

# Sistem Deteksi Masker Dan Pengenalan Wajah Menggunakan Algoritma Convolutional Neural Network

M. Fariz Maulana<sup>1</sup>, Hendrawaty<sup>2</sup>, Muhammad Rizka<sup>3\*</sup>

<sup>1,2,3</sup> Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe  
Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA

<sup>1</sup>m.farizmaulana2015@gmail.com

<sup>2</sup>waty.hendra@gmail.com

<sup>3\*</sup>rizka@pnl.ac.id

*Abstrak — Dimasa Pandemi COVID-19 saat ini pemakaian masker merupakan hal yang wajib digunakan untuk setiap orang ketika menjalani berbagai aktivitas untuk mencegah penyebaran virus COVID-19. Selama ini pendeteksian pemakaian masker dilakukan secara manual melalui pengamatan oleh petugas. Cara yang digunakan ini memiliki beberapa keterbatasan yaitu tidak bisa dilakukan setiap waktu dan setiap tempat. Berdasarkan permasalahan tersebut maka peneliti merancang System Deteksi Masker Dan Pengenalan Wajah Menggunakan Convolutional Neural Network. Sistem ini dirancang untuk lingkungan Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe. Inputan system berupa video realtime kamera yang dapat mendeteksi mahasiswa baik menggunakan masker maupun tidak dan setiap mahasiswa yang tidak menggunakan masker akan di kenali wajahnya. System akan mengirimkan notifikasi kepada pihak petugas apabila terdapat mahasiswa yang tidak menggunakan masker. System menggunakan dataset masker yang diambil dari data yang ada di website Kaggle dan untuk pengenalan wajah sistem menggunakan data wajah dari mahasiswa kelas TI 4C. Hasil yang diperoleh dalam pembuatan model CNN yang diuji menghasilkan akurasi 99% dan Pada pengujian realtime menghasilkan tingkat akurasi keseluruhan sebanyak 75%, akurasi deteksi masker sebesar 83 %, akurasi deteksi wajah sebesar 71%.*

*Kata kunci — Convolutional Neural Network, Deteksi Masker, Notifikasi, Pengenalan Wajah*

*Abstract — During the current COVID-19 pandemic, the use of masks is mandatory for everyone when carrying out various activities to prevent the spread of the COVID-19 virus. So far, the detection of the use of masks has been done manually through observation by officers. This method has several limitations, namely it cannot be done every time and every place. Based on these problems, the researchers designed a Mask Detection System and Face Recognition Using Convolutional Neural Network. This system is designed for the Department of Information and Computer Technology at the Lhokseumawe State Polytechnic. The system input is in the form of a real-time video camera that can detect students both wearing masks or not and every student who does not use a mask will be recognized by his face. The system will send a notification to the officer if there are students who do not wear masks. The system uses a mask dataset taken from the data on the Kaggle website and for facial recognition the system uses facial data from 4C IT class students. The results obtained in the making of the CNN model that were tested resulted in 99% accuracy and realtime testing resulted in an overall accuracy rate of 75%, mask detection accuracy of 83%, face detection accuracy of 71%.*

*Keywords — Convolutional Neural Network, Mask Detection, Notification, Face Recognition*

## I. PENDAHULUAN

Infeksi virus *Corona* yang disebut *COVID-19* (*Corona Virus Disease 2019*) pertama kali ditemukan di kota Wuhan, China pada akhir Desember 2019. Virus ini menular dengan sangat cepat dan menyebar ke hampir semua negara, termasuk Indonesia, hanya dalam waktu beberapa bulan. Hal tersebut membuat beberapa negara memberlakukan kebijakan *lockdown* untuk mencegah virus *Corona* makin meluas. Situasi ini mempengaruhi seluruh aspek kehidupan masyarakat baik itu ekonomi, sosial, budaya, pertahanan, kemanan, dan politik[1].

Mengenakan masker menjadi hal yang sangat penting dimasa pandemi seperti sekarang ini, masker atau penutup wajah berfungsi sebagai alat pelindung proses pernafasan, dengan cara melindungi masuk atau keluarnya udara (mulut dan hidung) dari paparan benda yang menularkan penyakit melalui saluran pernafasan, termasuk virus *COVID-19*. Sebab penularan *COVID-19* sebagian besar melalui *droplets*, masker dapat menjadi penghalang bagi *droplets* baik yang datang dari dalam diri maupun dari orang lain[2]. Tetapi hal tersebut hanya berlaku apabila sebagian besar sampai seluruh masyarakat mengikutinya, karena apabila banyak masyarakat yang tidak mengikutinya, pemakaian masker untuk pencegahan menjadi tidak terlalu efektif.

Pemerintah memberlakukan *New Normal*, yaitu upaya menjaga produktivitas masyarakat di tengah pandemi *COVID-19*. Pada *new normal* ini masyarakat harus mengubah pola hidup menjadi lebih sehat dan tetap memberlakukan protokol kesehatan seperti menjaga jarak dan selalu memakai masker pada lingkungan umum untuk mencegah penyebaran *COVID-19* selagi beraktivitas di luar. Oleh karena itu dibutuhkan kedisiplinan dari setiap masyarakat untuk mengikuti protokol yang ada[1].

