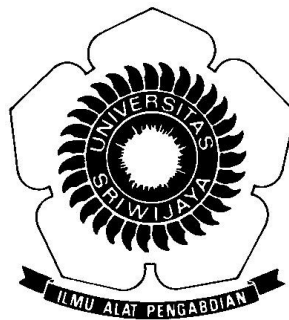


PENGENALAN CITRA WAJAH BERMASKER MENGGUNAKAN CNN  
DENGAN PENDEKATAN *ANGULAR MARGIN LOSS*

Diajukan Sebagai Syarat Untuk Menyelesaikan  
Pendidikan Program Strata-1 Pada  
Jurusan Teknik Informatika



Oleh:

Ferza Reyaldi  
NIM: 09021281924060

**Jurusan Teknik Informatika**  
**FAKULTAS ILMU KOMPUTER UNIVERSITAS SRIWIJAYA**  
Tahun 2022

## LEMBAR PENGESAHAN SKRIPSI

### Pengenalan Citra Wajah BERMASKER Menggunakan CNN Dengan Pendekatan *ANGULAR MARGIN LOSS*

Oleh:

Ferza Reyaldi

NIM: 09021281924060

Indralaya, 19 Desember 2022

Pembimbing 1

Pembimbing 2

Dr. Muhammad Fachrurrozi, M.T.  
NIP 198005222008121002

Muhammad Qurhanul Rizqie, M.T., Ph.D

Mengetahui,

Ketua Jurusan Teknik Informatika

Alvi Syahrini Utami, M.Kom.  
NIP 197812222006042003

## TANDA LULUS UJIAN KOMPREHENSIF SKRIPSI

Pada hari XX tanggal XX telah dilaksanakan telah dilaksanakan ujian komprehensif skripsi oleh Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya.

Nama : Ferza Reyaldi  
NIM : 09021281924060  
Judul : Pengenalan Citra Wajah Bermasker Menggunakan CNN dengan Pendekatan *Angular Margin Loss*

dan dinyatakan **LULUS**.

1. Penguji I

Nama .....  
NIP

2. Pembimbing I

Dr. Muhammad Fachrurrozi, M.T. .....  
NIP 198005222008121002

3. Pembimbing II

Muhammad Qurhanul Rizqie, M.T., Ph.D .....  
NIP

Mengetahui,  
Ketua Jurusan Teknik Informatika

Alvi Syahrini Utami, M.Kom.  
NIP 197812222006042003

## HALAMAN PERNYATAAN BEBAS PLAGIAT

Yang bertanda tangan di bawah ini:

Nama : Ferza Reyaldi

NIM : 09021281924060

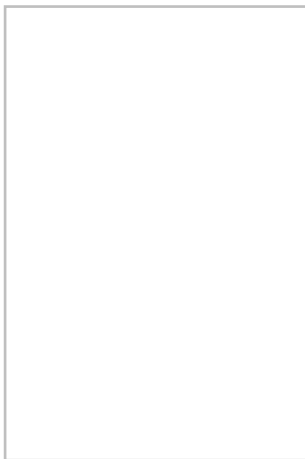
Program Studi : Teknik Informatika Reguler

Judul Skripsi : Pengenalan Citra Wajah Bermasker Menggunakan CNN  
dengan Pendekatan *Angular Margin Loss*

**Hasil Pengecekan *iThenticate*/Turnitin: ....%**

Menyatakan bahwa laporan proyek saya merupakan hasil karya sendiri dan bukan hasil penjiplakan/plagiat. Apabila ditemukan unsur penjiplakan/plagiat dalam laporan proyek ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya sesuai dengan ketentuan yang berlaku.

Demikian, pernyataan ini saya buat dengan sebenarnya dan tidak ada paksaan oleh siapa pun.



Indralaya, 19 Desember 2022

Ferza Reyaldi

NIM 09021281924060

## MOTTO DAN PERSEMBAHAN

*"You Only Live Once. But if you do it right, once is enough."*

- Mae West

*"Every journey has its final day. Don't rush."*

- Zhongli

Kupersembahkan Karya Tulis ini kepada:

- Allah SWT
- Orang Tua dan Saudara-Saudaraku
- My Besto Furendo

# ***MASKED-FACE IMAGE RECOGNITION USING CNN WITH ANGULAR MARGIN LOSS APPROACH***

**FERZA REYALDI (09021281924060)**

Informatics Engineering, Faculty of Computer Science, Sriwijaya University

Email: [reyaldiferza@gmail.com](mailto:reyaldiferza@gmail.com)

## **ABSTRACT**

Face recognition is a biometric recognition solution that does not require direct contact with the user and achieves a high level of accuracy. The use of face masks that were enforced during the Covid-19 Pandemic caused partial occlusion of facial features, making it difficult to identify a person's identity. Convolutional Neural Network (CNN) is a deep learning method used to extract and study facial features. This research will perform masked face recognition using CNN with the help of Angular margin loss. Angular margin loss allows to get very discriminatory features for face recognition. The dataset used in this study is the Real-World Masked Face Dataset (RMFD). Based on the results of this study, the use of angular margin loss shows an effect on the facial recognition model. The results of this study showed that the accuracy of the training data was 36.33% higher than the previous research model and the validation data accuracy was 12.65% higher than the previous research model.

