi unik wajah , kemudian di- encode dan dibandingkan dengan hasil decode yang sebelumnya dilakukan.

Menurut Rinaldo Singgalen (2017:150) dalam jurnal “ Sistem Pengenalan Wajah Sebagai Akses Loker Penyimpanan Barang” Face Recognition Merupakan salah satu teknologi biometrics yang telah dipelajari dan dikembangkan banyak oleh para ahli, dimana perangkat tersebut menggunakan individu yang satu dengan yang lain berdasarkan individu yang satu dengan yang lain nya berdasarkan data yang sudah ada.

D. *Deep Learning*

Deep Learning telah merevolusi industri teknologi. Penerjemahan mesin Modern, mesin pencari, dan asisten komputer semuanya didukung oleh deep learning. Kecenderungan ini hanya akan terus berlanjut seraya proses pembelajaran

yang mendalam memperluas jangkauannya menuju robotika, farmasi yang diperalat, energi, dan semua bidang teknologi modern lainnya. Ini segera menjadi penting bagi perangkat lunak modern profesional untuk mengembangkan pengetahuan bekerja tentang prinsip-prinsip pembelajaran mendalam. Bharath Ramsundar (2018)

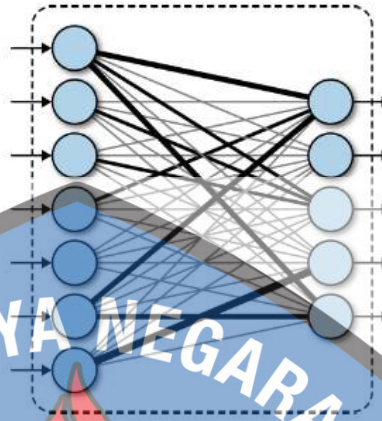
Menurut Mohik Sewak (2018) *Deep Learning* adalah semacam metode pembelajaran mesin yang didasarkan pada representasi data pembelajaran yang bertentangan dengan algoritma spesifik tugas. *Deep Learning* memungkinkan komputer untuk membangun konsep yang rumit dari konsep yang lebih sederhana dan lebih kecil. Misalnya, sistem belajar yang dalam mengenali gambar seseorang dengan menggabungkan ujung-ujung label yang lebih rendah dan menggabungkannya ke dalam berbagai bagian tubuh dengan cara yang hierarkis. Waktunya tidak begitu jauh ketika *Deep Learning* akan diperluas ke aplikasi yang memungkinkan mesin untuk berpikir sendiri.

D.1 Arsitektur Deep Learning

Menurut Bharath Ramsundar (2018) Sebagian besar arsitektur yang dalam dibangun dengan menggabungkan dan menggabungkan kembali satu set terbatas arsitektur primitif. Primitif seperti itu, biasanya disebut lapisan jaringan saraf, adalah blok bangunan dasar dari jaringan yang dalam. Jaringan yang terhubung sepenuhnya mengubah daftar input menjadi daftar output. Transformasi disebut terhubung penuh karena nilai input apapun dapat memengaruhi nilai output apapun. Lapisan ini akan memiliki banyak

parameter yang dapat dipelajari, bahkan untuk input yang relatif kecil.

Berikut ilustrasi arsitektur *deep learning* :



Gambar 4. Arsitektur *Deep Learning*

Sumber (Bharath Ramsundar, 2018)

D.2 Model Deep Learning

Deep learning memiliki banyak jenis model, dan setiap model jenis nya memiliki algoritma dan fungsi yang berbeda – beda. Jenis – jenis model *deep learning* dan kegunaannya ditunjukkan pada Tabel 1 berikut :

Tabel 3. Model *Deep Learning*

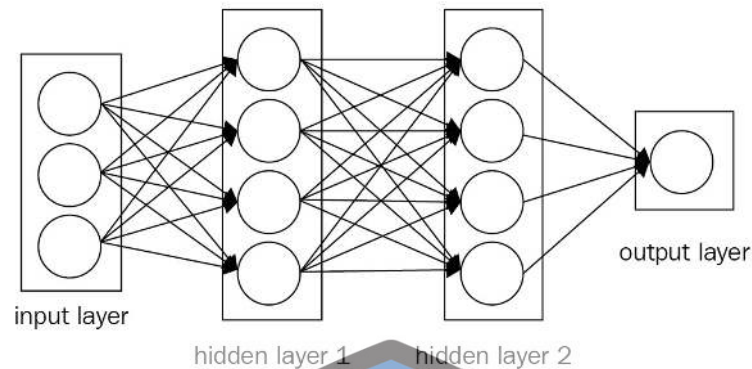
No	Model Deep Learning	Aplikasi dan Kegunaan
1.	ANN (<i>Artificial Nerual Network</i>)	Untuk regresi (Teknik untuk mengetahui data dimasa yang akan datang atau prediksi data dimasa depan) dan klasifikasi
2.	CNN (<i>Convulutional Nerual Network</i>)	Dapat digunakan untuk aplikasi <i>computer vision</i> , <i>face recognition</i> , <i>object detection</i> ,

		<i>image recognition, visual recognition</i> dimana CNN dapat mempelajari bagaimana mesin dengan algoritma bisa mengenali objek baik berupa gambar array video.
3.	RNN (<i>Recurrent Nerual Network</i>)	Dapat diaplikasikan untuk pengenalan suara (<i>voice/speech recognition</i>), analisis data kontinyu.
4.	RNTN (<i>Recursive Neural Tensor Network</i>)	Untuk pemrosesan teks (<i>text processing</i>) seperti analisis sentiment, penguraian dan pengenalan entitas (<i>entity recognition</i>)
5.	RBM (<i>Restricted Boltzmann Machine</i>)	Untuk mengekstral data yang tidak berlabel (<i>unlabelled</i>) dengan menggunakan pendetakan unsupervised serta secara otomatis dapat menemukan pola dalam data dengan merekonstruksi input
6	DBN (<i>Deep Belief Network</i>)	Untuk mempelajari himpunan fitur data tingkat tinggi dan lebih kompleks secara bertahap dari distribusi data.
7	<i>Autoencoders</i>	Digunakan untuk mereduksi dimensi fitur (<i>dimentionality reduction</i>). <i>Autoencoder</i> menerima data tidak berlabel; untuk dilakukan pengkodean dan selanjutnya merekonstruksi data seakurat mungkin

Dari **Tabel 3** diketahui bahwa model deep learning yang paling sesuai untuk pendeteksi masker wajah yaitu CNN (*Convolution Neural Network*). Pemilihan model ini sesuai dengan fungsi model CNN yaitu sebagai pengenalan wajah (*face recognition*) dan objek deteksi (*object detection*)

E. Convolution Neural Network

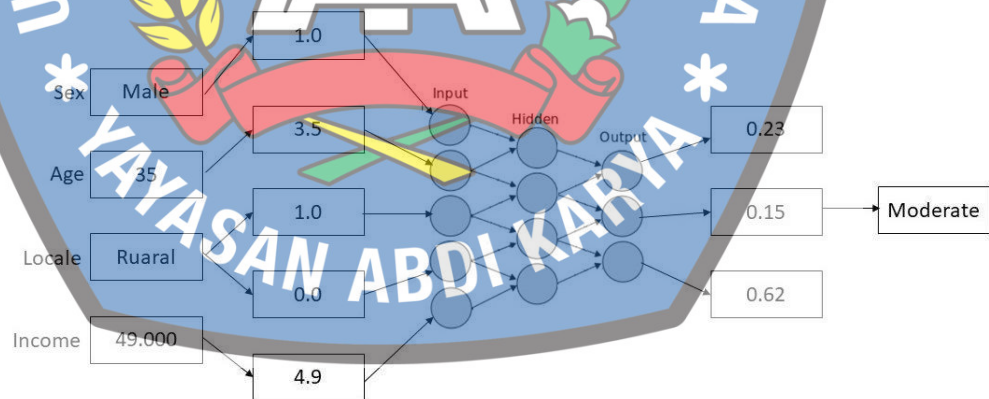
Convolution Neural Network atau ConvNets, sangat mirip dengan jaringan saraf biasa. Mereka masih terdiri dari neuron dengan bobot yang dapat dipelajari dari data. Setiap neuron menerima beberapa input dan melakukan produk titik. Mereka masih memiliki fungsi kerugian pada lapisan terakhir yang terhubung penuh. Mereka masih dapat menggunakan fungsi nonlinier. Data sebagai vektor tunggal dan melewati serangkaian lapisan tersembunyi. Setiap lapisan tersembunyi terdiri dari satu set *neuron*, di mana setiap neuron terhubung penuh ke semua neuron lain di lapisan sebelumnya. Dalam satu lapisan, setiap neuron benar-benar independen dan mereka tidak berbagi koneksi apa pun. Yang terakhir sepenuhnya lapisan terhubung, juga disebut lapisan keluaran, berisi skor kelas dalam kasus masalah klasifikasi gambar. Secara umum, ada tiga lapisan utama dalam ConvNet sederhana. Mereka adalah lapisan konvolusi, lapisan penyatuan, dan lapisan yang sepenuhnya terhubung (Mohik Sewak, 2018:61). Berikut gambar ilustrasi *Convolution neural network* (CNN) :



Gambar 5. CNN tiga layer

E.1 Neural Network

“*Neural network* merupakan salah satu metode yang menggunakan supervised learning untuk prediksi jika memberikan pasangan training input dan output.” (Widodo Budiharto, 2018:36) berikut contoh input dan output neural network :



Gambar 6. Contoh *Neural Network*

Sumber : Widodo Budiharto (2018) Gambar telah diolah Kembali

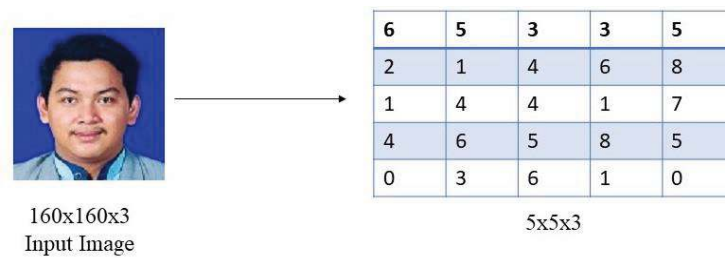
Tabel 4. Contoh Input dan Output Neural Network

Input (I0)			Bobot Tersembunyi(I1)				Output(I2)
0	0	1	0.1	0.2	0.5	0.2	0
0	1	1	0.2	0.6	0.7	0.1	1
1	0	1	0.3	0.2	0.3	0.9	1
1	1	1	0.2	0.1	0.3	0.8	0

Sumber : Widodo Budiharto (2018)

E.2 Convolution Layer

Dalam CNN, lapisan konvolusi pada dasarnya bertanggung jawab untuk menerapkannya atau lebih filter ke input. Inilah lapisan yang membedakan *convolutional neural network* dari jaringan saraf lainnya. Setiap lapisan konvolusi berisi satu atau lebih filter, yang dikenal sebagai kernel *convolutional*. Sebuah filter pada dasarnya adalah matriks bilangan bulat yang digunakan pada subset input gambar, yang berukuran sama dengan filter. Setiap piksel dari subset dikalikan dengan nilai yang sesuai di kernel, dan kemudian hasilnya dijumlahkan untuk satu nilai. Ini diulang sampai filter "slide" di seluruh gambar, sehingga membuat *feature map* keluaran. besarnya pergerakan antara aplikasi filter ke input gambar disebut sebagai langkah, dan hampir selalu simetris dalam tinggi dan lebar. (Pramod Singh 2020:77). Berikut merupakan sample perhitungan untuk proses lapisan konvolusi :



Gambar 7. Sample gambar data masukan

Gambar 7 merupakan gambar sampe *input image* yang mewakili dataset. Karena input image memiliki ukuran piksel 160 x 160, maka yang diambil Sebagian piksel saja atau bisa disebut matriks karena memiliki baris dan kolom yang akan dijadikan sample dalam proses konvolusi. Setiap piksel atau matriks memiliki nilai, nilai ini disebut sebagai nilai piksel. Dimana piksel memiliki rentang 0-255 dan sample nilai nya dapat dilihat pada matriks yang berukuran 5 x 5 x 3 pada gambar 2.4 proses ini agar dapat lebih mudah dalam memahami proses konvolusi pada *convolution layer* dan perhitungannya bisa lebih sederhana. Angka 3 diatas dimaksudnya bahwa gambar tersebut memiliki 3 channel warna yaitu, red, green, blue atau biasa disebut RGB. Untuk dapat hasil akhir dari proses konvolusi, maka digunakan rumus pada persamaan :

$$\text{Output} = \frac{W - F + 2P}{S} + 1 \dots\dots\dots$$

Dimana

W = Ukuran Gambar

$F = \text{Ukuran Kernal/filter}$

$P = \text{Padding}$

$S = \text{Stride}$

Untuk proses konvolusi dengan karnel dapat di lihat pada **Gambar 8**

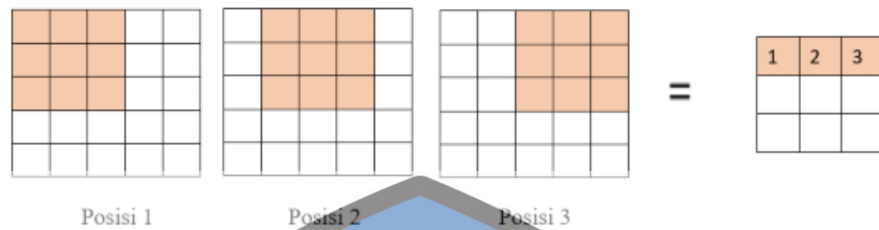


Sehingga hasil perhitungannya sebagai berikut :

$$\text{Output} = \frac{5 - 3 + 2(0)}{1} + 1$$

$$\text{Output} = 3$$

Dari perhitungan diatas maka output yang didapatkan untuk *feature map* dengan ukuran 3 x 3 adalah sebagai berikut :



Gambar 9. Sample pergeseran dengan nilai stride = 1 dan karnel 3 x 3



Gambar 11. Proses konvolusi pergeseran terakhir

Pada proses konvolusi untuk setiap pergeseran pada sample Gambar 9 dan 10 dilakukan melalui perkalian dot *product*. Perhitungannya adalah sebagai berikut :

Posisi 1 :

$$(6 \times 1) + (5 \times 4) + (3 \times 2) + (2 \times 2) + (1 \times 1) + (4 \times 0) + (1 \times 1) + (4 \times 2) + (4 \times 3) = 58$$

Posisi 2 :

$$(5 \times 1) + (3 \times 4) + (3 \times 2) + (1 \times 2) + (4 \times 1) + (6 \times 0) + (4 \times 1) + (4 \times 2) + (1 \times 3) = 43$$

Posisi 3 :

$$(3 \times 1) + (3 \times 4) + (5 \times 2) + (4 \times 2) + (6 \times 1) + (5 \times 0) + (4 \times 1) + (1 \times 2) + (7 \times 3) = 66$$