Politeknik Negeri Lhokseumawe (PNL) merupakan salah satu institusi pendidikan yang ada di kota Lhokseumawe. Kondisi pada kampus Politeknik Negeri Lhokseumawe pada saat berlakunya kebijakan PPKM yaitu menerapkan protokol kesehatan yang ketat, membatasi kegiatan dalam kampus, dan melakukan perkuliahan secara *online*. Adapun saat kondisi virus *Covid* mulai stabil maka dilakukan perkuliahan secara tatap muka untuk masa uji coba perkuliahan yang akan diikuti oleh mahasiswa 2 angkatan. Perkuliahan dilakukan dengan mengikuti protokol kesehatan seperti vaksinasi, swap, jaga jarak, cuci tangan, dan memakai masker. Salah satu hal yang paling penting dalam melakukan protokol kesehatan merupakan memakai masker saat melakukan kegiatan perkuliahan. Dalam memeriksa apakah mahasiswa menggunakan masker atau tidak dilakukan oleh dosen ataupun *security*.

Berikut merupakan permasalahan dalam mendeteksi masker secara manual pada pintu masuk pada Gedung TIK Politeknik Negeri Lhokseumawe:

1. Harus ada petugas khusus dalam mendeteksi orang yang tidak memakai masker
2. Petugas harus *standby* di depan pintu masuk
3. Lama petugas dalam bertugas mendeteksi/ memeriksa terbatas dalam beberapa jam.
4. Jika tidak ada petugas ditempat maka tidak akan ada yang mendeteksi orang yang tidak memakai masker.

Solusi yang diharapkan yaitu membuat sebuah sistem yang mampu membantu petugas dalam mendeteksi masker dengan menerapkan metode *Convolutional Neural Network (CNN)* secara *realtime* melalui *Camera*. Sistem terbagi menjadi dua

tahapan utama yaitu tahap deteksi lokasi wajah dan tahap klasifikasi wajah. Input yang terdiri dari banyak wajah dideteksi menggunakan metode *Haar Cascade Classifier*. Kemudian setiap wajah yang dideteksi, diklasifikasikan menggunakan metode *CNN* dan jika terdeteksi tidak pakai masker maka system akan kenali wajahnya dan akan diberikan notifikasi kepada pihak petugas/keamanan(*security*).

Dengan adanya *system* ini diharapkan dapat membantu petugas dalam melakukan pengecekan masker dan dapat meningkatkan kesadaran mahasiswa untuk menggunakan masker.

*CNN* seperti metode *neural network* pada umumnya, terbentuk oleh berlapis-lapis *neuron* yang memiliki berat (*weight*) dan bias yang dapat diatur. Setiap *neuron* memperoleh masukan dari *layer input*, melakukan *dot-product* pada *layer-layer* berikutnya dan menghasilkan keluaran pada *layer output*[3]. Pada umumnya semakin banyak jumlah *layer*, semakin tinggi akurasi dan semakin kompleks kemampuan dari jaringan, namun pada suatu titik akan terjadi *diminishing return*, dimana peningkatan jumlah *layer* tidak meningkatkan kinerja dari jaringan[4]. *Network* ini dilatih untuk mengambil data mentah dan membuat korelasi antara data tersebut terhadap skor yang diperoleh di akhir. Semakin tinggi skor akhir, semakin bias jaringan terhadap konfigurasi tertentu yang menghasilkan skor tersebut[5].

*CNN* merupakan *deep neural network*, arsitektur *neural network* yang terdiri dari 2 atau lebih dari 2 *hidden layer*. Ciri khas dari *CNN* adalah dalam *CNN* data masukan selalu berukuran 2 dimensi dan berupa gambar, sehingga ada property-properti tertentu yang dapat di-*encode* ke dalam arsitektur *network*[6].

*CNN* secara umum terdiri dari 3 jenis *layer neuron*, yaitu *Convolutional Layer*, *Pooling Layer*, and *Fully Connected Layer*. *Layer* ini kemudian ditumpuk untuk membentuk sebuah arsitektur *CNN* yang lengkap. Pada *layer pertama Convolution Layer*, memiliki *filter* yang berdimensi kecil yang kemudian dikonvolusikan di atas gambar (direpresentasikan ke dalam bentuk matriks) yang masuk. Ketika berada di atas piksel tertentu, *filter* kemudian menghasilkan keluaran berupa matriks *citra*. *Layer* kedua, *Pooling Layer* melakukan *downsampling* terhadap data untuk mengurangi harga komputasi dalam memproses matriks. Pada *layer* ketiga *Fully-Connected Layer*, merupakan *layer* yang memiliki koneksi penuh terhadap *layer-layer* sebelumnya, *layer* ini sama seperti pada *neural network* biasanya[7]. Pada *CNN*, terdapat 5 jenis *layer* yang digunakan, yaitu:

### 1. Convolutional Layers

Merupakan tulang belakang dari *CNN*, *layer konvolusi* mengambil data *citra*, dan mengaplikasikan sebuah *kernel/filter* yang bergeser diatas *citra*. *Kernel* tersebut kemudian melakukan ekstraksi fitur dengan cara melakukan *dot-product*. Hasil dari *dot-product* ini kemudian dapat diserahkan ke *layer* selanjutnya. Setiap *kernel* dari *layer konvolusi* memiliki *weight* sebesar ukuran dari *kernel* tersebut yang kemudian akan diubah-ubah menggunakan *backpropagation*. Satu *layer konvolusi* dapat memiliki lebih dari satu jenis *kernel*, dimana masing-masing *kernel* memiliki *weight* yang berbeda dan mengekstraksi fitur yang berbeda pula. Seluruh *kernel* ini kemudian akan membentuk *citra* baru (*citra* hasil perkalian *dot-product*) dengan kedalaman *channel* sesuai dengan jumlah *kernel*. *Layer konvolusi* menerima *input* dengan ukuran 2-d *spasial (matrix)* sedangkan *layer fully-connected* hanya menerima *input* dengan ukuran 1-d *spasial (array)*.

## 2. Pooling Layers

Layer Pooling digunakan untuk *downsampling matrix* gambar, sehingga harga dari arsitektur berkurang. Layer ini tidak memiliki *weight*, dan hanya berlaku sebagai *filter*.