**Keywords:** Masked Face Recognition, Convolutional Neural Network, Angular Margin Loss, Real-World Masked Face Dataset

# **Pengenalan Citra Wajah Bermasker Menggunakan CNN Dengan Pendekatan Angular Margin Loss**

**FERZA REYALDI (09021281924060)**

Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya

Email: [reyaldiferza@gmail.com](mailto:reyaldiferza@gmail.com)

## **ABSTRAK**

Pengenalan wajah adalah solusi pengenalan biometrik yang tidak memerlukan kontak langsung dengan pengguna dan tingkat akurasi yang dicapai tinggi. Pemakaian masker wajah yang diberlakukan saat Pandemi Covid-19 menyebabkan oklusi parsial pada fitur-fitur wajah sehingga identitas seseorang sulit dikenali. *Convolutional Neural Network* (CNN) merupakan metode deep learning yang digunakan untuk melakukan ekstraksi dan mempelajari fitur-fitur wajah. Penelitian ini akan melakukan pengenalan wajah bermasker menggunakan CNN dengan bantuan *Angular margin loss*. *Angular margin loss* memungkinkan untuk mendapatkan fitur yang sangat diskriminatif untuk pengenalan wajah. Dataset yang digunakan pada penelitian ini adalah *Real-World Masked Face Dataset* (RMFD). Berdasarkan hasil penelitian ini, penggunaan *angular margin loss* menunjukkan pengaruh terhadap model pengenalan wajah. Hasil penelitian ini mendapat nilai akurasi data latih lebih besar 36,33% dan akurasi data validasi lebih besar 12,65% dibandingkan model penelitian sebelumnya.

**Kata Kunci:** Pengenalan Wajah Bermasker, *Convolutional Neural Network*, *Angular Margin Loss*, *Real-World Masked Face Dataset*

## KATA PENGANTAR

Segala pujian dan syukur kepada Allah SWT atas karunia-Nya yang telah diberikan kepada penyusun sehingga dapat menyelesaikan penelitian skripsi berjudul “Pengenalan Citra Wajah Bermasker Menggunakan *CNN* dengan Pendekatan *Angular Margin Loss*” dengan baik.

Penyusun juga mengucapkan banyak terima kasih kepada pihak-pihak yang telah memberi dukungan dan membantu dalam penyelesaian skripsi ini, yaitu:

1. Orang tua dan saudara-saudara saya yang telah memberikan dukungan dan doa kepada saya.
2. Bapak Jaidan Jauhari, S.Pd., M.T. selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.
3. Ibu Alvi Syahrini Utami, M.Kom. selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya.
4. Bapak Dr. Muhammad Fachrurrozi, M.T., selaku Dosen Pembimbing I.
5. Bapak Bapak Muhammad Qurhanul Rizqie, M.T., Ph.D., selaku Dosen Pembimbing II sekaligus Dosen Pembimbing Akademik.
6. Bapak Alm. Drs. Megah Mulya, M.T. selaku Dosen Pembimbing Akademik dari Semester 1 sampai 3.
7. Seluruh Bapak dan Ibu Dosen Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya.
8. Seluruh Tata Usaha Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya.



9. Teman-teman Teknik Informatika Reguler A 2019 yang telah menemani dan membantu selama perkuliahan.
10. Teman-teman Dinas Akademik BEM Fasilkom UNSRI 2021.
11. Serta semua pihak yang telah membantu Penyusun selama penyusunan skripsi yang tidak dapat disebutkan satu-persatu.

Penyusun menyadari bahwa dalam penyusunan skripsi masih terdapat banyak kesalahan dan kekurangan yang disebabkan oleh keterbatasan pengetahuan dan pengalaman. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diperlukan untuk kemajuan penelitian di masa mendatang. Semoga penelitian ini dapat memberikan manfaat bagi kita semua.

Indralaya, 19 Desember 2022

Penyusun,

Ferza Reyaldi  
NIM 09021281924060

## DAFTAR ISI

### Halaman

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN SKRIPSI .....	ii
TANDA LULUS UJIAN KOMPREHENSIF SKRIPSI .....	iii
HALAMAN PERNYATAAN BEBAS PLAGIAT .....	iv
MOTTO DAN PERSEMBAHAN .....	v
ABSTRACT.....	vi
ABSTRAK.....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN.....	xvii
BAB I PENDAHULUAN.....	I-1
1.1    Pendahuluan .....	I-1
1.2    Latar Belakang Masalah .....	I-1
1.3    Rumusan Masalah .....	I-3
1.4    Tujuan Penelitian.....	I-3
1.5    Manfaat Penelitian.....	I-3
1.6    Batasan Masalah .....	I-4
1.7    Sistematika Penulisan .....	I-4
1.8    Kesimpulan.....	I-5
BAB II KAJIAN LITERATUR .....	II-1
2.1    Pendahuluan .....	II-1
2.2    Landasan Teori .....	II-1
2.2.1 Citra Digital .....	II-1
2.2.2 Pengenalan Wajah .....	II-2
2.2.3 Facial Landmark.....	II-3
2.2.4 Deep Learning .....	II-4
2.2.5 Convolutional Neural Network .....	II-4