Posisi 4 :

$$(2 \times 1) + (1 \times 4) + (4 \times 2) + (1 \times 2) + (4 \times 1) + (4 \times 0) + (4 \times 1) + (6 \times 2) + (5 \times 3) = 51$$

Posisi 5 :

$$(1 \times 1) + (4 \times 4) + (6 \times 2) + (4 \times 2) + (4 \times 1) + (1 \times 0) + (6 \times 1) + (5 \times 2) + (8 \times 3) = 81$$

Posisi 6 :

$$(4 \times 1) + (6 \times 4) + (8 \times 2) + (4 \times 2) + (1 \times 1) + (7 \times 0) + (5 \times 1) + (8 \times 2) + (5 \times 3) = 89$$

Posisi 7 :

$$(1 \times 1) + (4 \times 4) + (4 \times 2) + (4 \times 2) + (6 \times 1) + (5 \times 0) + (0 \times 1) + (3 \times 2) + (6 \times 3) = 63$$

Posisi 8 :

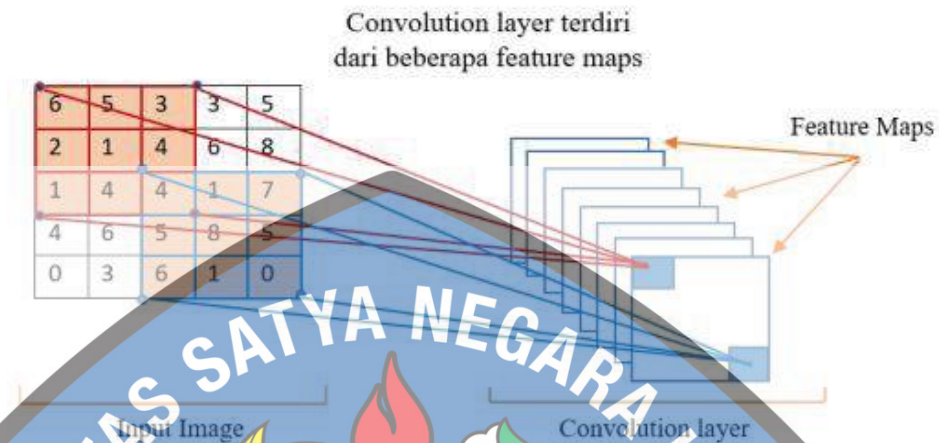
$$(4 \times 1) + (4 \times 4) + (1 \times 2) + (6 \times 2) + (5 \times 1) + (8 \times 0) + (3 \times 1) + (6 \times 2) + (1 \times 3) = 57$$

Posisi 9 :

$$(4 \times 1) + (1 \times 4) + (7 \times 2) + (5 \times 2) + (8 \times 1) + (5 \times 0) + (6 \times 0) + (1 \times 2) + (0 \times 3) = 48$$

Dapat dilihat bahwa gambar awal yang berukuran 5x5 berkurang menjadi lebih kecil yaitu 3 x 3. Hal ini merupakan salah satu tujuan utama kernal/feature detector . bisa dibayangkan jika megolah gambar yang berukuran besar dengan resolusi tinggi, maka bisa sampai ribuan piksel. Melalui kernal/feature detector proses pengolahan gambar menjadi semakin cepat karna data piksel yang diolah juga semakin kecil atau sedikit. Pada setiap proses CNN ada banyak kernal /feature detector. maka akan banyak

kumpulan dari feature map yang disebut dengan *convolution layer*. Ilustrasi ada pada **Gambar 11**.



Gambar 12. Ilustrasi *Convolution layer*

F. Tensorflow

Tensorflow merupakan machine learning engine milik google engine ini digunakan pada berbagai aplikasi Artificial Intelligence (AI) mereka seperti pengenalan suara, smart reply pada email, image searching, translator, dan sebagainya. Tensorflow memiliki license apache 2.0 license sehingga dapat bebas digunakan oleh semua orang. Teguh Wahyuno (2018:38)

Tensor adalah konstruksi matematika dasar di bidang-bidang seperti fisika dan teknik. Namun, secara historis, tensor telah membuat lebih sedikit terobosan dalam ilmu komputer, yang secara tradisional lebih dikaitkan dengan matematika dan logika diskrit. Keadaan ini telah mulai berubah secara signifikan dengan munculnya mesin pembelajaran dan fondasinya pada matematika vektor yang berkelanjutan.

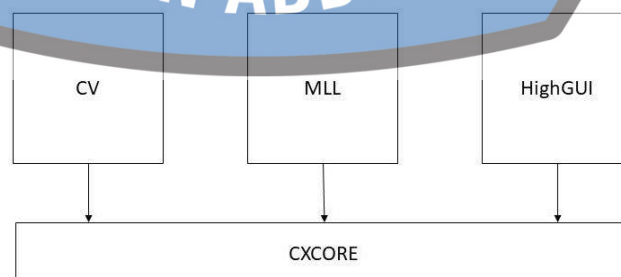
Mesin modern pembelajaran didasarkan pada manipulasi dan kalkulus tensor. Bharath Ramsundar (2018:19).

G. OpenCV (Open Source Computer Vision Library)

OpenCV (Open Source Computer Vision) merupakan perpustakaan fungsi pemrograman sumber terbuka, terutama digunakan untuk pemrosesan gambar waktu nyata. openCV dikembangkan oleh pusat penelitian Intel Rusia di Nizhny Novgorod dan sekarang didukung oleh Willow Garage dan Itseez. OpenCV sendiri sudah memiliki lebih dari 500 algoritma yang dioptimalkan untuk keperluan pemrosesan gambar dan video. OpenCV (Open Source Computer Vision) adalah perpustakaan sumber terbuka dengan tujuan pemrosesan gambar tertentu. Kuncinya adalah komputer memiliki kemampuan yang mirip dengan pemrosesan visual manusia. OpenCV sendiri sudah menyediakan banyak algoritma computer vision dasar. OpenCV juga menyediakan modul deteksi objek menggunakan algoritma Viola Jones.. (Prasetya & Nurviyanto, 2012). OpenCV menggunakan sebuah tipe face detector yang disebut Haarcascade Classifier. Face detector akan bekerja jika ada sebuah image yang berasal dari file gambar atau live video, face detector tersebut akan menguji tiap lokasi image dan akan mengklasifikasi file image atau live video tersebut sebagai wajah atau bukan wajah. Dalam mengklasifikasi wajah tersebut digunakan sebuah pemisalan skala yang tetap, contohnya 50×50 pixel. Jika sebuah wajah pada image lebih besar ataupun lebih kecil dari ukuran pixel yang sudah ditentukan, maka classifier akan terus menerus jalan beberapa kali untuk mencari wajah pada gambar tersebut.

Sedangkan menurut Aser Heber Ginting (2017) OpenCV (Open Computer Vision) adalah library API (application programming interface) yang sangat familiar dengan pengolahan citra computer vision. Computer vision sendiri merupakan cabang dari bidang pengolahan citra, yang memungkinkan komputer untuk melihat hal-hal seperti manusia. Dengan visi ini, komputer dapat membuat keputusan, tindakan, dan mengenali objek. Beberapa implementasi visi komputer adalah deteksi wajah, pengenalan wajah, pelacakan wajah/objek, pelacakan jalan, dll. OpenCV adalah perpustakaan open source visi komputer untuk C/C++. OpenCV dirancang untuk aplikasi waktu nyata. Ini memiliki fungsi pengambilan gambar atau video yang baik. OpenCV sendiri terdiri dari beberapa bagian, yaitu :

1. CV : Algoritma Image processing dan Vision.
2. ML : Machine learning library
3. Highgui : GUI, Image dan Video I/O.
4. CXCORE : Struktur data, support XML dan fungsi-fungsi grafis.



Gambar 13. Bagian-bagian pada OpenCV

H. Python

Python adalah bahasa pemrograman yang ditafsirkan, interaktif, berorientasi objek yang dapat berjalan di hampir semua platform, seperti keluarga UNIX, Mac, Windows, dll. Python merupakan salah satu bahasa pemrograman yang paling mudah dipelajari karena memiliki sintaks yang jelas dan elegan, dipadukan dengan modul yang out-of-the-box dan struktur data tingkat tinggi yang efisien. Bahasa pemrograman Python dibuat oleh Guido Van Rossum di Belanda pada tahun 1990, dan namanya berasal dari acara TV favorit Guido, Monty Python's Flying Circus. Jadi Python sebagai bahasa pemrograman banyak digunakan dalam industri dan pendidikan karena sintaksnya yang sederhana, ringkas, intuitif, dan library yang melimpah. Python juga merupakan pemrograman berorientasi objek (OOP). Data yang terkandung dalam Python adalah objek yang dibuat dari kelas. Pada pemrograman yang berorientasi pada objek merupakan suatu alat ampuh yang mampu untuk mengembangkan perangkat lunak yang dapat digunakan kembali.

H.1 Tipe Data

Tipe data adalah suatu media atau memory pada computer yang digunakan untuk menampung informasi. Python sendiri mempunyai tipe data yang cukup unik bila di bandingkan dengan Bahasa pemrograman lainnya. Dengan keunikan ini, pemrograman python menjadi lebih mudah dan dinamis.

Widodo Budiharto (2018:18) Berikut tipe data yang ada pada python :

Tabel 5. Tipe data python

No	Tipe Data	Penjelasan
1	Boolean	Menyatakan benar (<i>true</i>) yang bernilai 1 dan menyatakan salah (<i>false</i>) yang bernilai 0.
2	String	Menyatakan karakter atau kalimat yang bisa berupa huruf atau angka dan lain lain.
3	Float	Menyatakan bilangan yang memiliki koma.
4	Hexadecimal	Menyatakan bilangan dalam format hexa.
5	Complex	Menyatakan pasangan angka real dan imjiner.
6	List	Data untaian yang menyimpan berbagai tipe data yang isinya bisa di ubah – ubah.

Sumber : Widodo Budiharto (2018)

I.2 Operasi Dasar

Menurut Widodo (2018:19) ada beberapa operasi dasar penting seperti Boolean, perbandingan, dan aritmatika. Tipe Boolean sebagai berikut :

Tabel 6. Operasi dasar boolean

No	Operasi	Hasil
1	X or Y	If x is false, then y, else x

2	X and Y	If x is false, then x , else y
3	Not X	If x is false, then true, else false

Sumber : Widodo Budiharto (2018)

Sementara itu operasi perbandingan digunakan untuk membandingkan nilai yang di tunjukan pada **Tabel 7** berikut :

Tabel 7. Operasi Perbandingan

No	Operasi	Arti
1	<	Kurang dari
2	<=	Kurang dari sama dengan
3	>	Lebih besar dari
4	>=	Lebih besar dari sama dengan
5	=	Sama
6	!=	Tidak sama
7	Is	Identitas objek
8	Is Not	Identitas objek yang dinegasikan

Sumber : Widodo Budiharto (2018)

Untuk operasi matematika, gunakan symbol standar seperti berikut :

Tabel 8. Operasi matematika

No	Operasi	Hasil
1	$x + y$	Sum of x and y
2	$x - y$	Difference of x and y
3	$x * y$	Product of x and y
4	x / y	Quotient of x and y
5	$x // y$	(floored) quotient of x and y
6	$x \% y$	Reminder of x/y
7	$-x$	X negated
8	$+x$	X unchanged
9	$\text{abs}(x)$	Absolute value or magnitude of x
10	$\text{int}(x)$	X converted to integer
11	$\text{Long}(x)$	X converted to long integer
12	$\text{Complex}(re,im)$	A complex number with real part re, imaginary part im. Im default to zero
13	$\text{c.conjugate}(x)$	Conjugate of the complex number c, (identity on real number)
14	$\text{Divmod}(x,y)$	The pair $(x // y, x \% y)$

15	$\text{pow}(x, y)$	X the power y
16	$x ** y$	X the power y

Sumber : Widodo Budiharto (2018)

I. Webcam Eksternal

WebCam merupakan sejenis kamera yang dapat dipasang ke komputer atau laptop melalui port USB. WebCam memiliki banyak fungsi diantaranya yaitu untuk mengambil citra / gambar, mengambil suara / audio dengan mikropon serta dapat menyiarkan video secara real-time / live streaming. Mirip dengan kamera digital pada umumnya, webcam bekerja dengan menangkap cahaya melalui lensa berukuran kecil di bagian depan dengan bantuan detektor cahaya mikroskopik yang terpasang pada microchip penerima gambar yang umumnya berteknologi CMOS image sensor atau Charge-Couple Device (CCD). Gambar yang diambil oleh WebCam ditampilkan ke layar monitor, karena dikendalikan oleh komputer maka ada interface atau port yang digunakan untuk menghubungkan WebCam dengan komputer atau jaringan

WebCam berbeda dengan kamera digital, sebuah webcam tidak dilengkapi keping memori atau flash memory card. WebCam dirancang dengan tidak dapat menyimpan gambar karena fungsi utamanya hanya merekam dan mengirimkan gambar yang diperoleh secara langsung ke komputer / laptop. Itu sebabnya salah satu jenis webcam ada yang menggunakan kabel USB di bagian belakang. Kabel USB ini akan menyuplai listrik ke webcam dari komputer dan mengambil informasi

digital yang ditangkap oleh sensor webcam untuk diteruskan ke komputer . Contoh webcam ditunjukkan pada Gambar 2.14



Gambar 14. Webcam eksternal

BAB III

METODE PENELITIAN

A. Waktu Dan Tempat

A.1 Waktu Penelitian

Penelitian ini dimulai pada tanggal 01 Juni 2021 sampai dengan 15 Juli 2021.

A.2 Tempat Penelitian

Dalam penyusunan skripsi ini penulis melaksanakan studi kasus nya di Universitas Satya Negara Indonesia Kampus B yang beralamat di, Jl. H. Jampang No. 91, Jatimulya – Bekasi.