## 3. Fully Connected Layers

FC Layer pada CNN sama pada dasarnya dengan FC yang terdapat pada *neural network* lainnya. FC dan *Convolution layer* memiliki fungsi yang mirip dalam melakukan *training* dan menyimpan *weight* dari model. Namun, dikarenakan hanya mendukung data dengan 1 dimensi *spasial*, FC sebaiknya hanya diterapkan diakhir arsitektur. Ini guna mencegah kehilangan data akibat mengubah data gambar 2-d *spasial* menjadi data 1-d *spasial*.

## 4. Padding Layer

Ketika konvolusi dilakukan, ukuran citra akan mengecil. Untuk mempertahankan ukuran citra, maka dilakukan padding pada citra menggunakan *Padding Layer*. *Padding layer* tidak menyimpan *weight* dari *neural network*.

## 5. Dropout layer

*Dropout layer* dimasukkan untuk mematikan *neuron* secara *random* pada *layer* sebelumnya (biasanya ditaruh setelah *fully-connected layer*). Penggunaan *dropout layer* umumnya dilakukan untuk mencegah *overfitting* akibat *neuron* yang terlalu bergantung antar satu sama lain. Akibatnya, dihasilkan model yang lebih *independent* dan fleksibel, dengan performa *training* yang meningkat[5].

Python yang didirikan pada tahun 1991 oleh seorang developer Guido Van Rossum merupakan salah satu bahasa pemrograman tingkat tinggi yang bersifat interpreter, general-purpose dan pemrograman berorientasi objek. Dalam melakukan coding, python membantu programmer untuk melakukan coding yang lebih sedikit dan simpel dari pada Java atau C++[8]. Beberapa kelebihan python yang menjadi pilihan pada penelitian ini yaitu banyaknya library yang tersedia dan sampai sekarang masih terus dikembangkan. Kelebihan lain yang dimiliki python dapat diintegrasikan dengan bahasa pemrograman lain seperti C, C++ atau Java[2].

Pada penelitian ini menggunakan beberapa library antara lain keras, tensorflow, sklearn, numpy, opencv dan matplotlib. Keras merupakan suatu library dengan tujuan untuk membangun suatu sistem machine learning, library keras banyak digunakan pada pembangunan machine learning karena mudah diimplementasikan dan menghemat script code yang ditulis. Numpy merupakan suatu library yang berfungsi untuk mempermudah perhitungan array yang lebih dari satu dimensi, karena pada penelitian ini data masukan berupa citra yang memiliki nilai pixel yang banyak dan akan dimasukkan ke dalam array yang lebih dari satu dimensi. Opencv merupakan suatu library yang berfungsi untuk melakukan pengolahan citra. Matplotlib merupakan library yang berfungsi untuk menampilkan data analisis berbentuk chart dan dapat pula untuk menampilkan gambar secara rapih.

*Haar like feature* atau yang dikenal sebagai *Haar Cascade Classifier* merupakan *rectangular feature*, yang memberikan indikasi secara spesifik pada sebuah gambar atau *image*. *Haar cascade classifier* berasal dari gagasan Paul Viola dan Michael Jhon. Ide dari *Haar like feature* adalah mengenali obyek berdasarkan nilai sederhana dari fitur tetapi bukan merupakan nilai *pixel* dari *image* objek tersebut[9].

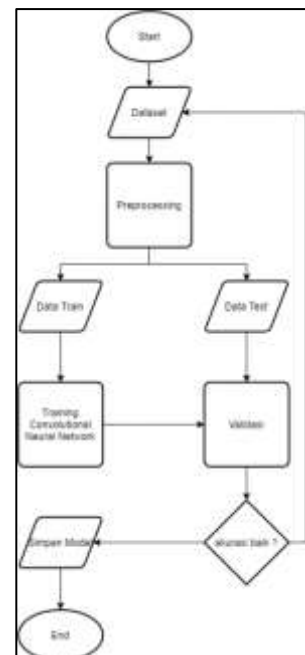
Algoritma *Haar Cascade Classifier* adalah salah satu algoritma yang digunakan untuk mendeteksi sebuah wajah. Algoritma tersebut mampu mendeteksi dengan cepat dan *realtime* sebuah benda termasuk wajah manusia. Algoritma *Haar Cascade Classifier* memiliki kelebihan yaitu perihai komputasi yang cepat karena tersebut hanya bergantung pada

jumlah piksel dalam persegi dari sebuah *image*. *Open CV* memiliki model yang dibuat berdasarkan konsep *Cascade Classifier* ini, namanya adalah *frontface haarcascade classifier* yang dapat digunakan untuk mendeteksi wajah[10].

## II. METODOLOGI PENELITIAN

### A. Arsitektur Umum

Gambar 1 menjelaskan arsitektur umum perancangan sistem. proses dimulai dari mengumpulkan dataset berupa dataset masker dan dataset wajah, Masing-masing data kemudian dilakukan proses *preprocessing*. Dataset selanjutnya dipisah menjadi data latih dan data uji. Data latih digunakan untuk melatih algoritma, sedangkan data uji diguakan untuk menguji, mengevaluasi, dan mengukur kinerja model *machine learning*. Kemudian merancang model *machine learning* dan mengimplementasikannya di dalam website.



Gambar 1. Arsitektur Umum

### B. Dataset

Dataset yang digunakan dalam penelitian berjumlah 12000 data. Dataset terbagi dua, yaitu data masker dan data wajah. Data masker berjumlah 1000 gambar yang diambil dari situs [www.kaggle.com/](http://www.kaggle.com/) dengan kata kunci "face mask". Data wajah terdiri dari 11 wajah mahasiswa, masing-masing data wajah mahasiswa berjumlah 100 gambar.