2.2.6 Additive Angular Margin Loss .....	II-7
2.2.7 TensorFlow .....	II-8
2.2.8 Metrik Evaluasi .....	II-9
2.2.9 Rational Unified Process (RUP).....	II-11
2.3 Penelitian Lain yang Relevan .....	II-12
2.3.1 Mandal, B., Okeukwu, A., & Theis, Y. (2021). Masked Face Recognition using ResNet-50.....	II-13
2.3.2 Li, Y., Guo, K., Lu, Y., & Liu, L. (2021). Cropping and Attention Based Approach for Masked Face Recognition. ....	II-13
2.3.3 Hariri, W. (2021). Efficient Masked Face Recognition Method during The COVID-19 Pandemic. ....	II-14
2.4 Kesimpulan.....	II-14
BAB III METODOLOGI PENELITIAN .....	III-1
3.1 Pendahuluan .....	III-1
3.2 Pengumpulan Data.....	III-1
3.2.1 Jenis Data .....	III-1
3.2.2 Sumber Data .....	III-1
3.2.3 Metode Pengumpulan Data .....	III-2
3.3 Tahapan Penelitian .....	III-2
3.3.1 Akuisisi Data .....	III-3
3.3.2 Kerangka Kerja.....	III-3
3.3.3 Kriteria Pengujian.....	III-5
3.3.4 Format Data Pengujian .....	III-5
3.3.5 Alat yang Digunakan dalam Pengujian .....	III-6
3.3.6 Pengujian Penelitian .....	III-6
3.3.7 Analisis Hasil Pengujian .....	III-7
3.4 Metode Pengembangan Perangkat Lunak .....	III-7
3.4.1 Fase Insepsi .....	III-7
3.4.2 Fase Elaborasi.....	III-8
3.4.3 Fase Konstruksi .....	III-8
3.4.4 Fase Transisi .....	III-9
3.5 Manajemen Proyek Penelitian .....	III-9
3.6 Kesimpulan.....	III-12
BAB IV PENGEMBANGAN PERANGKAT LUNAK .....	IV-1
4.1 Pendahuluan .....	IV-1

4.2	Rational Unified Process .....	IV-1
4.2.1	Fase Insepsi .....	IV-1
4.2.2	Fase Elaborasi.....	IV-16
4.2.3	Fase Kontruksi.....	IV-22
4.2.4	Fase Transisi .....	IV-25
4.3	Kesimpulan.....	IV-28
BAB V HASIL DAN ANALISIS PENELITIAN .....		V-1
5.1	Pendahuluan .....	V-1
5.2	Data Hasil Penelitian .....	V-1
5.2.1	Data Hasil Konfigurasi Model 1.....	V-2
5.2.2	Data Hasil Konfigurasi Model 2.....	V-3
5.2.3	Data Hasil Konfigurasi Model 3.....	V-4
5.2.4	Data Hasil Konfigurasi Model 4.....	V-5
5.2.5	Data Hasil Konfigurasi Model 5.....	V-6
5.2.6	Data Hasil Konfigurasi Model 6.....	V-7
5.2.7	Data Hasil Konfigurasi Model 7.....	V-8
5.2.8	Data Hasil Konfigurasi Model 8.....	V-9
5.3	Analisis Hasil Penelitian.....	V-10
5.4	Kesimpulan.....	V-12
BAB VI KESIMPULAN DAN SARAN .....		VI-1
6.1	Kesimpulan.....	VI-1
6.2	Saran .....	VI-1
DAFTAR PUSTAKA .....		vii
LAMPIRAN.....		xxii

## DAFTAR TABEL

Halaman

<b>Tabel III-1.</b> Tabel Evaluasi Model CNN dengan <i>Additive Angular Margin Loss</i> .....	III-6
<b>Tabel III-2.</b> WBS Penelitian Pengenalan Citra Wajah Bermasker Menggunakan CNN dengan Teknik <i>Additive Angular Margin Loss</i> .....	III-9
<b>Tabel IV-1.</b> Kebutuhan Fungsional Perangkat Lunak.....	IV-2
<b>Tabel IV-2.</b> Kebutuhan Non-Fungsional Perangkat Lunak .....	IV-2
<b>Tabel IV-3.</b> Sampel Citra Wajah Bermasker .....	IV-3
<b>Tabel IV-4.</b> Sampel Citra Wajah Tidak Bermasker .....	IV-7
<b>Tabel IV-5.</b> Konfigurasi Parameter Augmentasi Data .....	IV-10
<b>Tabel IV-6.</b> Pembagian Data .....	IV-11
<b>Tabel IV-7.</b> Skenario-Skenario Percobaan.....	IV-11
<b>Tabel IV-8.</b> Konfigurasi Parameter Tetap .....	IV-12
<b>Tabel IV-9.</b> Definisi Aktor .....	IV-13
<b>Tabel IV-10.</b> Definisi <i>Use Case</i> .....	IV-14
<b>Tabel IV-11.</b> Skenario <i>Use Case</i> Masukan Gambar Wajah.....	IV-14
<b>Tabel IV-12.</b> Skenario <i>Use Case</i> Mendeteksi dan Mengidentifikasi Citra Wajah .....	IV-15
<b>Tabel IV-13.</b> Keterangan Implementasi Kelas .....	IV-23
<b>Tabel IV-14.</b> Daftar File HTML .....	IV-24
<b>Tabel IV-15.</b> Rencana Pengujian <i>Use Case</i> Masukan Gambar Wajah .....	IV-26
<b>Tabel IV-16.</b> Rencana Pengujian <i>Use Case</i> Deteksi dan Identifikasi Wajah ..	IV-26
<b>Tabel IV-17.</b> Pengujian <i>Use Case</i> Masukan Gambar/Video Wajah .....	IV-27
<b>Tabel IV-18.</b> Pengujian <i>Use Case</i> Deteksi dan Identifikasi Wajah .....	IV-27
<b>Tabel V-1.</b> Daftar Variasi Konfigurasi Model .....	V-1
<b>Tabel V-2.</b> Hasil <i>Testing</i> Model 1 .....	V-3
<b>Tabel V-3.</b> Hasil <i>Testing</i> Model 2 .....	V-4
<b>Tabel V-4.</b> Hasil <i>Testing</i> Model 3 .....	V-5
<b>Tabel V-5.</b> Hasil <i>Testing</i> Model 4 .....	V-6