B. Analisa Kebutuhan Sistem

Untuk menunjang kebutuhan penelitian, penulis memerlukan software dan hardware sebagai berikut:

B.1 Perangkat Keras (*Hardware*)

Adapun perangkat keras atau hardware yang digunakan penulis dalam penelitian ini adalah sebagai berikut.:

Komputer/PC (Personal Computer) dengan spesifikasi:

1. *Processor: Core I5 4210U 1.70 Ghz.*
2. *Installed Memory (RAM): 8.00 Gb.*
3. *VGA Nvidia GeForce CUDA 3.0*
4. *System Type: 64-Bit Operation System.*

5. *Usb Webcam*

B.2 Perangkat Lunak (Software)

Adapun software atau perangkat lunak yang digunakan penulis dalam penelitian ini adalah sebagai berikut:

1. Anaconda
2. Python versi 3.9
3. Visual Studi Code
4. Tensorflow

C. Pengumpulan Data (*Data Gathering*)

Pada penelitian ini, pengumpulan data yang dilakukan dengan mengambil data gambar wajah tidak bermasker dan menggunakan masker pada situs online <https://www.kaggle.com/> yang diakses pada tanggal 26 April 2021. Dengan jumlah gambar wajah tidak bermasker sebanyak 5000 gambar dan gambar menggunakan masker sebanyak 5000 gambar. Dari data tersebut penulis membagi data gambar menjadi 3 subset untuk memudahkan penulis dalam melakukan *training data*, berikut subset yang digunakan dalam penelitian :

1. Data Training

Data training merupakan data yang digunakan untuk mentraining model dari *convolution neural network* untuk membuat prediksi atau menjalankan sebuah fungsi dari *convolutional neural network*

2. Data Testing

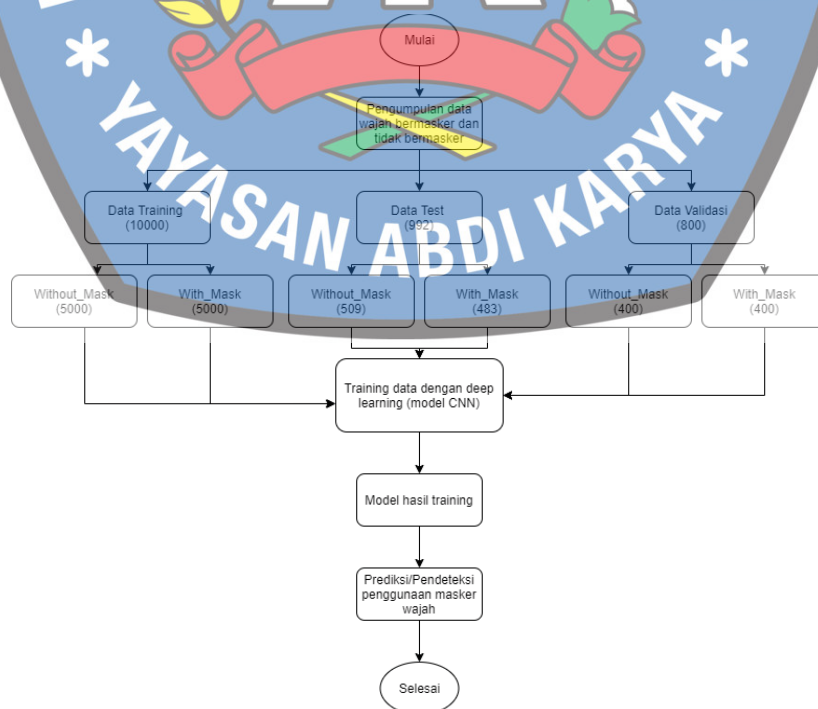
Data testing merupakan data yang digunakan untuk test model hasil training untuk memsimulasikan model dalam pada dunia nyata dan untuk mengetahui ke akuratan dan performa dari model hasil training.

3. Data Validation

Data validation merupakan data yang digunakan untuk memproses validasi dari model dan mencegah overfitting.

D. Prosedur Perancangan Sistem

Prosedur perancangan sistem merupakan gambaran sistem yang akan dibuat berdasarkan beberapa prosedur sistem menggunakan sebuah diagram alir (flowchart). Gambar diagram alir (flowchart) dari penelitian yang akan dilakukan, di tunjukan pada Gambar 15.



Gambar 15. Diagram Alir (Flowchart) Penelitian

Adapun tahapan perancangan sistem pada penelitian ini , yang di gambarkan melalui diagram alir(flowchart) pada gambar 15 diatas di uraikan melalui 3 tahapan berikut ini :

1. Tahap persiapan

a) Persiapan Hardware

- Laptop
- Usb Webcam

b) Persiapan software

- Python versi 3.9
- Jupyter Notebook
- Library tensorflow, OpenCV dan library pendukung lainnya.

c) Persiapan Data

- Download data citra sedang menggunakan masker dan tidak menggunakan masker.

2. Tahap Perancangan

a) Training Data

Training data merupakan proses yang dilakukan untuk melatih model arsitektur dari CNN yang didalamnya terdapat dataset citra

wajah dengan menggunakan masker atau tidak menggunakan masker, agar dapat memahami informasi pada data tersebut. Dan pada proses training data inilah yang akan dijadikan pola dan ekstraksi data oleh mesin atau computer berdasarkan arsitektur CNN yang telah dibuat, berikut pengujian penggunaan data training pada CNN :

Tabel 9. Parameter Model

No	Parameter	Value
1	Epoch	8
2	Target Size	128,128
3	Batch size	64
4	Learning Rate	0,001

1. *Epochs* adalah ketika seluruh dataset sudah melalui proses training pada Neural Netwok sampai dikembalikan ke awal untuk sekali putaran, karena satu Epoch terlalu besar untuk dimasukkan (feeding) kedalam komputer maka dari itu perlu dibagi kedalam satuan kecil (batches).
2. *Target Size* adalah target ukuran citra yang diinginkan memiliki ukuran berapa piksel dalam satu citra

3. *Batch Size* adalah jumlah sampel data yang disebar ke *Neural Network*. Contoh: jika kita mempunyai 100 dataset dan batch size kita adalah 5 maka algoritma ini akan menggunakan 5 sampel data pertama dari 100 data yang kita miliki (ke1, ke2, ke3, ke4, dan ke5) lalu disebar atau ditraining oleh *Neural Network* sampai selesai kemudian mengambil kembali 5 sampel data kedua dari 100 data (ke6, ke7, ke8, ke9, dan ke10), dan begitu seterusnya sampai 5 sampel data ke 20 ($100/5=20$).

4. *Learning Rate* parameter penyetelan dalam algoritme pengoptimalan yang menentukan ukuran langkah pada setiap iterasi sambil bergerak menuju minimum fungsi kerugian.

b) Model hasil *training*

Model hasil training ini merupakan model yang akan digunakan untuk prediksi ataupun pengujian sistem dengan program yang terpisah. Model ini berkentensi (.h5) dimana didalam model terdapat bobot, label kelas dan arsitektur *deep learning* model CNN secara keseluruhan.

3. Tahap Pengujian

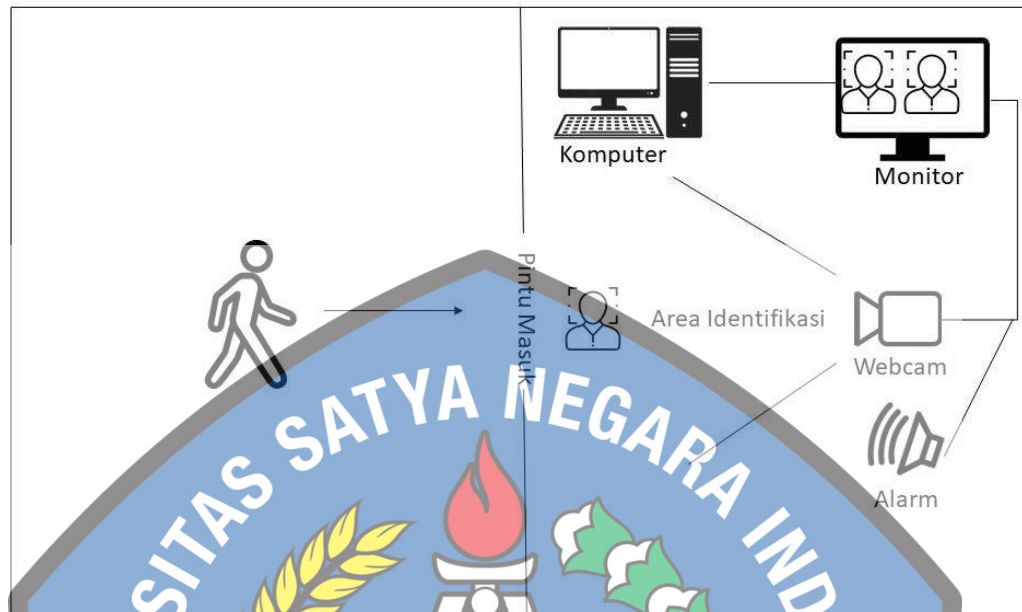
Tahap pengujian dilakukan dimana sistem yang telah dirancang dan menghasilkan model hasil training akan diuji kemampuannya dengan tujuan untuk mengetahui apakah sistem dapat bekerja atau berfungsi dengan baik sebagaimana yang diinginkan. Dalam hal ini , tahap

pengujian memiliki arti yang sama dengan tahap prediksi dan proses pendeteksi masker, berikut tahap pengujian pada penelitian ini :

- a) Pengujian kemampuan sistem dalam mengenali wajah
- b) Pengujian kemampuan sistem mendeteksi penggunaan masker dengan *single object*.
- c) Pengujian kemampuan sistem mendeteksi penggunaan masker dengan object lebih dari satu.
- d) Pengujian kemampuan sistem mendeteksi penggunaan masker berdasarkan jarak jangkauan terdekat sampai terjauh.
- e) Pengujian kemampuan sistem mendeteksi wajah dengan beberapa posisi (hadap kanan , hadap kiri , hadap atas dan hadap bawah)



E. Konsep Umum Sistem

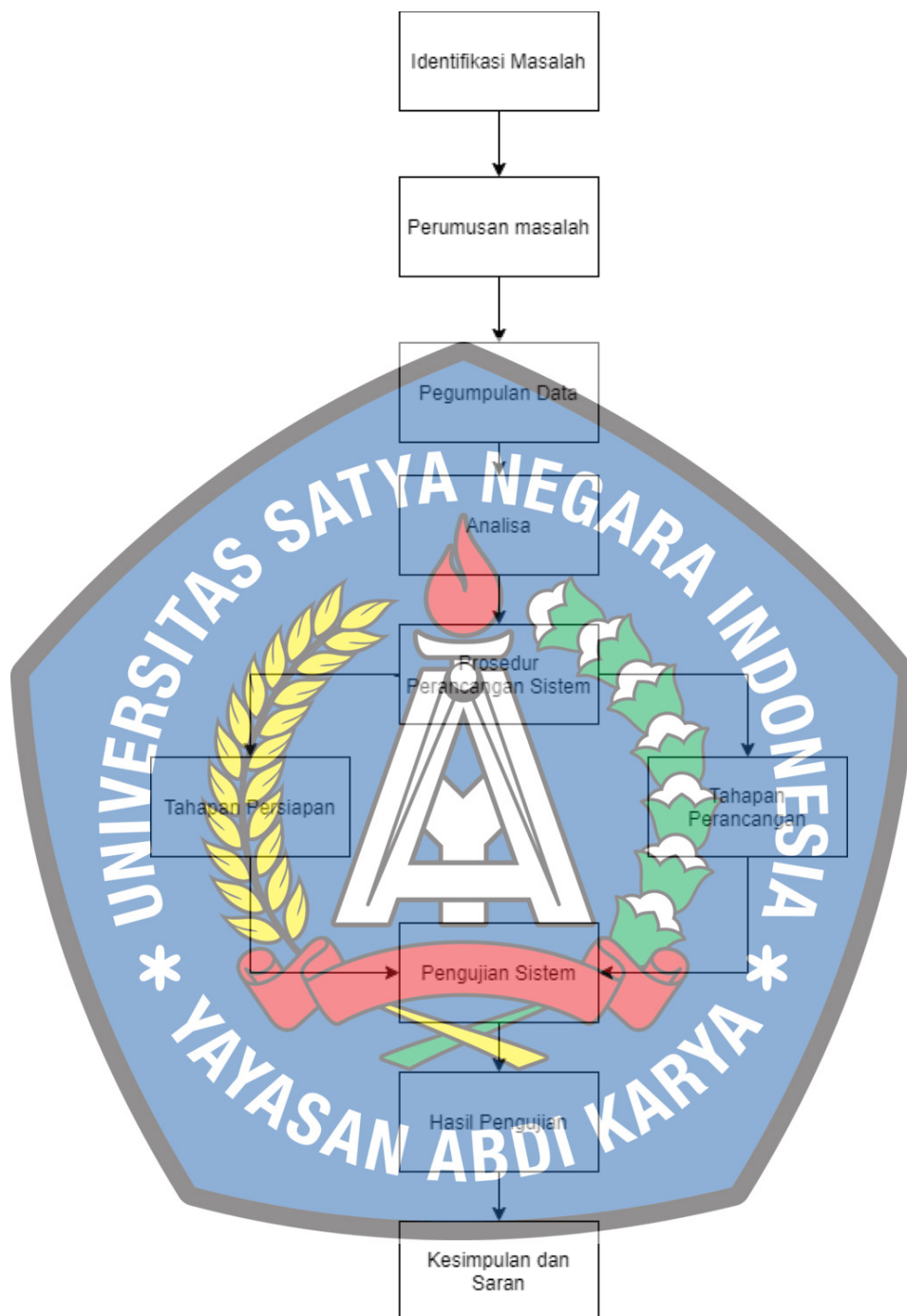


Gambar 16. Konsep Umum Sistem

Dalam konsep umum ini, alat diletakkan di pintu masuk yang di dalamnya terdapat 1 buah komputer untuk menjalankan aplikasi dan 1 unit monitor untuk menampilkan kondisi aplikasi saat sedang mendeteksi penggunaan masker, 1 Unit Webcam sebagai media input sebuah gambar ke aplikasi dan 1 unit alarm berupa speaker untuk notifikasi jika ada kondisi yang tidak sesuai.

F. Kerangka Berfikir

Merupakan suatu gambaran dari pembahasan yang akan dipecahkan sehingga mendapatkan suatu solusi. Dimana setiap alur dan tahapannya dibuat untuk membantu penulis memusatkan pada permasalahan yang diteliti. Adapun kerangka berfikir dalam penulisan ini adalah sebagai berikut:



Gambar 17. Kerangka berfikir

Sesuai dengan kerangka berfikir penelitian diatas, penelitian ini dilakukan dalam beberapa tahapan, yaitu sebagai berikut :

1. Identifikasi masalah yang ada pada tempat penelitian yaitu pada Universitas Satya Negara Indonesia Kampus B.
2. Merumuskan masalah yang ada dengan variable penelitian yang digunakan.
3. Pengumpulan data disini berupa data – data penelitian terdahulu , buku – buku yang berkaitan dengan penelitian dan pengumpulan dataset citra yang menggunakan masker dan tidak menggunakan masker.
4. Membuat prosedur perancangan sistem agar penelitian ini bisa lebih terarah dan tersusun, dalam prosedur perancangan sistem terdapat 2 bagian yaitu tahapan persiapan dan tahapan perancangan dimana tahapan persiapan merupakan tahapan peneliti mempersiapkan kebutuhan – kebutuhan yang menyerupai hardware atau software, dalam tahapan perancangan peneliti mempersiapkan apa saja yang akan dijalankan dan dirancang untuk menyelesaikan penelitian.
5. Pengujian sistem berfungsi untuk menguji hasil dari analisa yang ada untuk mengetahui sistem yang di buat bisa berjalan sesuai yang diinginkan oleh peneliti.
6. Hasil pengeujian merupakan langkah untuk mengetahui dimana letak kesalahan yang terjadi jika ada eror pada sistem.