### C. Preprocessing

Proses preprocessing bertujuan untuk memperkecil (*resize*) tiap citra yang ada menjadi ukuran tertentu untuk memudahkan proses untuk merancang model *machine learning*. Tahapan-tahapan yang dilakukan meliputi:

#### 1) Grayscale

*Grayscale* mengubah seluruh citra RGB menjadi citra *grayscale*. Tahap ini digunakan untuk menyederhanakan model citra.

#### 2) Resize

Proses *resize* dilakukan untuk menyamakan ukuran dari seluruh citra yang akan diproses, dengan dilakukan proses *resize* maka proses analisis akan lebih mudah dan cepat dikarenakan ukuran citra sesuai dengan kemampuan dari perangkat yang digunakan.

3) *Augmentasi*

Tahapan ini merupakan proses untuk mengatasi masalah dengan dataset kecil pada beberapa kelas, dapat menambah data untuk menambah jumlah sampel. Augementasi yang dilakukan pada penelitian ini adalah rotasi gambar dan juga menambah cahaya atau juga mengurangi cahaya. Setelah di lakukan augmentasi ini, jumlah citra akan bertambah sebanyak 20 kali lipat pada setiap dataset yang berawal dataset setiap mahasiswa 100 menjadi 2000

4) *Labelling*

Tahapan ini merupakan proses pelabelan wajah mahasiswa dan masker untuk setiap dari proses augmentasi. Pada algoritma *machine learning* tidak dapat secara langsung memproses data kategorikal terutama untuk masalah klasifikasi menggunakan metode *deep learning*. Pada proses *encoding label*, data yang bersifat kategorikal diubah kedalam bentuk nilai numerik.

D. *Perancangan Model Machine Learning*

Tahap merancang model machine learning diawali dengan mendefinisikan data latih (x) dan data uji (y). Data latih (x) bersifat independent yang digunakan untuk melatih algoritma, sedangkan data uji (y) bersifat dependen yang digunakan untuk menguji, mengevaluasi, dan mengukur kinerja model *machine learning*. Proporsi data latih sebesar 85% dan data uji sebesar 15%. Tahapan-tahapan algoritma *convolutional neural network* adalah:

1) *Convolution & ReLU*

Citra yang telah di-processing dari tahap sebelumnya akan dilakukan proses convolution. Pada tahap ini, citra akan digambarkan dalam bentuk matriks yang terdiri dari angka 0 sampai 255. Lalu, citra akan di konvolusi (*convolution*) dengan beberapa *filter*, hasilnya akan diteruskan pada proses *ReLU* yang dilakukan dengan mengubah setiap elemen citra konvolusi yang bernilai dibawah 0 (*minus/negatif*) diubah atau dimutlakan menjadi 0.

2) *Pooling layer*

Membagikan nilai output dari *convolution layer* menjadi beberapa kotak kecil, kemudian menggunakan fungsi *max polling* untuk mengambil nilai maksimal dari setiap kotak dengan untuk menyusun matriks citra yang telah ditentukan untuk mengurangi dimensi fitur matriks.

3) *Flatten*

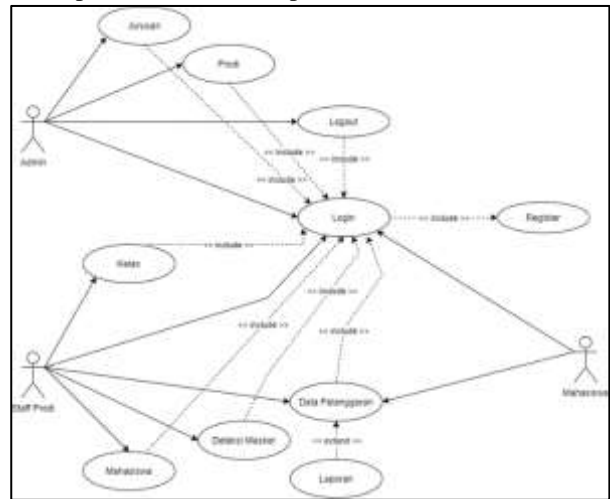
Mengubah nilai masukan menjadi sebuah nilai *array* hasil *pooling*. Setiap hasil dari proses *Pooling Layer* akan diubah ke dalam *array* satu dimensi.

4) *Fully Connected Layer*

Proses dimana matriks yang sudah di-*flatten* dimasukkan untuk melewati jaringan *neuron* agar dapat memprediksi probabilitas keluaran Nilai *flatten* dari proses akan digunakan untuk melatih *neural network*.

E. *Use Case Diagram*

*Use case diagram* adalah satu dari berbagai jenis diagram UML (*Unified Modelling Language*) yang menggambarkan hubungan interaksi antara sistem dan aktor. *Use case* dapat mendeskripsikan tipe interaksi antara pengguna sistem dengan sistemnya. Pada penelitian ini terdapat 3 use case yaitu use case admin, staff prodi dan mahasiswa.