<b>Tabel V-6.</b> Hasil <i>Testing</i> Model 5 .....	V-7
<b>Tabel V-7.</b> Hasil <i>Testing</i> Model 6 .....	V-8
<b>Tabel V-8.</b> Hasil <i>Testing</i> Model 7 .....	V-9
<b>Tabel V-9.</b> Hasil <i>Testing</i> Model 8 .....	V-10
<b>Tabel V-10.</b> Perbandingan Hasil <i>Training-Validation</i> Model.....	V-10
<b>Tabel V-11.</b> Perbandingan Hasil <i>Testing</i> Model .....	V-11

## DAFTAR GAMBAR

	Halaman
<b>Gambar II-1.</b> Representasi Elemen Citra Digital .....	II-2
<b>Gambar II-2.</b> Facial Landmark.....	II-3
<b>Gambar II-3.</b> Sampel wajah yang telah teranota <i>facial landmark</i> .....	II-3
<b>Gambar II-4.</b> Contoh proses kerja CNN .....	II-5
<b>Gambar II-5.</b> Arsitektur Model ResNet-50 .....	II-6
<b>Gambar II-6.</b> Arsitektur Model InceptionV3 .....	II-7
<b>Gambar II-7.</b> Confusion matrix.....	II-9
<b>Gambar II-8.</b> Fase-fase RUP .....	II-12
<b>Gambar III-1.</b> Tahapan-tahapan penelitian .....	III-2
<b>Gambar III-2.</b> Diagram Alir Klasifikasi Citra Wajah Bermasker Menggunakan CNN dengan Teknik <i>Additive Angular Margin Loss</i> .....	III-3
<b>Gambar IV-1.</b> <i>Use Case Diagram</i> Perangkat Lunak.....	IV-13
<b>Gambar IV-2.</b> Rancangan Tampilan Halaman Depan.....	IV-17
<b>Gambar IV-3.</b> Rancangan Tampilan Halaman Pengenalan Wajah .....	IV-18
<b>Gambar IV-4.</b> Rancangan Tampilan Halaman Pengenalan Wajah Ketika Menampilkan Hasil Identifikasi.....	IV-18
<b>Gambar IV-5.</b> <i>Activity Diagram</i> Masukan Gambar Wajah.....	IV-19
<b>Gambar IV-6.</b> <i>Activity Diagram</i> Deteksi dan Identifikasi Citra Wajah .....	IV-20
<b>Gambar IV-7.</b> <i>Sequence Diagram</i> Masukan Gambar Wajah .....	IV-21
<b>Gambar IV-8.</b> <i>Sequence Diagram</i> Deteksi dan Identifikasi Citra Wajah .....	IV-21
<b>Gambar IV-9.</b> Diagram Kelas .....	IV-22
<b>Gambar IV-10.</b> Antarmuka Halaman Depan.....	IV-24
<b>Gambar IV-11.</b> Antarmuka Halaman Pengenalan Wajah .....	IV-25
<b>Gambar IV-12.</b> Antarmuka Halaman Pengenalan Wajah setelah Ditekan Tombol <i>Predict</i> .....	IV-25
<b>Gambar V-1.</b> Grafik Akurasi Model 1 .....	V-2
<b>Gambar V-2.</b> Grafik Akurasi Model 2 .....	V-3
<b>Gambar V-3.</b> Grafik Akurasi Model 3 .....	V-4

<b>Gambar V-4.</b> Grafik Akurasi Model 4 .....	V-5
<b>Gambar V-5.</b> Grafik Akurasi Model 5 .....	V-6
<b>Gambar V-6.</b> Grafik Akurasi Model 6 .....	V-7
<b>Gambar V-7.</b> Grafik Akurasi Model 7 .....	V-8
<b>Gambar V-8.</b> Grafik Akurasi Model 8 .....	V-9



## **DAFTAR LAMPIRAN**

<b>Lampiran 1.</b> Kode Program.....	xxii
--------------------------------------	------

# **BAB I**

## **PENDAHULUAN**

### **1.1 Pendahuluan**

Bab I membahas latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, sistematika penulisan dan kesimpulan berdasarkan penelitian yang diajukan secara rinci.