BAB IV

IMPLEMENTASI DAN EVALUASI SISTEM

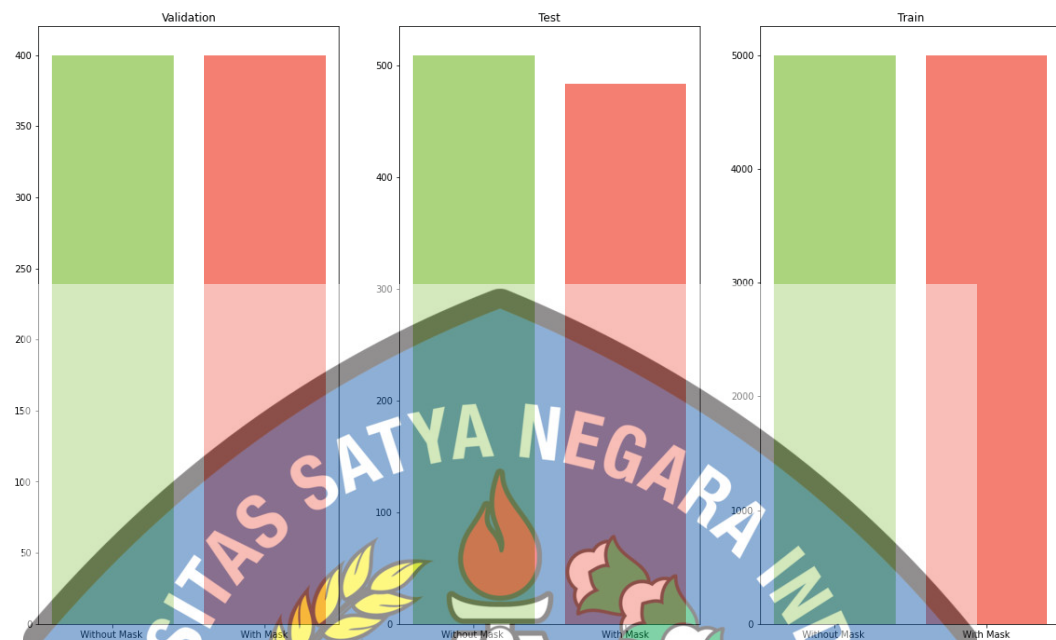
A. Umum

Pada penelitian ini, sistem yang dibuat merupakan sistem pengenalan wajah (face recognition) yang dikembangkan untuk dapat mendeteksi penggunaan masker wajah. Penelitian ini menggunakan metode pembelajaran *deep learning* dengan model *Convolutional Neural Network (CNN)* dan *Framework Tensorflow* sebagai *library* pendamping. Proses utama dalam pembuatan sistem ini diawali dengan pengumpulan data dan *training data* hingga menghasilkan sebuah model dan pengujian model hasil *training* dan pengujian dalam mendeteksi penggunaan masker wajah. Dalam bab ini, akan dibahas tentang pengujian sistem dengan beberapa kondisi dan pembahasan hasil *training dataset*. Pengujian ini dilakukan untuk mengetahui apakah sistem yang telah dirancang dapat bekerja dengan baik sebagaimana yang diinginkan, dari hasil pengujian, akan didapatkan data – data berupa tampilan keluaran sistem dan akurasi sistem.

B. Distribusi Gambar

Pada penelitian ini, penulis memiliki beberapa class atau kategori gambar yang mana diantaranya masing – masing memiliki class “with_mask” dan without_mask” dan dibedakan dalam masing – masing direktori yaitu *training*, *test* dan *validation*. Direktori ini dibuat untuk memudahkan dalam meload dataset yang akan di uji. Berikut distribusi gambar yang digunakan dalam pengujian dataset :

Image distribution



Gambar 18. Distribusi gambar

Dari gambar diatas bar diagram berwarna hijau merupakan dataset gambar yang tidak menggunakan masker dan bar diagram berwarna merah merupakan dataset gambar yang menggunakan masker

C. Augmentasi Gambar Sample

Peneliti melakukan beberapa augmentasi data untuk mendapatkan hasil yang lebih baik, beberapa parameter dipilih sebagai hasil dari standar yang ada dan beberapa dipilih dengan bantuan trial and error dan berpengaruh pada persentase akurasi. Berikut hasil augmemntasi gambar *sample* :

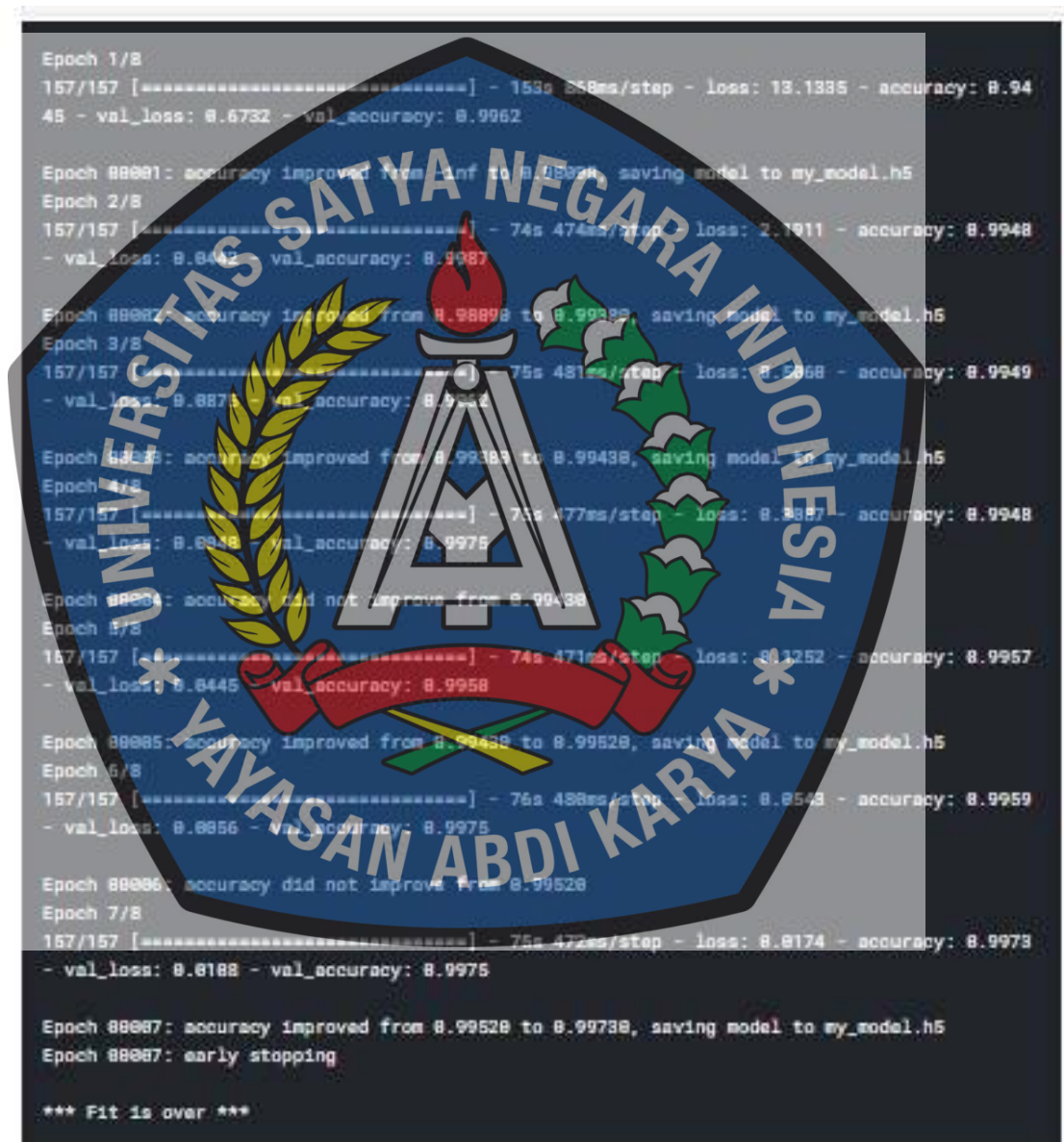


Gambar 19. Augmentasi citra *sample*



D. Pengujian Hasil Training Dataset

Dalam proses pengujian dataset peneliti menggunakan parameter berdasarkan **Tabel 7** Parameter Model [34] sehingga menghasilkan nilai *accuracy*, *loss* dan *validation* sebagai berikut :



```
Epoch 1/8
157/157 [#####] - 153s 258ms/step - loss: 13.1335 - accuracy: 0.9445 - val_loss: 0.6732 - val_accuracy: 0.9962

Epoch 88081: accuracy improved from inf to 0.9938, saving model to my_model.h5
Epoch 2/8
157/157 [#####] - 74s 474ms/step - loss: 2.7911 - accuracy: 0.9948 - val_loss: 0.8442 - val_accuracy: 0.9987

Epoch 88082: accuracy improved from 0.98898 to 0.9938, saving model to my_model.h5
Epoch 3/8
157/157 [#####] - 75s 481ms/step - loss: 0.8068 - accuracy: 0.9949 - val_loss: 0.8875 - val_accuracy: 0.9984

Epoch 88083: accuracy improved from 0.99383 to 0.99438, saving model to my_model.h5
Epoch 4/8
157/157 [#####] - 73s 477ms/step - loss: 0.8867 - accuracy: 0.9948 - val_loss: 0.8948 - val_accuracy: 0.9975

Epoch 88084: accuracy did not improve from 0.99438
Epoch 5/8
157/157 [#####] - 74s 471ms/step - loss: 0.8252 - accuracy: 0.9957 - val_loss: 0.8445 - val_accuracy: 0.9958

Epoch 88085: accuracy improved from 0.99438 to 0.99528, saving model to my_model.h5
Epoch 6/8
157/157 [#####] - 76s 488ms/step - loss: 0.8548 - accuracy: 0.9959 - val_loss: 0.8856 - val_accuracy: 0.9975

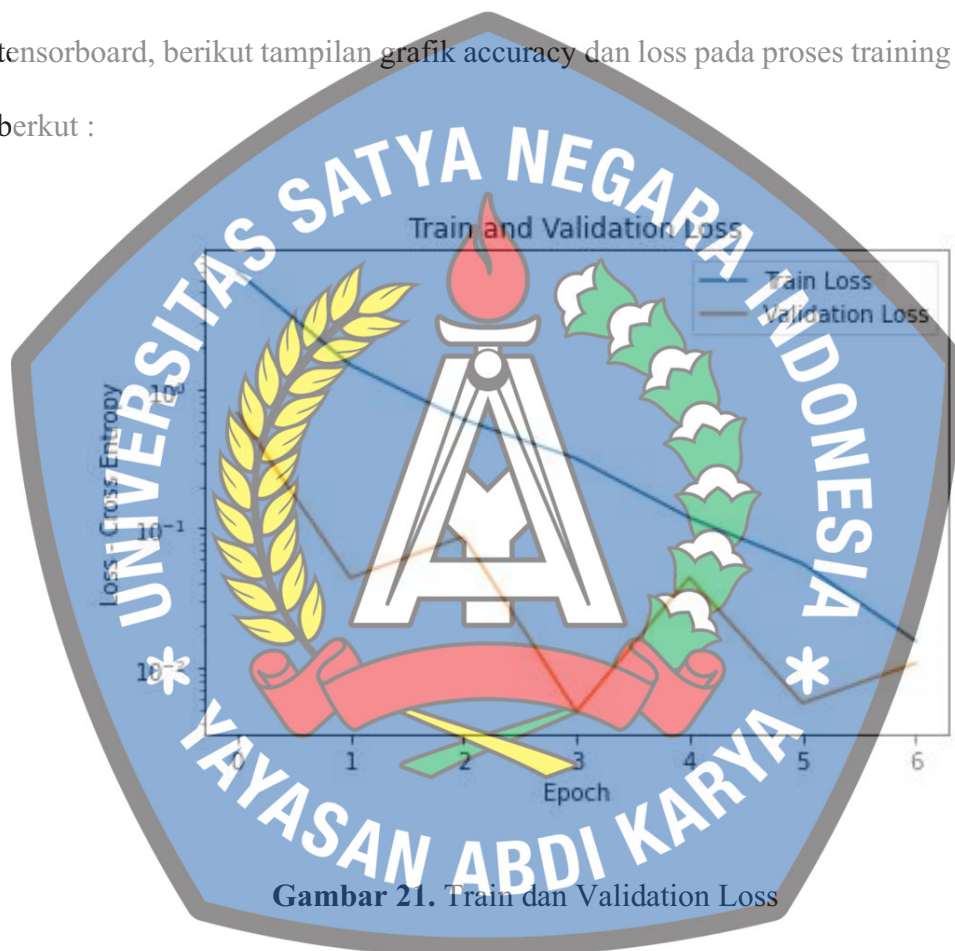
Epoch 88086: accuracy did not improve from 0.99528
Epoch 7/8
157/157 [#####] - 75s 472ms/step - loss: 0.8174 - accuracy: 0.9973 - val_loss: 0.8188 - val_accuracy: 0.9975

Epoch 88087: accuracy improved from 0.99528 to 0.99738, saving model to my_model.h5
Epoch 88087: early stopping

*** Fit is over ***
```