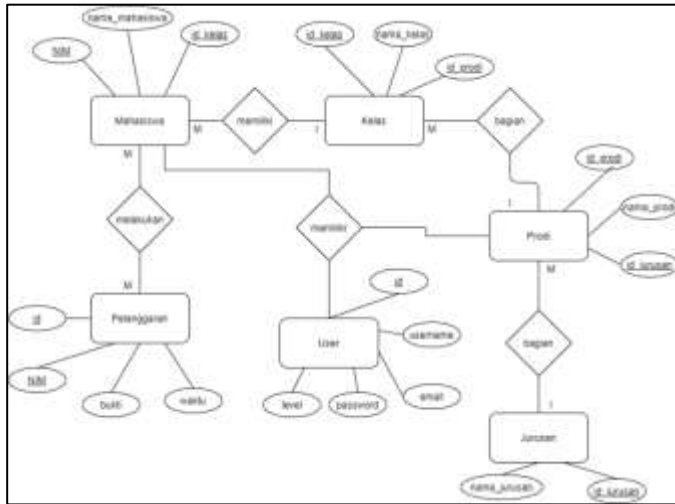
Gambar 2. *Use Case Diagram*

Adapun penjelasan dari Gambar 2 sebagai berikut:

- 1) **Mahasiswa**  
Mahasiswa merupakan salah satu user pada sistem ini. Mahasiswa hanya bisa melihat data pelanggaran.
- 2) **Staff Prodi**  
Staff Prodi merupakan salah satu user pada sistem ini. Staff Prodi dapat melakukan proses lebih banyak dari pada Mahasiswa, seperti menginput data kelas, data mahasiswa, melihat data pelanggaran dan dapat melihat tampilan monitor deteksi masker.
- 3) **Admin**  
Admin merupakan user yang mengelola sistem ini. Admin dapat melakukan proses seperti menginput data jurusan, data prodi dan membuat akun untuk staf prodi.
- 4) **Jurusan**  
Jurusan mempunyai data-data jurusan yang akan berelasi dengan prodi.
- 5) **Prodi**  
Prodi mempunyai data-data prodi yang akan berelasi dengan kelas.
- 6) **Kelas**  
Kelas mempunyai data-data kelas yang akan berelasi dengan prodi.
- 7) **Mahasiswa**  
Mahasiswa mempunyai data-data mahasiswa yang akan berelasi dengan pelanggaran.
- 8) **Data Pelanggaran**  
Data Pelanggaran berfungsi untuk menyimpan data-data mahasiswa yang melakukan pelanggaran seperti tidak menggunakan masker.
- 9) **Laporan Halaman**  
laporan dapat mencetak laporan data pelanggaran masker.
- 10) **Deteksi Masker**  
Deteksi Makser merupakan menu untuk melihat tampilan kamera deteksi masker.
- 11) **Login**  
Pengguna harus melakukan login supaya dapat mengakses berbagai fitur pada aplikasi.
- 12) **Register**  
*Register* merupakan Langkah pertama yang dilakukan pengguna untuk mendaftarkan data diri ke dalam aplikasi agar dapat mengakses aplikasi.
- 13) **Logout**  
Menjelaskan tentang aktifitas yang dapat dilakukan pengguna. Jika pengguna memilih menu tersebut maka pengguna akan keluar dari Aplikasi.

F. Entity Relationship Diagram

Entity Relationship Diagram (ERD) yang digunakan dalam sistem ini yaitu entitas tabel users, jurusan, prodi, kelas, mahasiswa dan pelanggaran. Relasi antar tabel dalam aplikasi deteksi masker dan pengenalan wajah ini dapat dilihat pada Gambar3.



Gambar 3. Entity Relationship Diagram

Pada Gambar 3 ERD Sistem Deteksi Masker Dan Pengenalan Wajah Menggunakan Algoritma Convolutional Neural Network memiliki 6 entitas utama yang digunakan. Berikut penjelasannya:

- 1) Tabel User  
Tabel user memiliki 5 attribute yaitu id, username, email, password, level. id sebagai primary key.
- 2) Tabel Jurusan  
Tabel Jurusan memiliki 2 attribute yaitu id\_jurusan, nama\_jurusan. id\_jurusan sebagai primary key.
- 3) Tabel Prodi  
Tabel Prodi memiliki 3 attribute yaitu id\_prodi, nama\_prodi, id\_jurusan. id\_prodi sebagai primary key dan id\_jurusan sebagai foreign key.
- 4) Tabel Kelas  
Tabel Kelas memiliki 3 attribute yaitu id\_kelas, nama\_kelas, id\_prodi. id\_kelas sebagai primary key dan id\_prodi sebagai foreign key.
- 5) Tabel Mahasiswa  
Tabel Mahasiswa memiliki 3 attribute yaitu NIM, nama\_mahasiswa, id\_kelas. NIM sebagai primary key dan id\_kelas sebagai foreign key.
- 6) Tabel Pelanggaran  
Tabel Pelanggaran memiliki 4 attribute yaitu id, NIM, bukti, waktu. id sebagai primary key dan NIM sebagai foreign key.

III. HASIL DAN PEMBAHASAN

A. Implementasi Proses

Implementasi proses menjabarkan prosedur dan tahapan yang terlibat berdasarkan perancangan model machine learning yang telah dijabarkan.

1) Grayscale

Proses grayscale menjadikan semua citra RGB menjadi citra keabuan (grayscale). Hasil proses grayscale ditunjukkan pada Gambar 4.



Gambar 4. Hasil proses Grayscale

2) Reziye Citra

Proses reziye citra menjadikan semua citra menjadi citra dengan reziye 50x50. Hasil proses reziye citra ditunjukkan pada Gambar 6.



Gambar 5. Hasil Proses Reziye Citra

3) Augmentasi

Proses Augmentasi dilakukan pada setiap citra yang terdapat di dalam dataset yang akan dilakukan transformasi berupa rotasi gambar, penggeseran gambar, penyesuaian warna grayscale baik cerah maupun gelap, secara random pada setiap citra yang terdapat pada dataset dengan rentang parameter yang telah ditetapkan. Setelah di lakukan augmentasi ini, jumlah citra akan bertambah sebanyak 20 kali lipat pada setiap dataset yang berawal dataset setiap mahasiswa 100 menjadi 2000. Hasil proses augmentasi ditunjukkan pada Gambar 6.