### **1.2 Latar Belakang Masalah**

Pengenalan wajah adalah salah satu solusi pengenalan biometrik yang lebih disukai karena tidak memerlukan kontak langsung dengan pengguna dan tingkat akurasi yang dicapai tinggi. Namun, wajib pemakaian masker wajah mulai diberlakukan di tempat-tempat umum saat Pandemi Covid-19 yang bertujuan untuk menjaga agar pandemi tetap terkendali (Boutros et al., 2021). Hal tersebut menyebabkan oklusi parsial pada wajah dikarenakan pemakaian aksesoris pakaian berupa masker tidak dapat dihindari (Montero et al., 2021). Masalah khusus ini menjadi tantangan utama di bidang pengenalan wajah dikarenakan fitur-fitur wajah yang tersedia berkurang (Hariri, 2022). Hal ini dikarenakan pengenalan wajah memainkan peran penting dalam kehidupan sehari-hari seperti pemeriksaan paspor, ATM, kartu kredit, verifikasi pemilih (*voter verification*), pintu pintar, investigasi kriminal atau teroris dan banyak tujuan lainnya (Ejaz et al., 2019).

Metode-metode seperti *Principal Component Analysis* (Ejaz et al., 2019) dan *Convolutional Neural Network* (Mandal et al., 2021) diusulkan oleh peneliti untuk melakukan ekstraksi fitur pada citra wajah.

*Principal Component Analysis* (PCA) adalah prosedur statistik yang juga didefinisikan sebagai transformasi linier ortogonal. Algoritma ini menekankan variasi dan memunculkan pola yang kuat dalam kumpulan data. Ini digunakan untuk meminimalkan dataset besar ke dataset kecil masih berisi hampir semua informasi sebagai dataset besar. Namun, pengenalan wajah memiliki tingkat pengenalan yang buruk dikarenakan fitur dari wajah bermasker lebih sedikit menyebabkan turunnya tingkat pengenalan (Ejaz et al., 2019).

*Convolutional Neural Network* (CNN) merupakan arsitektur *deep learning* yang tersusun atas banyak layer, seperti *input layer*, *convolution layer*, *pooling layer*, dan *fully connected layer* (Xiong et al., 2017). Arsitektur CNN cocok digunakan untuk masalah klasifikasi karena CNN dapat melakukan ekstraksi dan mempelajari fitur - fitur pada data secara otomatis (Chandra et al., 2017). Namun arsitektur CNN membutuhkan daya komputasi yang cukup tinggi dikarenakan penggunaan filter berupa matriks 2 dimensi dalam ekstraksi fitur pada data (Nurmaini et al., 2020).

*Angular Margin Loss* adalah *loss function* yang memungkinkan untuk mendapatkan fitur yang sangat diskriminatif untuk pengenalan wajah. *Angular Margin Loss* memiliki interpretasi geometris yang jelas karena korespondensi yang tepat dengan jarak geodesik pada *hypersphere* (Deng et al., n.d.).

Pada penelitian ini, *Angular Margin Loss* diimplementasikan pada arsitektur CNN yang akan dilatih untuk meningkatkan tingkat diskriminatif fitur-fitur pada citra wajah bermasker yang terbatas.

### 1.3 Rumusan Masalah

Rumusan masalah dari penelitian yang dilakukan adalah sebagai berikut:

1. Bagaimana mengembangkan perangkat lunak pengenalan citra wajah bermasker menggunakan arsitektur CNN?
2. Bagaimana Performa model identifikasi wajah bermasker pada arsitektur CNN yang akan dibangun?

### 1.4 Tujuan Penelitian

Berdasarkan rumusan masalah, tujuan dari penelitian yang dilakukan adalah sebagai berikut:

1. Membangun model pengenalan citra wajah bermasker menggunakan algoritma CNN dengan pendekatan *Angular Margin Loss*.
2. Mengukur dan mengetahui akurasi model pengenalan citra wajah bermasker menggunakan algoritma CNN dengan pendekatan *Angular Margin Loss*.

### 1.5 Manfaat Penelitian

Penelitian yang dijalankan diharapkan memberi manfaat sebagai berikut:

1. Penelitian ini dapat membantu dalam pengembangan model identifikasi citra wajah bermasker yang lebih baik.
2. Penelitian ini dapat menjadi rujukan dalam penelitian lebih lanjut mengenai pengenalan wajah bermasker menggunakan algoritma CNN.

### **1.6 Batasan Masalah**

Untuk mencegah penelitian ini terlalu meluas dan tidak terarah, peneliti membatasi lingkup masalah pada penelitian ini dengan rincian sebagai berikut:

1. Dataset yang digunakan untuk melatih model CNN dimensi adalah *Real-World Masked Face Dataset (RMFD)*.

### **1.7 Sistematika Penulisan**

Sistematika penulisan skripsi ini adalah sebagai berikut:

## **BAB I. PENDAHULUAN**

Bab ini membahas secara rinci tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, sistematika penulisan, dan kesimpulan.

## **BAB II. KAJIAN LITERATUR**

Bab ini membahas secara rinci mengenai penelitian – penelitian lain yang relevan dan landasan teori yang menjadi dasar dalam menyusun penelitian ini.

## **BAB III. METODOLOGI PENELITIAN**

Bab ini membahas secara rinci mengenai kerangka kerja, instrumen penelitian, data yang digunakan dalam penelitian, dan perencanaan dari kegiatan – kegiatan penelitian.

#### **BAB IV. PENGEMBANGAN PERANGKAT LUNAK**

Bab ini membahas secara rinci mengenai proses pengembangan perangkat lunak yang sudah direncanakan pada BAB III, dan melakukan pengujian pada perangkat lunak yang digunakan untuk penelitian.