Gambar 20. Hasil training dataset

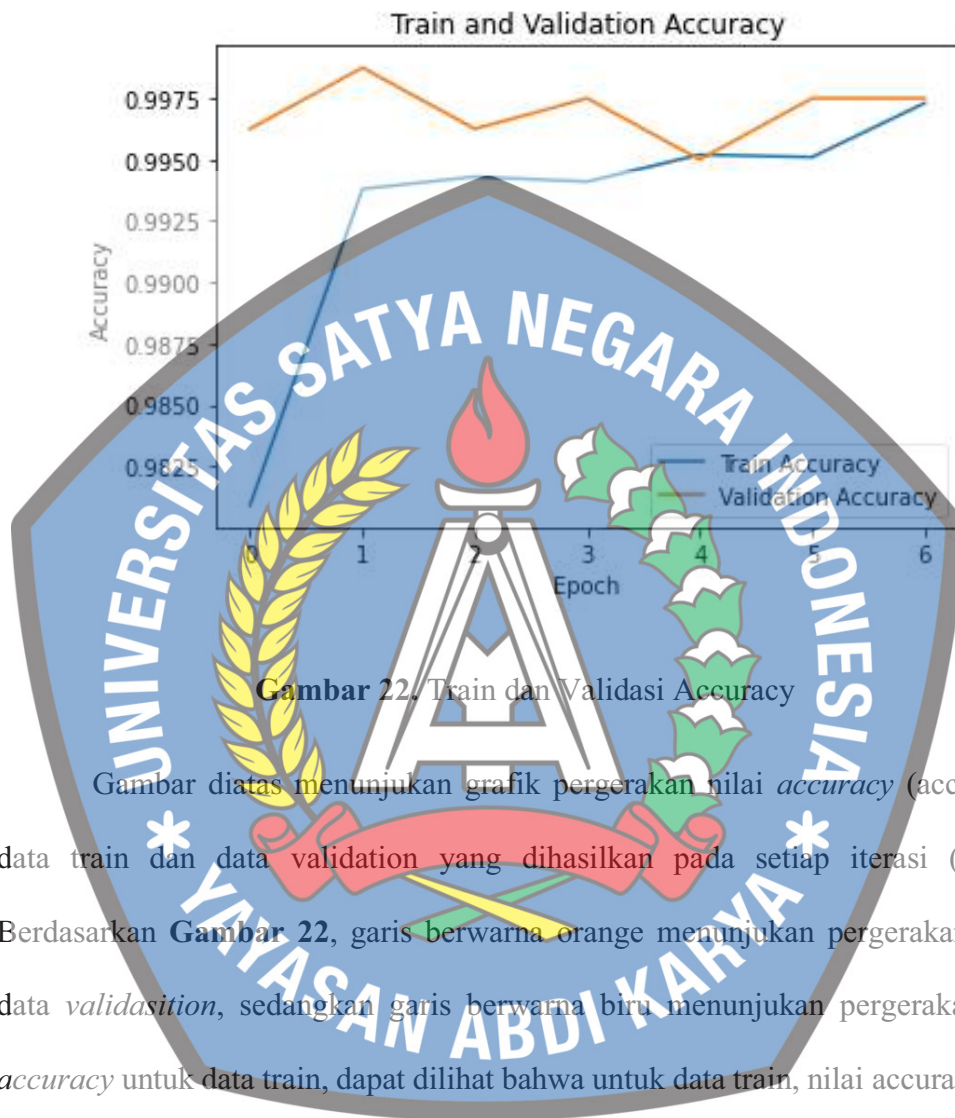
Dari hasil uji data training tersebut, didapati nilai *accuracy* tertinggi berada pada *epoch* 7 dengan nilai *accuracy* mencapai 99,5% serta nilai loss mencapai 0,017 % dan nilai *validation accuracy* sebesar 99,7% dan *validation loss* sebesar 0,016 setelah proses training dilakukan, tampilan grafik *accuracy*, *loss* dan *validation* bisa dilihat dengan membuat program plot model ataupun melalui tensorboard, berikut tampilan grafik *accuracy* dan *loss* pada proses training sebagai berikut :



Gambar 21. Train dan Validation Loss

Gambar diatas menunjukkan grafik pergerakan nilai (*loss*) untuk data train dan validasi yang dihasilkan pada setiap iterasi (*epoch*). Berdasarkan **Gambar 21** garis berwarna orange menunjukkan pergerakan untuk data *validation loss*, sedangkan garis berwarna biru menunjukkan pergerakan nilai *loss* untuk data train. Grafik di atas menunjukkan nilai *loss* untuk kedua data, dapat dilihat bahwa untuk data *train loss* dan *validation loss*, nilai yang didapatkan terus turun hingga epoch

ke-7. Menandakan model yang digunakan memiliki nilai eror yang cenderung rendah.



Gambar 22. Train dan Validasi Accuracy

Gambar diatas menunjukkan grafik pergerakan nilai *accuracy* (acc) untuk data train dan data *validation* yang dihasilkan pada setiap iterasi (*epoch*). Berdasarkan **Gambar 22**, garis berwarna orange menunjukkan pergerakan untuk data *validasition*, sedangkan garis berwarna biru menunjukkan pergerakan nilai *accuracy* untuk data train, dapat dilihat bahwa untuk data train, nilai *accuracy* yang didapatkan terus naik hingga epoch ke-7. Menandakan bahwa model yang digunakan memiliki nilai yang tinggi dalam memprediksi *with_mask* dan *without_mask*.

E. Uji Coba dan Evaluasi

Dalam melakukan uji coba dan evaluasi peneliti menggunakan *confusion matrix* seperti gambar dibawah ini :

		Predicted class		
		yes	no	Total
Actual class	yes	TP	FN	P
	no	FP	TN	N
Total		P'	N'	P + N

Gambar 23. *Confusion Matrix*

Pada gambar tersebut ditunjukkan, nilai *True Positive* didefinisikan sebagai *positive tuple* yang diklasifikasikan dengan benar oleh model. *True Negative* adalah *negative tuple* yang diklasifikasikan dengan benar oleh model. Sementara itu, *False Positive* adalah *negative tuple* yang diklasifikasikan sebagai kelas positif oleh model. *False Negative* adalah *positive tuple* yang diklasifikasikan sebagai kelas negatif oleh model klasifikasi. Berdasar *confusion matrix* pada gambar tersebut, kinerja model klasifikasi dapat dihitung,

Confusion matrix

True labels	Predicted labels	
	Without Mask	With Mask
Without Mask	482	1
With Mask	4	505

Gambar 24. *Confusion Matrix*

Dari **Gambar 24** diketahui bahwa nilai actual dengan data test sebanyak 506 gambar. Model memproses data test yang menghasilkan nilai TP (*True Positive*) dengan class without_mask sebanyak 482 gambar dengan nilai FP (*False Negative*) sebanyak 4 citra, untuk nilai FN (*False Negative*) sebanyak 1 gambar dan nilai TN (*true negative*) sebanyak 505 gambar.

Rumus perhitungan akurasi terhadap nilai *confusion matriks* adalah sebagai berikut :

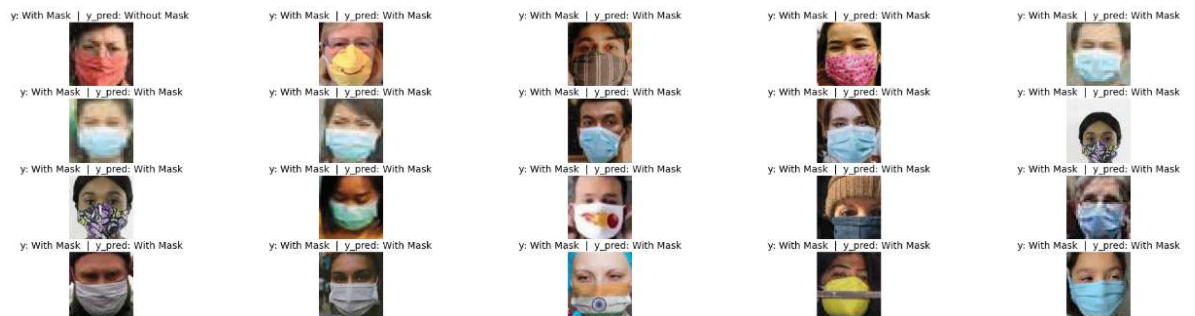
$$\text{Akurasi} = \frac{TP+TN}{TP+FP+TN+FN} \times 100\%$$

$$\text{Akurasi} = \frac{482+505}{482+4+505+1} \times 100\%$$

$$\text{Akurasi} = \frac{987}{995} \times 100\%$$

$$\text{Akurasi} = 99,1956 \text{ dibulatkan menjadi } 99,2\%$$

Dari hasil perhitungan akurasi dalam nilai *confusion matrix* berdasarkan *sample* data test didapatkan presentase sebanyak 99,2 %. Selanjutnya dilakukan prediksi model berdasarkan data test untuk memprediksi *sample* yang menggunakan masker, berikut hasil output dari prediksi model untuk data test sebagai berikut :






Gambar 25. Hasil prediksi *with_mask*

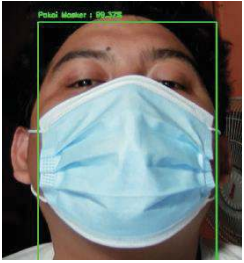
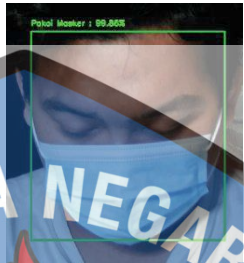
Dari gambar diatas, dapat dilihat bahwa prediksi dari model yang digunakan pada data test berhasil memprediksi gambar yang menggunakan masker dengan benar, sebanyak 20 sample data gambar yang ada pada data *test*.

F. Pengujian Kemampuan Sistem Mendeteksi Masker

Pengujian ini dilakukan dengan menggunakan sample secara *realtime* dan dengan posisi yang berbeda – beda, pengujian ini dilakukan untuk mengetahui sampai mana sistem yang dibangun dapat mendeteksi penggunaan masker dari beberapa kondisi.

Tabel 11. Pengujian sistem menggunakan masker





No	Kondisi	Sample	Hasil	
			Berhasil	Gagal
1	Posisi wajah tegak lurus		√	
2	Posisi wajah menghadap kekiri		√	
3	Posisi wajah menghadap kekanan		√	


4	Posisi wajah menghadap keatas		√	
5	Posisi wajah menghadap kebawah		√	

Pada tabel 11 dalam pengujian sistem dilakukan dalam kondisi menggunakan masker dengan 5 posisi sample yang diambil secara *realtime* dengan 5 kondisi diantaranya posisi menghadap tegak lurus, posisi menghadap kiri, posisi menghadap kanan, posisi menghadap ke atas, dan posisi menghadap kebawah. Pengujian ini berfungsi untuk mengetahui sistem yang berjalan dan membaca penggunaan masker dari berbagai posisi sistem berhasil mendeteksi penggunaan masker.

Tabel 12. Pengujian sistem tidak menggunakan masker

No	Kondisi	Sample	Hasil	
			Berhasil	Gagal

1	Posisi wajah tegak lurus		√	
2	Posisi wajah menghadap kekiri		√	
3	Posisi wajah menghadap kekanan		√	
4	Posisi wajah menghadap keatas		√	

5	Posisi wajah menghadap kebawah		√	
---	--------------------------------	--	---	--

Pada **tabel 12** dalam pengujian sistem dilakukan dalam kondisi tidak menggunakan masker dengan 5 posisi sample yang di ambil secara *realtime* dengan 5 kondisi diantaranya posisi menghadap tegak lurus, posisi menghadap kiri, posisi menghadap kanan, posisi menghadap ke atas, dan posis menghadap kebawah. Pengujian ini berfungsi untuk mengetahui sistem yang berjalan dapat mendeteksi pengguna masker yang tidak menggunakan berbagai posis sistem berhasil mendeteksi sample yang tidak memakai masker dengan beberapa posisi.



Pada pengujian selanjutnya sistem di uji dari beberapa gangguan dan jarak yang mana bertujuan untuk mengetahui apakah sistem dapat tetap mendeteksi seusai yang diinginkan dengan beberapa gangguan dan jarak. Pengujian sistem dilakukan dengan 5 gangguan dan 2 jarak yang di ambil secara *realtime* diantaranya

1. Wajah di tutupi dengan tangan.
2. Bagian wajah di tutupi benda selain masker.
3. Kondisi cahaya backlight dengan masker.
4. Kondisi cahaya backlight tidak menggunakan masker.
5. Posisi pembacaan dengan jarak deteksi sejauh 2 meter.

6. Posisi pembacaan dengan jarak deteksi sejauh 3 meter.
7. Sistem mendeteksi 2 object dalam sample
8. Sistem mendeteksi 3 object dalam sample

Berikut hasil dari pengujian gangguan dan jarak deteksi dapat dilihat pada **tabel 13** sebagai berikut :

Tabel 13. Pengujian sistem dengan gangguan eksternal

No	Kondisi	Sample	Hasil		Accuracy
			Berhasil	Tidak	
1	Wajah di tutupi dengan tangan				99%
2	Bagian wajah di tutupi benda selain masker			√	87%

3	Kondisi cahaya backlight dengan masker		✓		99%
4	Kondisi cahaya tidak menggunakan masker		✓		99%
5	Posisi pembacaan dengan jarak deteksi sejauh 2 meter		✓		99%
6	Posisi pembacaan dengan jarak deteksi sejauh 3 meter		✓		Tidak terdeteksi
7	Sistem mendeteksi 2 object dalam sample		✓		

8	Sistem mendeteksi 3 object dalam sample		√		
---	---	---	---	--	--

Pada **tabel 13** diketahui bahwa pada pengujian sistem dengan gangguan dan jarak deteksi adalah sebagai berikut :

1. Sample menutup wajah dengan tangan sistem berhasil mengenali sample yang tidak menggunakan masker melainkan menutupi dengan tangan.
2. Bagian wajah di tutupi dengan benda lain, sample tetap terdeteksi menggunakan masker dengan penurunan accuracy hingga 87%.
3. Pada pengujian dengan kondisi gangguan cahaya backlight dengan sample menggunakan masker, sistem berhasil mendeteksi penggunaan masker dengan accuracy sebanyak 99%.
4. Pada pengujian dengan kondisi gangguan cahaya backlight dengan sample menggunakan tidak menggunakan masker, sistem berhasil mendeteksi penggunaan masker dengan accuracy sebanyak 99%.
5. Pada pengujian dengan jarak deteksi mencapai 2 meter, sistem berhasil mendeteksi penggunaan masker dengan akurasi sebanyak 99%.
6. Pada pengujian dengan jarak deteksi mencapai 3 meter, sistem gagal mendeteksi penggunaan masker dan tidak bisa mendeteksi wajah pada jarak 3 meter.

7. Pada pengujian sistem mendeteksi 2 object secara bersamaan, sistem berhasil mendeteksi kedua object dengan menggunakan masker.
8. Pada pengujian sistem mendeteksi 3 object secara bersamaan, sistem berhasil mendeteksi ketiga object dimana sistem mampu membedakan secara bersamaan object yang tidak menggunakan masker dan menggunakan masker secara bersamaan.

Pada pengujian secara realtime dengan beberapa kondisi dapat diketahui *accuracy* yang di dapat dari hasil pengujian realtime menggunakan rumus confusion matrix, berikut hasil pengujian sistem secara realtime :

Tabel 14. Hasil kemampuan sistem secara *realtime*

No	Keterangan	Nilai
1	Total sample data pengujian	18
2	Berhasil Uji	16
3	Gagal Uji	2

Dari **tabel 14** kemudian dihitung menggunakan rumus confusion matrix sebagai berikut :

$$\text{Akurasi} = \frac{TP+TN}{TP+FP+TN+FP} \times 100\%$$

$$\text{Akurasi} = \frac{18+0}{18+2+0+0} \times 100\%$$

$$\text{Akurasi} = \frac{18}{20} \times 100\%$$

$$\text{Akurasi} = 90 \%$$

Dari hasil perhitungan confusion matrix didapati nilai akurasi pada pengujian kemampuan sistem mendeteksi masker sebesar 90% dari nilai keberhasilan sebesar 16 kali berhasil dan 2 kali gagal.