Gambar 6. Hasil proses Augmentasi

4) Labelling

Proses Labeling dataset dilakukan setelah melakukan augmentasi dataset, setiap gambar yang telah di augmentasi dilakukan pelabelan berdasarkan nama masing-masing folder di dalam dataset. Setiap label dataset masing-masing mempunyai 2000 gambar yang memiliki nilai numerik setiap masing-masing gambar. Label juga berfungsi untuk membaca nilai output dari numerik menjadi nilai kategorikal untuk bisa

di pahami hasil dari output deteksi. Hasil proses *Labelling* ditunjukkan pada Gambar 7.

```

le = LabelEncoder()
le.fit(names)
labels = le.classes_

name_vec = le.transform(names)

categorical_name_vec = to_categorical(name_vec)
print("number of class :", len(labels))
print(labels)

```

✓ 0.5s

```

number of class : 12
['1857301003' '1857301019' '1857301021' '1857301023' '1857301033'
'1857301035' '1857301041' '1857301047' '1857301055' '1857301059'
'1857301061' 'mask']

```

Gambar 7. Hasil Proses *Labelling*

5) Perancangan Model *Machine Learning*

Tahap merancang model *machine learning* diawali dengan mendefinisikan data latih (*x*) dan data uji (*y*). Data latih (*x*) bersifat independent yang digunakan untuk melatih algoritma, sedangkan data uji (*y*) bersifat dependen yang digunakan untuk menguji, mengevaluasi, dan mengukur kinerja model *machine learning*. Proporsi data latih adalah 85% dari data, yaitu berjumlah 20400 gambar dan proporsi data uji adalah 15% dari data, yaitu berjumlah 3600 gambar.

Terdapat beberapa proses dalam menggunakan algoritma *convolutional neural network* yaitu *convolution layer*, *pooling layer*, dan *fully connection layer* yang terdiri dari *flatten* dan *dense*. Ringkasan struktur *layer* ditunjukkan Gambar 8.

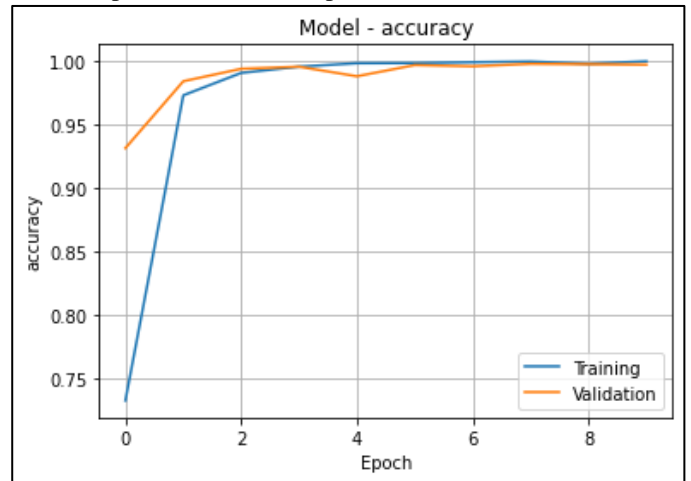
Layer (type)	Output Shape	Param #
conv2d_36 (Conv2D)	(None, 48, 48, 64)	640
conv2d_37 (Conv2D)	(None, 46, 46, 64)	36928
max_pooling2d_18 (MaxPoolin g2D)	(None, 23, 23, 64)	0
conv2d_38 (Conv2D)	(None, 21, 21, 128)	73856
conv2d_39 (Conv2D)	(None, 19, 19, 128)	147584
max_pooling2d_19 (MaxPoolin g2D)	(None, 9, 9, 128)	0
flatten_9 (Flatten)	(None, 10368)	0
dense_27 (Dense)	(None, 128)	1327232
dense_28 (Dense)	(None, 64)	8256
dense_29 (Dense)	(None, 12)	780
activation_9 (Activation)	(None, 12)	0

-----  
Total params: 1,595,276  
Trainable params: 1,595,276  
Non-trainable params: 0

Gambar 8. Ringkasan Struktur *Layer* Model

6) Pengujian dan Evaluasi

Pada pengujian ini penulis menguji dengan 10 *epoch*. Gambar 9 menunjukkan hasil akurasi 10 *epoch*.



Gambar 9. Grafik Akurasi 10 *epoch*

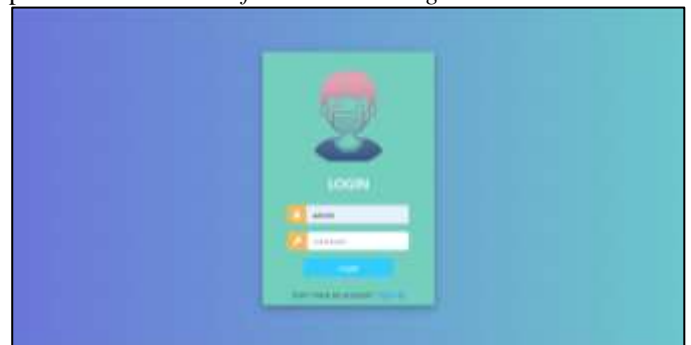
Berdasarkan hasil pengujian menunjukkan bahwa model mampu menyesuaikan data latih dengan baik ke dalam tahap pembelajaran. Hal tersebut digambarkan melalui peningkatan grafik *accuracy* seiring dengan penambahan jumlah *epochs*. Model memiliki *accuracy* 99.90% dan *val\_accuracy* 99,74%. Nilai *val\_accuracy* menunjukkan perbandingan hasil data validasi dengan hasil prediksi model pada setiap periode pelatihan.