#### **BAB V. HASIL DAN ANALISIS PENELITIAN**

Bab ini membahas secara rinci mengenai hasil dari perangkat lunak yang digunakan pada penelitian dan melakukan analisa pada hasil tersebut.

#### **BAB VI. KESIMPULAN DAN SARAN**

Bab ini membahas secara rinci mengenai kesimpulan yang dapat ditarik dari hasil penelitian yang telah dilakukan serta saran – saran yang dapat digunakan untuk mengembangkan penelitian tersebut.

### **1.8 Kesimpulan**

Bab I menjelaskan terkait latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, Batasan masalah dan sistematika penulisan. Berdasarkan penjelasan diatas, penelitian dalam pengenalan citra wajah bermasker

menggunakan CNN dengan teknik *Angular Margin Loss* diharapkan memberikan hasil yang baik sesuai hipotesis.

## **BAB II**

### **KAJIAN LITERATUR**

#### **2.1 Pendahuluan**

Bab ini berisi landasan teori yang berhubungan dengan masalah yang diteliti, penelitian sebelumnya yang berkaitan dengan penelitian yang diajukan dalam beberapa tahun terakhir, dan hipotesis yang diajukan.

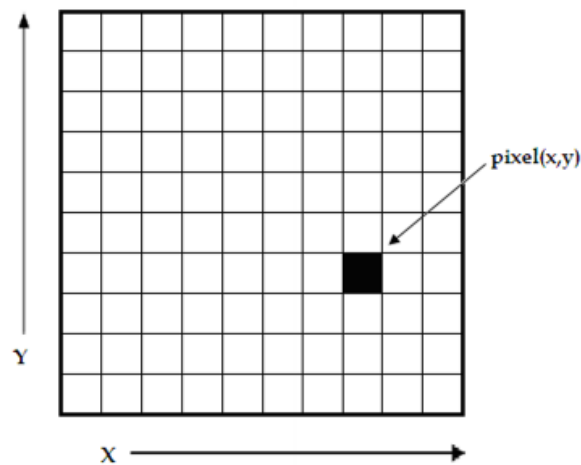
#### **2.2 Landasan Teori**

Bab ini berisi mengenai dasar-dasar teori yang berhubungan dengan pengenalan citra wajah Bermasker menggunakan CNN dan pendekatan *Angular Margin Loss*. Bab ini juga menjelaskan penelitian lain mengenai metode yang pernah digunakan dalam beberapa tahun terakhir serta membahas beberapa dasar teori yang mendukung penelitian ini.

##### **2.2.1 Citra Digital**

Citra adalah representasi visual dari suatu objek. Citra digital adalah proyeksi citra tiga dimensi ke dalam bidang dua dimensi. Citra digital direpresentasikan sebagai fungsi dua dimensi  $f(x,y)$ , dimana  $x$  dan  $y$  mewakili lokasi elemen citra atau piksel (*pixel*) dan memiliki intensitas nilai yang diskrit. Secara matematis, citra digital digambarkan sebagai representasi matriks dari citra dua dimensi menggunakan sejumlah elemen sel titik yang terbatas yang disebut *pixel* (Tyagi, 2018).





**Gambar II-1.** Representasi Elemen Citra Digital

### 2.2.2 Pengenalan Wajah

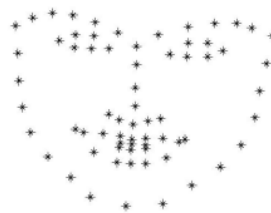
Pengenalan wajah (*face recognition*) merupakan subdivisi masalah dari pengenalan pola visual. Teknologi pengenalan wajah merupakan teknologi biometrik yang melakukan identifikasi seseorang berdasarkan fitur-fitur wajahnya (*facial features*) (L. Li et al., 2020). Teknologi ini banyak disukai dikarenakan pengenalan wajah tanpa memerlukan kontak langsung dengan pengguna dan memiliki tingkat akurasi tinggi (Boutros et al., 2021).

Sistem pengenalan wajah mencakup deteksi wajah (*face detection*), posisi wajah (*face position*), dan pengenalan identitas (*identity recognition*). Algoritma deteksi wajah adalah untuk mengetahui sistem koordinat semua wajah dalam satu citra. Algoritma ini melakukan pemindaian seluruh citra untuk menentukan apakah area kandidat adalah wajah. Keluaran dari deteksi wajah dapat berbentuk persegi, persegi panjang, dll. Posisi wajah adalah posisi koordinat fitur wajah dalam sistem koordinat deteksi wajah. Dibandingkan dengan deteksi wajah, waktu perhitungan algoritma penentuan posisi wajah jauh lebih singkat (L. Li et al., 2020).

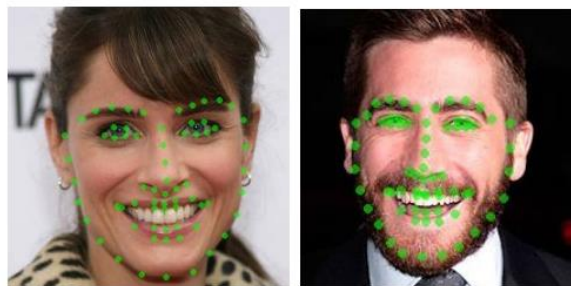
Pada penelitian ini, penulis menggunakan salah satu algoritma *deep learning* dalam membangun model klasifikasi citra wajah bermasker dengan bantuan TensorFlow sebagai *library* utama.