G. Pengujian Quality Of Service (QoS)

Pada pengujian Quality Of Service (QoS) berfungsi untuk mengetahui atau mendefinisikan kualitas dari sistem yang dibangun dengan beberapa parameter yang ada pada quality of service (QoS) untuk mengetahui sejauh mana kualitas sistem dan transmisi data yang diperoleh pada sistem yang dibangun, berikut hasil pengujian sistem berdasarkan metode QoS sebagai berikut :

Tabel 15. Parameter Delay (Latency)

No	Kondisi	Rata – Rata Delay (ms)	Keterangan	
			Indeks	Kategori
1.	Pakai Masker	199	3	Bagus
2.	Tidak Pakai Masker	200	3	Bagus

$$\text{Delay (Latency)} = \frac{\text{Total Delay}}{\text{Paket Data Diterima}}$$

$$\text{Delay (Latency)} = \frac{25597}{128} = 199 \text{ Pakai Masker}$$

$$\text{Delay (Latency)} = \frac{25601}{128} = 200 \text{ Tidak pakai masker}$$

Tabel 16. Parameter Paket Loss

No	Kondisi	Paket Yang Diterima	Paket Yang Dikirim	Rata – Rata Paket Loss(%)	Keterangan	
					Indeks	Kategori
1.	Pakai Masker	128	128	0%	4	Sangat Bagus
2.	Tidak Pakai Masker	128	128	0%	4	Sangat Bagus

$$Paket Loss = \frac{(Paket Data Dirim - Paket Data Diterima) \times 100 \%}{Paket Data Diterima}$$

$$Paket Loss = \frac{(128 - 128) \times 100 \%}{128} = 0\% \text{ Pakai masker}$$

$$Paket Loss = \frac{(128 - 128) \times 100 \%}{128} = 0\% \text{ Tidak pakai masker}$$

Tabel 17. Parameter Throughput

No	Kondisi	Rata – Rata throughput (bps)	Keterangan	
			Indeks	Kategori
1.	Pakai Masker	64	3	Bagus
2.	Tidak Pakai Masker	64	3	Bagus

$$Throughput = \frac{\text{paket diterima}}{\text{lama pengamatan}}$$

$$Throughput = \frac{128}{2} = 64$$

$$Throughput = \frac{128}{2} = 64$$

Tabel 18. Parameter Jifter

No	Kondisi	Jifter	Keterangan	
			Indeks	Kategori
1.	Pakai Masker	198	1	Jelek
2.	Tidak Pakai Masker	198	1	Jelek

$$\text{Throughput} = \frac{\text{Total variasi delay}}{\text{Total paket yang diterima}}$$

$$\text{Throughput} = \frac{25398}{128} = 198 \text{ Pakai Masker}$$

$$\text{Throughput} = \frac{25401}{128} = 198 \text{ Pakai Masker}$$



BAB V

KESIMPULAN DAN SARAN

A. KESIMPULAN

Implementasi *deep learning* menggunakan *framework tensorflow* dengan metode *convolution neural network* untuk mendeteksi masker wajah merupakan sistem yang di rancang untuk dapat mendeteksi penggunaan masker wajah dimasa pandemi saat ini, berbagai permasalahan yang muncul dalam penggunaan masker wajah telah diupayakan untuk dapat ditangani dengan sistem ini, Adapun kesimpulan yang dapat diambil adalah sebagai berikut :

1. Penggunaan *deep learning* dengan metode *convolution neural network* berhasil memprediksi gambar yang menggunakan masker wajah dengan nilai akurasi mencapai 99,7 %, nilai validation mencapai 99,7% dan nilai loss mencapai 0,01% sehingga model ini dapat digunakan untuk mendeteksi penggunaan masker wajah secara realtime.
2. Penggunaan Bahasa pemrograman python dan *framework tensorflow* telah berhasil mengimplementasikan model dari hasil training kedalam kode program untuk dapat mendeteksi penggunaan masker secara realtime
3. Pada pengujian kemampuan sistem mendeteksi masker wajah, sistem berhasil mendeteksi penggunaan masker dengan berbagai arah, kemudian sistem diuji dengan berbagai kondisi gangguan dimana sistem gagal mendeteksi masker jika object yang terdeteksi menutupi nya selain masker

wajah dengan nilai akurasi rendah 87% defiasi ideal 5% paling tinggi klo lebih dari atau sama dengan 10%.

4. Pada pengujian sistem dengan penggunaan jarak deteksi, sistem gagal mendeteksi masker wajah dengan jarak lebih dari 2 meter , sehingga jarak ideal yang digunakan pada sistem kurang dari 2 meter.

B. SARAN

Setelah melakukan Analisa terhadap sistem, skripsi ini masih memiliki beberapa kekurangan dan memungkinkan untuk dapat dikembangkan lebih lanjut. Oleh karnanya penulis merasa perlu memberikan saran – saran sebagai berikut :

1. Penggunaan perangkat hardware *graphic prosesing unit* (GPU) dengan spesifikasi lebih tinggi agar dapat mempercepat proses training data.
2. Untuk penelitian kedepan nya, bisa mencoba menggunakan jumlah layer (*lapisan*) CNN yang lebih banyak dengan mencoba kombinasi parameter – parameter lainnya.
3. Untuk penelitian selanjutnya, bisa menggunakan webcam usb atau IP Camera dengan resolution lebih tinggi dan kualitas gambar yang lebih baik agar proses deteksi masker bisa lebih baik.
4. Perlu adanya tambahan fitur pendeteksi suhu tubuh dan alat cuci tangan otomatis agar penerapan protokol Kesehatan bisa lebih lengkap.

5. Sebaiknya, pada proses mendeteksi penggunaan masker wajah camera atau webcam ditaruh ditempat cahaya yang cukup agar proses deteksi bisa lebih baik.



DAFTAR PUSTAKA

- Bharath Ramsundar, Reza Bosagh Zadeh. 2018. *TensorFlow for Deep Learning From Linear Regression to Reinforcement Learning*. United States Of America : O'Reilly Media, Inc, 2018.
- Putra, Darma. 2010. *Pengolahan Citra Digital*. Yogyakarta : ANDI, 2010
- Mohit Sewak, Md.Rezaul Karim, Pradeep Pujari. 2018. *Practical Convolutional Nerual Networks*. Mumbai : Packt Publishing Ltd, 2018.
- Pramod Singh, Avinash Manure. 2020. *Learn TensorFlow 2.0 : Implement Mechine Learning and Deep Learning*. India : Apress, 2020.
- Wahyuno,Teguh. 2018. *Fundamental Of Python For Mechine Learning*. Yogyakarta : GAVA MEDIA, 2018
- Silaparasetty, Vinita. 2020. *Deep Learning Project Using TensorFlow 2 : Neural Network Development with Python and Keras*. India :Apress, 2020.
- Budiharto, Widodo. 2018. *Pemrograman Python Untuk Ilmu Komputer Dan Teknik*. Yogyakarta : ANDI OFFSET, 2018.
- Pulung Adi Nugroho, Indah Fenriana, Rudi Arijanto M.Kom. 2020. Implementasi Deep Learning Menggunakan Convolutional Nerual Network (CNN) Pada Ekspresi Manusia. *Jurnal ALGOR*, 2-1.
- Dufan J.P. Mananjang, Sherwin R.U.A.Sompie, Agustinus Jacobus. 2020. Implementasi Framework Tensorflow Object Detection Dalam Mengklasifikasi Jenis Kendaraan Motor. *Jurnal Teknik Informatika*. 15-3 ,171-178.
- Hermawan, Egie. (2021). Klasifikasi Pengenalan Wajah Menggunakan Masker atau Atau Tidak Dengan Mengimplementasi Metode CNN (Convolutional Neural Network). *Jurnal Industri Kreatif dan Informatika Series (JIKIS)*. 01-01.
- Kurniadi Ahmad, Dkk. (2020). Implementasi Convolutional Neural Network Untuk Klasifikasi Varietas Pada Citra Daun Sawi Menggunakan Keras. *Journal Of Computer and Information Technology*. 4-1 , 25-33
- Rahim Arham, Dkk. (2020). Convolutional Neural Network Untuk Klasifikasi Penggunaan Masker. *Jurnal Teknologi Informasi dan Komunikasi*. 10-2, 109-115.

LAMPIRAN

1. Lembar persetujuan siding Skripsi

TGA-9



UNIVERSITAS SATYA NEGARA INDONESIA
FAKULTAS TEKNIK
Jalan Arjen Pondok Indah No. 11 Jakarta Selatan 12240
Telp (021) 7398393 (Hunting), Fax. (021) 7200352
Website <http://www.usni.ac.id>

LEMBAR PERSETUJUAN SIDANG SKRIPSI/TUGAS AKHIR

Setelah melalui proses bimbingan yang cukup dan pemeriksaan seluruh dokumen Skripsi/Tugas Akhir,
maka dengan ini kami menyatakan bahwa:

Nama	FAHRI HUSAINI
NIM	0111201503125085
Fakultas/Prodi	Teknik Informatika
Judul Skripsi	Implementasi Deep Learning Menggunakan Framework TensorFlow Dengan Model Convolutional Neural Networks Untuk Mendeteksi Marker Wajah

Disetujui untuk diuji secara komprehensif dalam Sidang Skripsi / Tugas Akhir, sesuai dengan jadwal yang
ditetapkan Panitia Sidang Skripsi/Tugas Akhir.

Demikian surat persetujuan ini dibuat untuk dapat digunakan sebagaimana mestinya.

Jakarta, 28 Juli 2021

Pembimbing I


(Hermalom Sitorus S.T., M.Kom)

Pembimbing II


(Wawan Kurniawan S.Kom., M.Kom)

2. Koding Latih Model

```
import numpy as np
import os
from PIL import Image

# Menampilkan Visualisasi dan evaluasi
import matplotlib.pyplot as plt
import seaborn as sns
from tensorflow.math import confusion_matrix

# Library Python dan tensorflow
import tensorflow as tf
from tensorflow.keras.preprocessing.image import
ImageDataGenerator, load_img, img_to_array
from tensorflow.keras import Model
from tensorflow.keras.layers import Flatten, Dense, Dropout
from tensorflow.keras.applications import DenseNet201
from tensorflow.keras import optimizers
from tensorflow.keras.losses import CategoricalCrossentropy
from tensorflow.keras.callbacks import EarlyStopping,
ModelCheckpoint

path = '../input/face-mask-12k-images-dataset/Face Mask Dataset/'
example_with_mask = path + '/Train/WithMask/1035.png'
example_without_mask = path + '/Train/WithoutMask/10.png'

# Variabel Parameter Latih
BATCH_SIZE = 64
EPOCHS = 8
TARGET_SIZE = (128, 128)
CLASSES = ['Without Mask', 'With Mask']

plt.imshow(load_img(example_with_mask))

plt.imshow(load_img(example_without_mask))

fig, axes = plt.subplots(1, 3, figsize=(20, 12))

for set_ in os.listdir(path):
    total = []
    ax = axes[os.listdir(path).index(set_)]
    for class_ in os.listdir(path+'/'+set_):
        count=len(os.listdir(path+'/'+set_+'/'+class_))
        total.append(count)
    ax.bar(CLASSES, total, color=['#a8e37e', '#fa8072'])
    ax.set_title(set_)
plt.suptitle('Image distribution', size=33)
```

In []:

In []:

In []:

In []:

```
plt.show()
```

In []:

```
train_datagen = ImageDataGenerator(rescale=1./255,
                                    rotation_range=10,
                                    width_shift_range=0.2,
                                    height_shift_range=0.2,
                                    zoom_range=0.25,
                                    horizontal_flip=True,
                                    samplewise_center=True,

samplewise_std_normalization=True, fill_mode='nearest')
test_datagen = ImageDataGenerator(rescale=1./255)

img = load_img(example_with_mask)
example_aug = img_to_array(img)/255.
#input memiliki 4 axis - menambahkan axis kosong untuk batch
example_aug = example_aug[np.newaxis]
plt.figure(figsize=(20,10))

for i,img in enumerate(train_datagen.flow(example_aug,
batch_size=1)):
    plt.subplot(4, 6, i+1)
    #remove empty axis
    plt.imshow(np.squeeze(img))

    if i == 23:
        break

plt.show()

train_set = train_datagen.flow_from_directory(directory=
path+'Train', batch_size=BATCH_SIZE, class_mode='categorical',
target_size=TARGET_SIZE)
validation_set = test_datagen.flow_from_directory(path +
'Validation',target_size=TARGET_SIZE)
```

In []:

```
def craete_model():
```

In []:

```
    denseNet_model = DenseNet201(input_shape=TARGET_SIZE + (3,),
weights='imagenet', include_top=False)
    denseNet_model.trainable = False

    flatten = Flatten()(denseNet_model.layers[-1].output)
    fc = Dense(units=512, activation='relu')(flatten)
    dropout = Dropout(0.35)(fc)
    output = Dense(2, activation='softmax')(dropout)

    model = Model(inputs=denseNet_model.input, outputs=output)

    model.summary()
```

```

return model

model = craete_model()

starter_learning_rate = 1e-2
end_learning_rate = 1e-6
decay_steps = 10000
learning_rate =
optimizers.schedules.PolynomialDecay(starter_learning_rate,decay_s
teps,end_learning_rate,power=0.4)

# Definisikan Optimizer, Loss & Metrics

opt = optimizers.Adam(learning_rate=learning_rate)
loss = CategoricalCrossentropy()
met = 'accuracy'

# untuk Compile Model
model.compile(optimizer=opt, loss=loss, metrics=[met])

my_callbacks = [
    EarlyStopping(monitor='val_accuracy',
min_delta=1e-5, patience=5,
mode='auto', restore_best_weights=False, verbose=1),
    ModelCheckpoint(filepath='my_model.h5',
monitor='accuracy', save_best_only=True, save_weights_only=False,
mode='auto', save_freq='epoch', verbose=1)
]

history = model.fit(train_set,
                    epochs=EPOCHS, steps_per_epoch=len(train_set),
# jumlah mini_batches yang ada didalam each epoch.
                    validation_data=validation_set,
                    callbacks=[my_callbacks],
                    verbose=1)

print('\n*** Fit is over ***')
model.save('my_model.h5')
#model.save_weights("my_model.h5")

train_loss = np.array(history.history['loss'])
val_loss = np.array(history.history['val_loss'])
plt.semilogy(train_loss, label='Train Loss')
plt.semilogy(val_loss, label='Validation Loss')

plt.legend(loc='upper right')

```

```
plt.xlabel('Epoch')
plt.ylabel('Loss - Cross Entropy')
plt.title('Train and Validation Loss')
```

```
plt.show()
```

In []:

```
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation
Accuracy')
plt.legend(loc='lower right')
plt.xlabel('Epoch'),
plt.ylabel('Accuracy')
plt.title('Train and Validation Accuracy')
plt.show()
```

In []:

```
test_set = test_datagen.flow_from_directory(path +
'Test',target_size=TARGET_SIZE,shuffle=False)
```

In []:

```
# Model Evaluate
loss, accuracy = model.evaluate(test_set)
print('Test Accuracy: ', '%033[1m',round(accuracy*100,
2),'%\033[0m')
```