B. Implementasi Sistem

Implementasi sistem menjabarkan implementasi dari perancangan sistem.

1) Halaman Login

Halaman *Login* adalah halaman proses untuk masuk ke dalam sebuah aplikasi. Halaman *login* sistem dapat dilihat pada Gambar 10 *Interface* Halaman *Login*

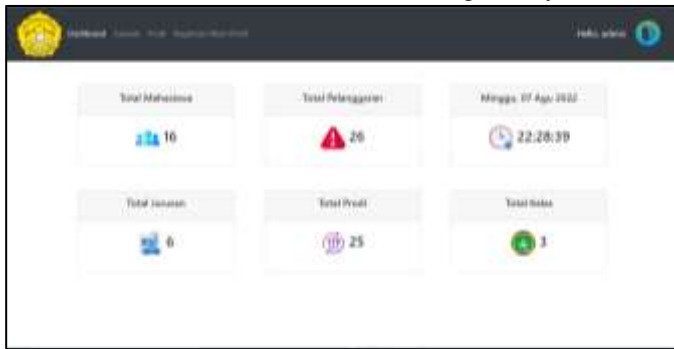


Gambar 10. Halaman Login

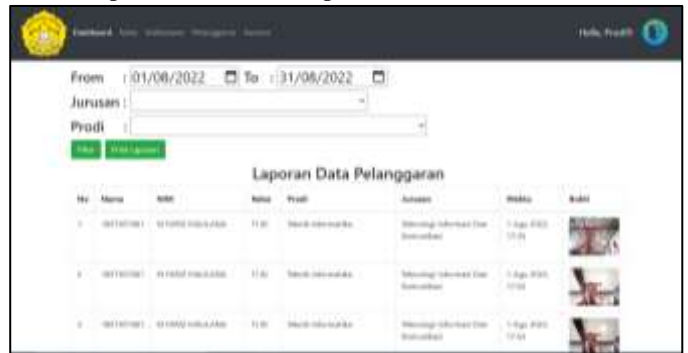
2) Halaman Dashboard

Halaman ini adalah halaman awal yang akan diakses oleh *Operator* setelah proses *login* selesai. Pada halaman *Dashboard Operator* ini akan menampilkan jumlah data yang terdapat pada sistem seperti: Jumlah Mahasiswa, Jumlah Pelanggaran, Jumlah Jurusan, Jumlah Prodi, dan Jumlah Kelas. Berikut tampilan halaman *Dashboard Operator* pada Gambar 11 *Interface* Halaman *Dashboard Operator*.





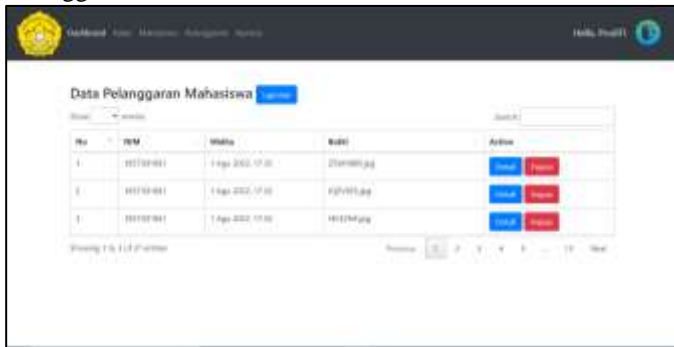
Gambar 11. Halaman Dashboard



Gambar 14. Halaman Laporan

3) Halaman Data Pelanggaran

Pada halaman ini staf prodi dapat melihat *detail* data mahasiswa dan hapus data pelanggaran mahasiswa pada tabel pelanggaran mahasiswa. Data Pelanggaran diinput *otomatis* pada program jika terdeteksi mahasiswa yang tidak menggunakan masker. Berikut tampilan halaman pelanggaran mahasiswa pada Gambar 12 *Interface* Halaman Data Pelanggaran Mahasiswa.



Gambar 12. Halaman Data Pelanggaran Mahasiswa



Gambar 15. Output Laporan .pdf

4) Halaman *Detail* Pelanggaran

Halaman *detail* pelanggaran menampilkan identitas dan foto dari mahasiswa yang tidak menggunakan masker. Identitas yang ditampilkan berupa Nama, NIM, Kelas, Prodi, Jurusan, dan waktu pelanggaran. Data tersebut diinputkan secara *otomatis* oleh sistem saat mahasiswa tidak menggunakan masker. *Interface* Halaman *detail* pelanggaran dapat dilihat pada Gambar 13.



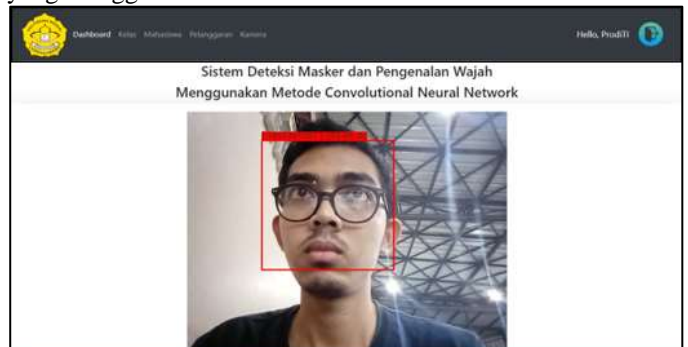
Gambar 13. Halaman Detail Pelanggaran

5) Halaman Laporan Data Pelanggaran

Halaman laporan merupakan halaman yang hanya dapat diakses oleh staf Prodi. Pada halaman ini menampilkan *report/laporan* dari pelanggar yang terdeteksi. Untuk mencetak laporan *user* dapat melakukan filter bulan, jurusan, dan prodi, selanjutnya *user* bisa klik *Print* Laporan maka secara otomatis laporan akan di unduh. Halaman laporan dapat dilihat pada Gambar 14 dan Gambar 15 menampilkan hasil report dari data yang sudah difilter sebelumnya.