### 2.2.3 Facial Landmark

Dalam visi komputer, wajah manusia dikenali melalui tengara wajah (*facial landmark*). *Facial landmark* mendefinisikan bentuk wajah. *Facial landmark detection* biasanya dilakukan melalui proses pelokalan titik kunci wajah *fiducial* (fidusia) (Wu & Ji, 2019). Titik kunci wajah tersebut antara lain sudut mata, ujung hidung, bibir, dll (Zhang et al., 2018).



**Gambar II-2.** Facial Landmark



**Gambar II-3.** Sampel wajah yang telah teranota *facial landmark*

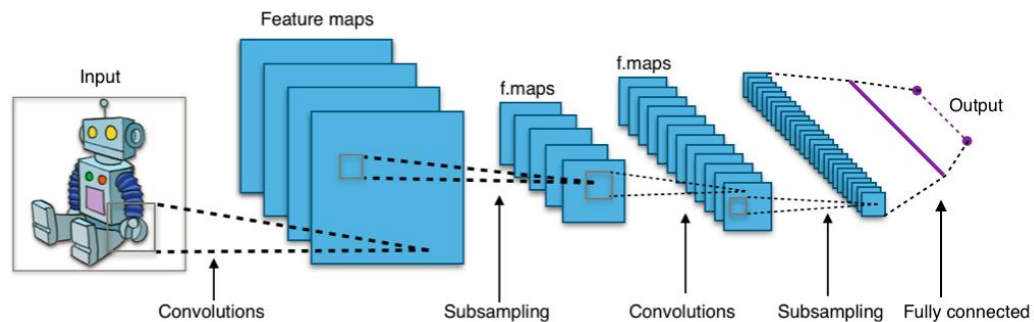
### 2.2.4 Deep Learning

Pembelajaran mendalam (*deep learning*) adalah cabang ilmu dari pembelajaran mesin (*machine learning*). *Deep learning* dapat mengetahui fitur-fitur data melalui pembelajaran untuk permasalahan klasifikasi secara otomatis tanpa perlu melalui tahap ekstraksi fitur secara spesifik (L. Li et al., 2020).

*Deep learning* telah menunjukkan tingkat akurasi superior pada sistem pengenalan wajah (Anwar & Raychowdhury, 2020), sehingga *deep learning* disebut algoritma *state-of-the-art* (termutakhir) dari sistem pengenalan wajah. *Deep learning* ini belajar untuk mengekstrak fitur-fitur penting dari citra wajah dan menanamkannya ke dalam vektor n-dimensi dengan jarak antar kelas yang kecil dan jarak antar kelas yang besar (Montero et al., 2021).

### 2.2.5 Convolutional Neural Network

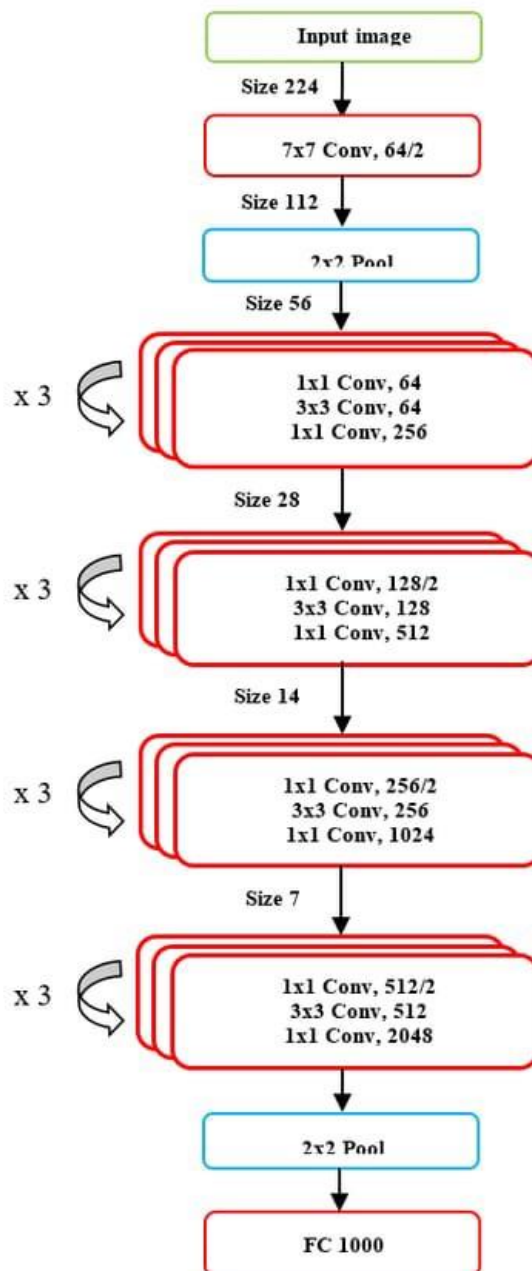
*Convolutional Neural Network* (CNN) adalah salah satu jenis jaringan saraf tiruan yang paling efektif. CNN telah banyak menunjukkan keunggulannya dalam berbagai aplikasi seperti klasifikasi gambar (*image classification*), pengenalan objek (*object recognition*), pengambilan objek (*object retrieval*), dan deteksi objek (*object detection*). CNN bisa terdiri atas banyak lapisan berjenjang (*cascaded layer*). Lapisan-lapisan berjenjang tersebut berfungsi untuk mengontrol tingkat pergeseran, skala, dan distorsi. Jenis lapisan-lapisan berjenjang tersebut antara lain *input layer*, *convolutional layer*, *subsampling layer*, *full-connected layer*, dan *output layer* (Alzu'bi et al., 2021). Berikut pada Gambar II-4 merupakan contoh proses kerja dari CNN.