In []:

```
# True Label & Predict of a particular Batch
image, label = test_set.next()
num_imgs = 20
lab_names = ['With Mask', 'Without Mask']
images = image[0:num_imgs,:,:, :]
labels = label[0:num_imgs,:]
predict = np.round(model.predict(images))

image_rows = 4
image_col = int(num_imgs/image_rows)

_, axs = plt.subplots(image_rows, image_col, figsize=(32,8))
axs = axs.flatten()
```

```
for i in range(num_imgs):
    img = images[i,:,:, :]
    lab = labels[i,:]
    axs[i].imshow(img)
    pred = predict[i]
    axs[i].axis('off')
    lab, pred = np.argmax(lab), np.argmax(pred)
    axs[i].set_title(label = f'y: {lab_names[lab]} | y_pred:
{lab_names[pred]}', fontsize=14)
```

```
plt.show()
```

In []:


```
y_pred = model.predict(test_set).argmax(axis=-1)
y_test = test_set.classes
```

```
Confusion_Matrix = confusion_matrix(y_test,y_pred)
```

In []:

```
fig, ax = plt.subplots(figsize=(25,10))
sns.heatmap(Confusion_Matrix,xticklabels=CLASSES,yticklabels=CLASS
ES, ax=ax, annot=True,fmt="1.0f",cbar=False,annot_kws={"size":
40})
ax.set_xlabel('Predicted labels');ax.set_ylabel('True labels');
plt.title("Confusion matrix",fontsize=30)
```

In []:

```
model.save('my_model.h5')
```

3. Koding Deteksi Masker Realtime

```
# USAGE
# python detect_mask_video.py
# import the necessary packages
from tensorflow.keras.applications.mobilenet_v2 import
preprocess_input
from tensorflow.keras.preprocessing.image import
img_to_array
from tensorflow.keras.models import load_model
from imutils.video import VideoStream
from playsound import playsound
import numpy as np
import argparse
import imutils
import time
import cv2
import os
import serial
import sys
```

```

def detect_and_predict_mask(frame, faceNet, maskNet):

    # ambil dimensi bingkai

    (h, w) = frame.shape[:2]

    blob = cv2.dnn.blobFromImage(frame, 1.0, (300, 300), (104.0,
177.0, 123.0))

        faceNet.setInput(blob)

        detections = faceNet.forward()

    # nisialisasi daftar wajah ke lokasi yang sesuai,
    # dan daftar prediksi dari jaringan masker wajah

        faces = []
        locs = []
        preds = []

    # loop detections
        for i in range(0, detections.shape[2]):

    # ekstrak confidence (yaitu, probabilitas) yang terkait
    dengan deteksi

            confidence = detections[0, 0, i, 2]

    # menyaring deteksi yang lemah dengan memastikan confidence
    lebih besar dari confidence minimum

            if confidence > args["confidence"]:

                try :

    # hitung (x, y)-koordinat kotak pembatas untuk object
    tersebut

            box = detections[0, 0, i, 3:7] * np.array([w, h, w,
h])(startX, startY, endX, endY) = box.astype("int")

    # pastikan kotak pembatas berada dalam dimensi frame

            (startX, startY) = (max(0, startX), max(0, startY))

            (endX, endY) = (min(w - 1, endX), min(h - 1, endY))

    # ekstrak ROI wajah, ubah dari saluran BGR ke RGB

    # resize it to 224x224, dan preprocess

            face = frame[startY:endY, startX:endX]

```

```

        face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)
        face = cv2.resize(face, (224, 224))
        face = img_to_array(face)
        face = preprocess_input(face)

# tambahkan wajah dan kotak pembatas ke daftar masing-masing
faces.append(face)

locs.append((startX, startY, endX, endY))

except Exception as e :
    print(e)

# hanya membuat prediksi jika setidaknya satu wajah
terdeteksi
if len(faces) > 0:
    # untuk inferensi yang lebih cepat, dibuat
    prediksi batch
    # prediksi satu per satu untuk loop
    faces = np.array(faces, dtype="float32")
    preds = maskNet.predict(faces, batch_size=32)

# kembalikan 2-tupel dari lokasi wajah dan yang sesuai
lokasi
return (locs, preds)

if __name__ == '__main__':

```

```

# membangun parser argumen dan mengurai argumen
ap = argparse.ArgumentParser()
ap.add_argument("-f", "--face", type=str,
                default="face_detector",
                help="path to face detector model directory")
ap.add_argument("-m", "--model", type=str,

```

```

        default="mask_detector.model",

        help="path to trained face mask detector model")

    ap.add_argument("-c", "--confidence", type=float,
                    default=0.5,

                    help="minimum probability to filter weak
                    detections")

    args = vars(ap.parse_args())

    # load model CNN dari File disk
    print("[INFO] loading face detector model...")
    prototxtPath = os.path.sep.join([args["face"],
    "deploy.prototxt"])
    weightsPath = os.path.sep.join([args["face"],
    "res10_300x300_ssd_iter_140000.caffemodel"])
    faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

    # load model CNN dari File disk
    print("[INFO] loading face mask detector model...")
    maskNet = load_model(args["model"])

    # inisialisasi untuk menampilkan video
    print("[INFO] starting video stream...")

    # ganti link video stream dari cctv / dari webcam
    lainnya di bawah ini

    # isi dengan 0 jika pakai default punya laptop
    vs = VideoStream(src=0).start()

    time.sleep(2.0)

    while True:

```



```

frame = vs.read()

frame = imutils.resize(frame, width=720)

(locs, preds) = detect_and_predict_mask(frame, faceNet,
maskNet)

for (box, pred) in zip(locs, preds):

# unpack the bounding box and predictions

(startX, startY, endX, endY) = box

(mask, withoutMask) = pred

# Menambahkan Text dan warna kedalam border

label = "Pakai Masker " if mask > withoutMask else "Tidak
Pakai Masker "color = (0, 255, 0) if label == "Pakai Masker
" else (0, 0, 255)c = None

# include the probability in the label

label = "{}: {:.2f}%".format(label, max(mask, withoutMask) *
100)

# display the label and bounding box rectangle on the output
frame

cv2.putText(frame, label, (startX, startY - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)

cv2.rectangle(frame, (startX, startY), (endX, endY), color,
2)

#untuk menampilkan alarm

#if mask < withoutMask:

#path = os.path.abspath("1.wav")

#playsound(path)

# show the output frame

cv2.imshow("Frame", frame)

key = cv2.waitKey(1) & 0xFF

# tekan Q untuk Close

if key == ord("q"):

```

```
break
```

```
# untuk membersihkan jendela kerja
```

```
cv2.destroyAllWindows()
```

```
vs.stop()
```



4. Kartu Bimbingan Skripsi Pembimbing I

Form -4



UNIVERSITAS SATYA NEGARA INDONESIA
FAKULTAS TEKNIK
 Jalan Arteri Pondok Indah No. 11 Jakarta Selatan 12240
 Telp (021) 7398393 (Hunting), Fax. (021) 7200352
 Website <http://www.usni.ac.id>

KARTU BIMBINGAN SKRIPSI/TUGAS AKHIR

FAKULTAS TEKNIK

Nama : Fahri Husaini
 No. Mhs : 011701503125085
 Prodi : Teknik Informatika
 Dosen Pembimbing I : Hernalom Sitorus S.T., M.Kom
 Dosen Pembimbing II : Wawan Kurniawan S.Kom., M.Kom
 Judul : Implementasi Deep Learning Menggunakan Framework Tensorflow Dengan Metode Convolutional Neural Network Untuk Mendeteksi Masker Wajah

No	Tanggal	Catatan Pembimbing I	Pid. dosen pembimbing
1	09/03/21	Pembahasan latar belakang	JPM
2	16/03/21	Pembahasan metode	JPM
3	26/03/21	Pembahasan Tinjauan pustaka	JPM
4	09/04/21	Pembahasan metode penelitian	JPM
5	26/04/21	Revisi hasil seminar pro posek: - latar belakang - tinjauan pustaka - metode penelitian	JPM
6	30/04/21	Pembahasan Hubungan	JPM

Form -4



UNIVERSITAS SATYA NEGARA INDONESIA
FAKULTAS TEKNIK

Jalan Arteri Pondok Indah No. 11 Jakarta Selatan 12240
Telp (021) 7398393 (Hunting), Fax. (021) 7200352
Website <http://www.usni.ac.id>

KARTU BIMBINGAN SKRIPSI/TUGAS AKHIR

FAKULTAS TEKNIK

Nama : Fahri Husaini
No. Mhs : 011701503125085
Prodi : Teknik Informatika
Dosen Pembimbing I : Hermalom Sitorus S.T., M.Kom
Dosen Pembimbing II : Wawan Kurniawan S.Kom., M.Kom
Judul :

Implementasi Deep Learning Menggunakan Framework Tensorflow Dengan Metode
Convolutional Neural Network Untuk Mendeteksi Masker Wajah

No	Tanggal	Catatan Pembimbing I	Ttd dosen pembimbing.
		Varabel Judul	
7	06/05/21	Revisi sistem	
8	17/05/21	Pembahasan latar belakang dan kaitannya	
9	11/06/21	Uji coba sistem	
10	21/06/21	Pembahasan BAB IV	
11	02/07/21	Pembahasan hasil pengujian sistem	
12	15/07/21	Pembahasan kesimpulan dan saran	
	26/07/21	ACC Sidang	

5. Kartu Bimbingan Skripsi Pembimbing II

Form -5



UNIVERSITAS SATYA NEGARA INDONESIA
FAKULTAS TEKNIK
 Jalan Arteri Pondok Indah No. 11 Jakarta Selatan 12240
 Telp (021) 7398393 (Hunting), Fax. (021) 7200352
 Website <http://www.usni.ac.id>

KARTU BIMBINGAN SKRIPSI/TUGAS AKHIR
FAKULTAS TEKNIK

Nama : Fahri Husaini
 No. Mhs : 011701503125085
 Prodi : Teknik Informatika
 Dosen Pembimbing I : Hernaloni Sitorus S.T., M.Kom
 Dosen Pembimbing II : Wawan Kurniawan S.T., M.Kom
 Judul :
 Implementasi Deep Learning Menggunakan Framework Tensorflow Dengan Metode
 Convolutional Neural Network Untuk Mendeteksi Masker Wajah

No	Tanggal	Catatan Pembimbing II	Ttd dosen pembimbing.
1		Menyusun Format, format heading dan layout	
2		Revisi Format, tabel dan gambar	
3		Koreksi BAB I s/d III	
4		Revisi jarak tulisan	
5		Koreksi BAB III dan IV	
6		letak Daftar Pustaka	
7		Revisi BAB II	
8		acc sedang	

6. Surat Ijin Riset



UNIVERSITAS SATYA NEGARA INDONESIA
FAKULTAS TEKNIK
Jolon Arteri Pondok Indah No. 11, Jakarta Selatan 12240
Telp. (021) 739 8393 (Hunting). Fax. (021) 720 0352
website <http://www.usni.ac.id>

Nomor : 329 /D-TEK/VII/2021
Lampiran : -
Perihal : Riset

26 Juli 2021

Kepada
Yth. : Bapak Pimpinan
Universitas Satya Negara Indonesia Kampus B
JL.H jampang No.91 , Jatimulya Keb. tambun selatan , Kab.Bekasi , Jawa barat 17510

Bersama ini kami dari Fakultas Teknik (FT) Universitas Satya Negara Indonesia (USNI) memberitahukan dan memohon dengan hormat hal-hal sebagai berikut:

1. Dalam rangka mengakhiri Pendidikan Program Strata-1 (S-1) Program Studi Teknik Informatika (S-1) Fakultas Teknik Universitas Satya Negara Indonesia, salah satu syaratnya adalah Mahasiswa/i diwajibkan menulis skripsi.
2. Sehubungan dengan butir 1 (satu) di atas salah seorang Mahasiswa/i Fakultas Teknik Program Studi Teknik Informatika (S-1)USNI, yaitu :

Nama	: FAHRI HUSAINI
NIM	: 011701608125085
Fakultas/Prodi	: Teknik/ Teknik Informatika (S-1)
No Tlp	: 087383665559
3. Memilih instansi yang Bapak pimpin sebagai objek penelitian untuk keperluan menulis skripsi.
3. Hasil penelitian yang dilakukan Mahasiswa/i tersebut hanya untuk keperluan ilmiah semata dan tidak akan disebarluaskan.

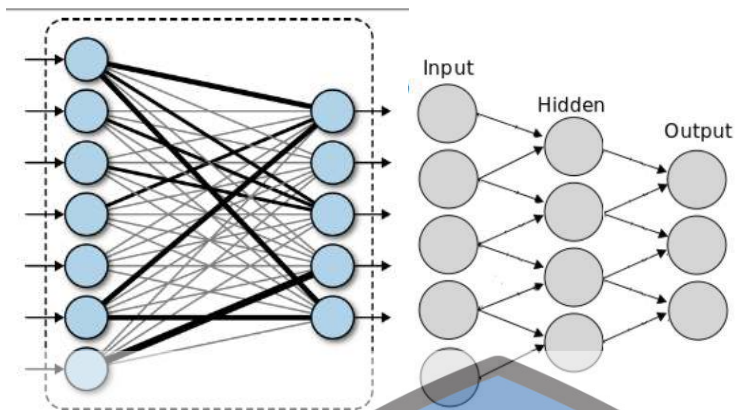
Demikian permohonan ini kami ajukan, atas kesediaannya kami mengucapkan terima kasih.


Ir. Nurhayati, M.Si.