6) Halaman Kamera

Halaman kamera merupakan halaman khusus staf prodi untuk menampilkan *monitor* deteksi masker. Pada halaman ini terdapat video *realtime* dari kamera yang digunakan untuk proses *mask detection*. Wajah yang tidak menggunakan masker akan di tandai dengan kotak merah dan data *frame* wajah dan nim yang ditampilkan akan dikirim kedalam data pelanggaran mahasiswa sedangkan wajah yang terdeteksi menggunakan masker akan di deteksi dengan kotak hijau. Gambar 16 menampilkan wajah yang di deteksi tidak menggunakan masker dan Gambar 17 menampilkan wajah yang menggunakan masker.



Gambar 16. Tampilan kamera tidak menggunakan masker



Gambar 17. Tampilan kamera menggunakan masker

C. *Pengujian system*

Pengujian sistem merupakan tahapan akhir dari pembangunan perangkat lunak, dimana analisis sistem melakukan pengujian sistem secara menyeluruh untuk memastikan sistem telah berjalan dengan baik. Pada tahap ini dilakukan pengujian sistem deteksi makser dan pengujian pengenalan wajah.

TABEL I.  
PENGUJIAN SYSTEM

No	Deteksi	Kondisi	Prediksi	Hasil
1	Masker	Jarak < 1 m dan cahaya cerah	Mask	Sesuai
2	Masker	Jarak > 1 m	Mask	Sesuai
3	Masker	Jarak > 1 m	Mask	Sesuai
4	Masker	Jarak > 1 m	Mask	Tidak Sesuai
5	Masker	Jarak < 1 m dan cahaya redup	Mask	Sesuai
6	Masker	Jarak < 1 m dan cahaya redup	Tidak dikenal	Sesuai
7	Wajah	Jarak < 1 m	1857301061	Sesuai
8	Wajah	Jarak < 1 m	1857301023	Sesuai
9	Wajah	Jarak < 1 m dan multidetect	1857301061 1857301023	Sesuai
10	Wajah	Jarak < 1 m dan multidetect	1857301023	Sesuai
11	Wajah	Jarak < 1 m dan multidetect	Tidak dikenal 1857301023 1857301033	Tidak Sesuai
12	Wajah	Jarak > 1 m dan multidetect	Tidak dikenal 1857301023 Tidak dikenal	Tidak Sesuai
13	Wajah	Jarak > 1 m	1857301003 Tidak dikenal	Sesuai
14	Wajah	Jarak > 1 m	1857301035	Sesuai
15	Wajah	Jarak > 1 m	1857301023	Tidak Sesuai
16	Wajah	Jarak > 1 m	1857301003	Tidak Sesuai
17	Wajah	Jarak > 1 m	1857301003	Sesuai
18	Wajah	Jarak > 1 m	1857301003	Sesuai
19	Wajah	Jarak < 1 m	1857301003	Sesuai
20	Wajah	Jarak > 1 m	Tidak dikenal	Sesuai

IV. KESIMPULAN

Setelah melalui proses pengujian dan implementasi, dapat disimpulkan bahwa sistem dapat digunakan untuk deteksi masker dan pengenalan wajah. Namun, faktor yang mempengaruhi sistem menjadi kurang optimal diantaranya yaitu intensitas cahaya, jarak dan kualitas kamera. *Sistem Deteksi Masker Dan Pengenalan Wajah Menggunakan algoritma Convolutional Neural Network* menghasilkan akurasi sebesar 99,90% dan val\_accuracy sebesar 99,74% menggunakan hyperparameters epochs 10 dan untuk pengujian *realtime* menghasilkan tingkat akurasi keseluruhan sebanyak 75% dari 20 pengujian.

REFERENSI

[1] M. N. Baay, "Sistem Otomatis Pendeteksi Wajah Bermasker Menggunakan Deep Learning," *ITS*, 2021.  
 [2] A. S. Nainggolan, "Implementasi Convolutional Neural Network Untuk Deteksi Pengguna Masker," *UIN Suska Riau*, 2021.  
 [3] H. Abhirawa, "Pengenalan Wajah Menggunakan Convolutional Neural Network," *Univ. Telkom*, 2017.  
 [4] Felix, "Implementasi Convolutional Neural Network Untuk Identifikasi Jenis Tanaman Melalui Daun,"

*STMIK Mikroskil*, 2020.  
 [5] V. D. Win, "Pengenalan Wajah Menggunakan Convolutional Neural Network," *Inst. Teknol. Sepuluh Nop. Surabaya*, 2018.  
 [6] I. W. Suartika, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101," *Inst. Teknol. Sepuluh Nop.*, 2016.  
 [7] R. A. Z. Fajar Astuti Hermawati, "Sistem Deteksi Pemakaian Masker Menggunakan Metode Viola-Jones dan Convolutional Neural Networks (CNN)," 2021.  
 [8] R. F. Muharram, "Implementasi Artificial Intelligence Untuk Deteksi Masker Secara Realtime Dengan Tensorflow Dan Ssd Mobilenet Berbasis Python," *Univ. Indraprasta PGRI*, 2021.  
 [9] B. Budiman, "Pendeteksian Penggunaan Masker Wajah Dengan Metode Convolutional Neural Network," *Univ. Tarumanagara*, 2020.  
 [10] G. A. Anarki, "Penerapan Metode Haar Cascade Pada Aplikasi Deteksi Masker," *Inst. Teknol. Nas. Malang*, 2021.