Cc.Arsip

6. Lampiran Gambar

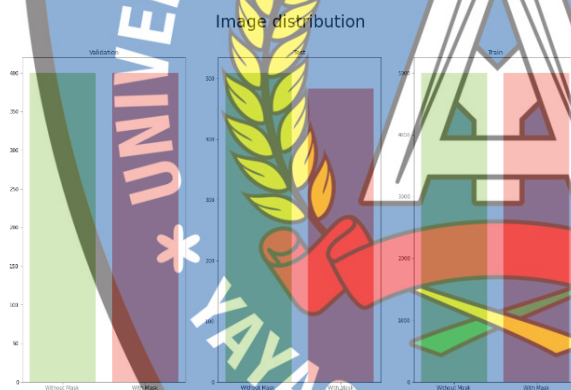


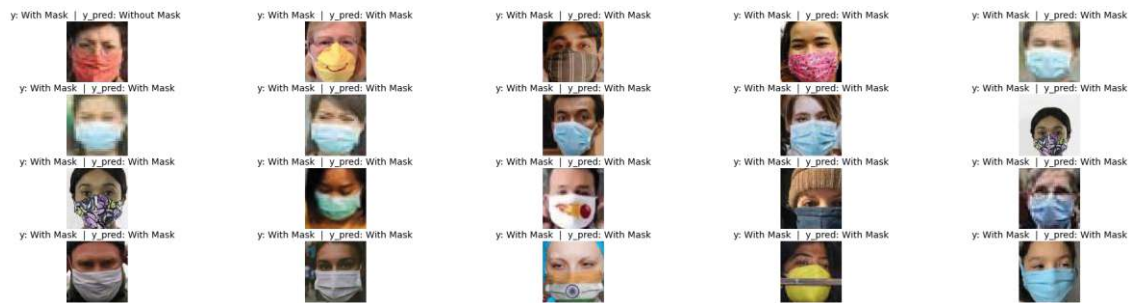


160x160x3
Input Image

6	5	3	3	5
2	1	4	6	8
1	4	4	1	7
4	6	5	8	5
0	3	6	1	0

5x5x3





Confusion matrix

	With Mask	Without Mask
With Mask	482	1
Without Mask	4	505

```

Epoch 1/8
157/157 [#####] - 153s 86ms/step - loss: 13.1335 - accuracy: 0.9445 - val_loss: 0.6732 - val_accuracy: 0.9962

Epoch 00001: accuracy improved from -inf to 0.98998, saving model to my_model.h5
Epoch 2/8
157/157 [#####] - 74s 47ms/step - loss: 1.1911 - accuracy: 0.9948 - val_loss: 0.6442 - val_accuracy: 0.9987

Epoch 00002: accuracy improved from 0.98898 to 0.99338, saving model to my_model.h5
Epoch 3/8
157/157 [#####] - 75s 48ms/step - loss: 0.5868 - accuracy: 0.9949 - val_loss: 0.6735 - val_accuracy: 0.9962

Epoch 00003: accuracy improved from 0.99388 to 0.99438, saving model to my_model.h5
Epoch 4/8
157/157 [#####] - 75s 47ms/step - loss: 0.3887 - accuracy: 0.9948 - val_loss: 0.6848 - val_accuracy: 0.9975

Epoch 00004: accuracy did not improve from 0.99438
Epoch 5/8
157/157 [#####] - 74s 47ms/step - loss: 0.1252 - accuracy: 0.9957 - val_loss: 0.6445 - val_accuracy: 0.9958

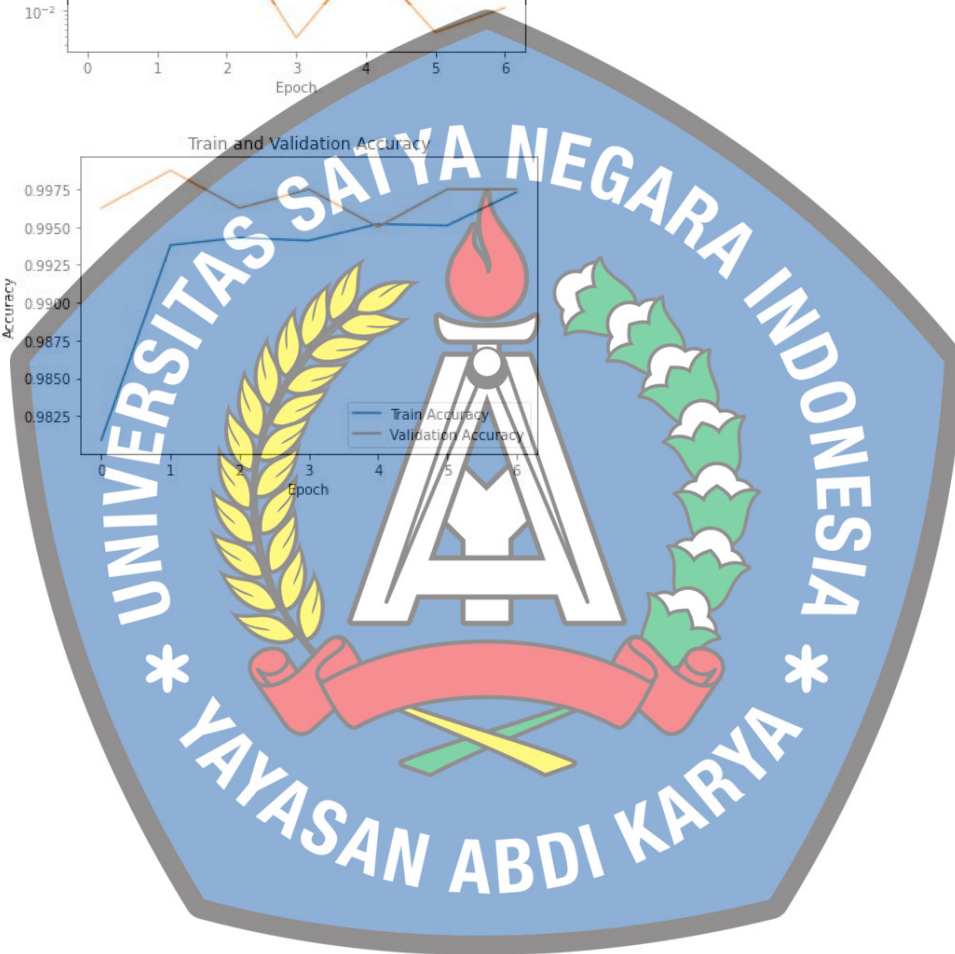
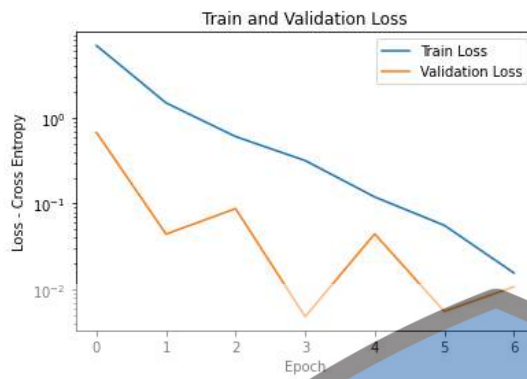
Epoch 00005: accuracy improved from 0.99438 to 0.99528, saving model to my_model.h5
Epoch 6/8
157/157 [#####] - 75s 48ms/step - loss: 0.8543 - accuracy: 0.9959 - val_loss: 0.6856 - val_accuracy: 0.9975

Epoch 00006: accuracy did not improve from 0.99528
Epoch 7/8
157/157 [#####] - 75s 47ms/step - loss: 0.8174 - accuracy: 0.9973 - val_loss: 0.6188 - val_accuracy: 0.9975

Epoch 00007: accuracy improved from 0.99528 to 0.99738, saving model to my_model.h5
Epoch 00007: early stopping

*** Fit is over ***

```



7. Lampiran Tabel

Tabel 1. Contoh Citra ukuran berwarna

(i,j)	0	1	2
0	31	42	46
	33	44	48
	22	33	37
1	42	44	64
	44	46	66
	33	35	55
2	188	186	193
	189	187	194
	181	179	186

Tabel 2. Contoh citra hasil konversi grayscale

(i,j)	0	1	2
0	29	40	44
1	40	42	62
2	188	184	191

Tabel 3. Model Deep Learning

No	Model Deep Learning	Aplikasi dan Kegunaan
1.	ANN (<i>Artificial Nerual Network</i>)	Untuk regresi (Teknik untuk mengetahui data dimasa yang akan datang atau prediksi data dimasa depan) dan klasifikasi



2.	CNN (<i>Convolutional Neural Network</i>)	Dapat digunakan untuk aplikasi <i>computer vision</i> , <i>face recognition</i> , <i>object detection</i> , <i>image recognition</i> , <i>visual recognition</i> dimana CNN dapat mempelajari bagaimana mesin dengan algoritma bisa mengenali objek baik berupa gambar array video.
3.	RNN (<i>Recurrent Neural Network</i>)	Dapat diaplikasikan untuk pengenalan suara (<i>voice/speech recognition</i>), analisis data kontinyu.
4.	RNTN (<i>Recursive Neural Tensor Network</i>)	Untuk pemrosesan teks (<i>text processing</i>) seperti analisis sentiment, penguraian dan pengenalan entitas (<i>entity recognition</i>)
5.	RBM (<i>Restricted Boltzmann Machine</i>)	Untuk mengekstral data yang tidak berlabel (<i>unlabelled</i>) dengan menggunakan pendetaksian unsupervised serta secara otomatis dapat menemukan pola dalam data dengan merekonstruksi input
6.	DBN (<i>Deep Belief Network</i>)	Untuk mempelajari himpunan fitur data tingkat tinggi dan lebih kompleks secara bertahap dari distribusi data.
7.	<i>Autoencoders</i>	Digunakan untuk mereduksi dimensi fitur (<i>dimensionality reduction</i>). <i>Autoencoder</i> menerima data tidak berlabel; untuk dilakukan pengkodean dan selanjutnya merekonstruksi data seakurat mungkin

Tabel 4. Contoh Input dan Output Neural Network

Input (I0)			Bobot Tersembunyi(I1)				Output(I2)
0	0	1	0.1	0.2	0.5	0.2	0
0	1	1	0.2	0.6	0.7	0.1	1
1	0	1	0.3	0.2	0.3	0.9	1
1	1	1	0.2	0.1	0.3	0.8	0

Tabel 5. Tipe data python

No	Tipe Data	Penjelasan
1	Boolean	Menyatakan benar (<i>true</i>) yang bernilai 1 dan menyatakan salah (<i>false</i>) yang bernilai 0.
2	String	Menyatakan karakter atau kalimat yang bisa berupa huruf atau angka dan lain lain.
3	Float	Menyatakan bilangan yang memiliki koma.
4	Hexadecimal	Menyatakan bilangan dalam format hexa.
5	Complex	Menyatakan pasangan angka real dan imjiner.
6	List	Data untai yang menyimpan berbagai tipe data yang isinya bisa di ubah – ubah.

Tabel 6. Operasi dasar boolean

No	Operasi	Hasil
1	X or Y	If x is false, then y, else x
2	X and Y	If x is false, then x , else y
3	Not X	If x is false, then true, else false

Tabel 7. Operasi Perbandingan

No	Operasi	Arti
1	$<$	Kurang dari
2	$<=$	Kurang dari sama dengan
3	$>$	Lebih besar dari
4	$>=$	Lebih besar dari sama dengan
5	$==$	Sama
6	$!=$	Tidak sama
7	Is	Identitas objek
8	Is Not	Identitas objek yang dinegasikan

Tabel 8. Operasi matematika

No	Operasi	Hasil
1	$x + y$	Sum of x and y
2	$x - y$	Difference of x and y
3	$x * y$	Product of x and y
4	x / y	Quotient of x and y
5	$x // y$	(floored) quotient of x and y
6	$x \% y$	Reminder of x / y
7	$-x$	X negated
8	$+x$	X unchanged
	$\text{abs}(x)$	Absolute value or magnitude of x
	$\text{int}(x)$	X converted to integer
	$\text{Long}(x)$	X converted to long integer

	Complex (re,im)	A complex number with real part re, imaginary part im. Im default to zero
	c.conjugate (x)	Conjugate of the complex number c, (identity on real number)
	Divmod (x,y)	The pair (x // y, x%y)
	pow (x, y)	X the power y
	x ** y	X the power y

Sumber : Widodo Budiharto (2018)

Tabel 9. Parameter Model

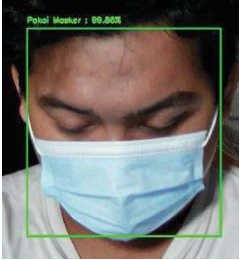
No	Parameter	Value
1	Epoch	8
2	Target Size	128,128
3	Batch size	64
4	Learning Rate	0,001

Tabel 10. Timeline Penelitian

No	Kegiatan	Waktu Penelitian (Tahun 2021) Dalam Minggu Ke -																								Keterangan
		Maret				April				Mei				Juni				July				Agustus				
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
1	Pengajuan Judul Penelitian																									
2	Penyusunan Proposal																									
3	Seminar Proposal																									
4	Pengurusan Administrasi Ijin penelitian																									
5	Pengumpulan Data																									
6	Analisis dan Penafsiran Data																									
7	Tahap Pengujian Analisis																									
8	Penyusunan Laporan Akhir																									
9	Sidang Skripsi																									
10	Perbaikan Hasil Sidang Skripsi																									
11	Pengumpulan Skripsi																									



Tabel 11. Pengujian sistem menggunakan masker

No	Kondisi	Sample	Hasil	
			Berhasil	Gagal
1	Posisi wajah tegak lurus		√	
2	Posisi wajah menghadap kekiri		√	
3	Posisi wajah menghadap kekanan		√	
4	Posisi wajah menghadap keatas		√	


5	Posisi wajah menghadap kebawah		√	
---	--------------------------------	--	---	--


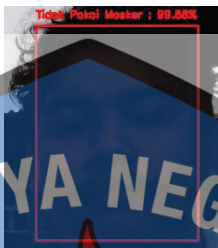
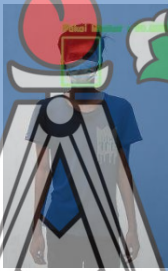


Tabel 12. Pengujian sistem tidak menggunakan masker

No	Kondisi	Sample	Hasil	
			Berhasil	Gagal
1	Posisi wajah tegak lurus		√	
2	Posisi wajah menghadap ke kiri		√	
3	Posisi wajah menghadap ke kanan		√	

4	Posisi wajah menghadap keatas		√	
5	Posisi wajah menghadap kebawah		√	

Tabel 13. Pengujian sistem dengan gangguan eksternal

No	Kondisi	Sample	Hasil		Accuracy
			Berhasil	Tidak	
1	Wajah di tutupi dengan tangan		√		99%
2	Bagian wajah di tutupi benda selain masker			√	87%

3	Kondisi cahaya backlight dengan masker		√		99%
4	Kondisi cahaya tidak menggunakan masker		√		99%
5	Posisi pembacaan dengan jarak deteksi sejauh 2 meter		√		99%
6	Posisi pembacaan dengan jarak deteksi sejauh 3 meter		√		Tidak terdeteksi
7	Sistem mendeteksi 2 object dalam sample		√		

8	Sistem mendeteksi 3 object dalam sample		√		
---	---	---	---	--	--

Tabel 14. Hasil kemampuan sistem secara *realtime*

No	Keterangan	Nilai
1	Total sample data pengujian	18
2	Berhasil Uji	16
3	Gagal Uji	2

Tabel 15. Parameter Delay (Latency)

No	Kondisi	Rata – Rata Delay (ms)	Keterangan	
			Indeks	Kategori
1.	Pakai Masker	199	3	Bagus
2.	Tidak Pakai Masker	200	3	Bagus

Tabel 16. Parameter Paket Loss

No	Kondisi	Paket Yang Diterima	Paket Yang Dikirim	Rata – Rata Paket Loss(%)	Keterangan	
					Indeks	Kategori
1.	Pakai Masker	128	128	0%	4	Sangat Bagus
2.	Tidak Pakai Masker	128	128	0%	4	Sangat Bagus

Tabel 17. Parameter Throughput

No	Kondisi	Rata – Rata throughput (bps)	Keterangan	
			Indeks	Kategori
1.	Pakai Masker	64	3	Bagus
2.	Tidak Pakai Masker	64	3	Bagus

Tabel 18. Parameter Jitter

No	Kondisi	Jitter	Keterangan	
			Indeks	Kategori
1.	Pakai Masker	198	1	Jelek
2.	Tidak Pakai Masker	198	1	Jelek