**Gambar II-4.** Contoh proses kerja CNN

Pada penerapannya, terdapat banyak model pra-latih CNN yang dapat digunakan untuk menyelesaikan permasalahan klasifikasi. *AlexNet*, *AlexNetOWTbn*, *GooLeNet*, *Overfeat*, *VGG* merupakan sekian contoh arsitektur model CNN yang umum digunakan. Arsitektur-arsitektur tersebut menggunakan banyak lapisan konvolusi. Namun, hal tersebut menyebabkan muncul masalah baru, seperti sulitnya optimasi jaringan, masalah gradien menghilang, dan masalah degradasi (Mukti & Biswas, 2019).

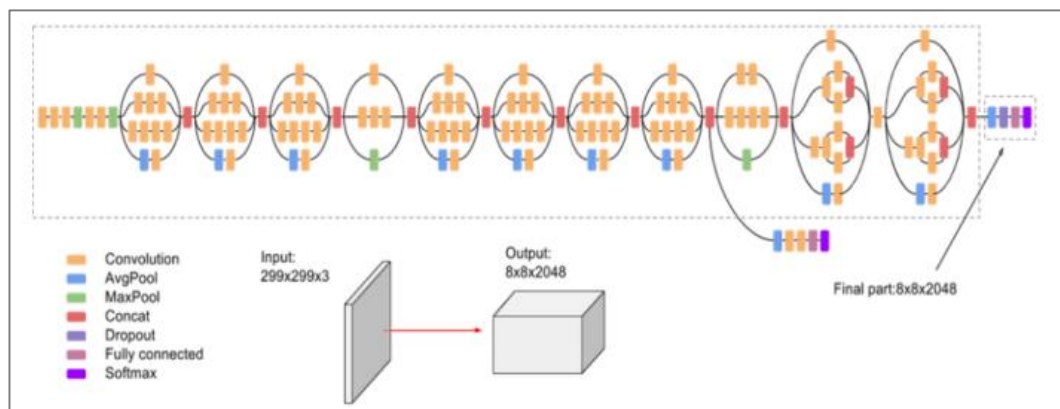
Ide baru yang dapat ditawarkan untuk permasalahan tersebut adalah *Residual Network* (ResNet). ResNet memiliki kelebihan untuk menyelesaikan tugas-tugas rumit dan juga meningkatkan akurasi deteksi. ResNet mencoba mengatasi kesulitan dalam proses pelatihan CNN yang dalam, kejenuhan dan penurunan akurasi. Dalam penelitian, penulis menggunakan arsitektur ResNet-50. Sesuai dengan namanya, ResNet-50 menggunakan 50 lapisan residual (*residual layer*). Pada Gambar II-5 menunjukkan arsitektur model CNN ResNet-50 (Mukti & Biswas, 2019).



**Gambar II-5.** Arsitektur Model ResNet-50

Selain ResNet, Ide baru yang dapat ditawarkan adalah InceptionV3. Model Inception-v3 adalah model pra-latih yang memiliki performa superior dalam pengenalan objek jika dibandingkan dengan versi pendahulunya, GoogleNet

(InceptionV1). Model InceptionV3 mencakup tiga bagian: blok konvolusional dasar, modul Inception yang ditingkatkan, dan *classifier* (Lin et al., 2019). Pada Gambar II-6 menunjukkan arsitektur model CNN InceptionV3 (Khan et al., 2020).



**Gambar II-6.** Arsitektur Model InceptionV3

### 2.2.6 Additive Angular Margin Loss

*Additive Angular Margin Loss* atau yang disebut juga dengan ArcFace, adalah salah satu jenis *loss function* yang dapat digunakan dalam membangun model *Deep Convolutional Neural Network* (DCNN). Keuntungan-keuntungan penggunaan ArcFace sebagai *loss function* dapat dirangkum atas empat hal, yaitu *engaging*, *effective*, *easy*, dan *efficient* (Deng et al., n.d.).

- *Engaging*, ArcFace secara langsung mengoptimalkan jarak batas geodesik (*geodesic distance margin*)
- *Effective*, ArcFace mencapai kinerja mutakhir pada sepuluh tolok ukur pengenalan wajah termasuk set data gambar dan video skala besar.
- *Easy*, ArcFace hanya membutuhkan beberapa baris dan mudah diimplementasikan pada *deep learning* berbasis grafik komputasi.

- *Efficient*, ArcFace hanya menambahkan kompleksitas komputasi yang dapat diabaikan selama pelatihan.

Secara matematis, ArcFace dapat dirumuskan seperti rumus II-1.

$$L_3 = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s(\cos(\theta_{y_i} + m))}}{e^{s(\cos(\theta_{y_i} + m))} + \sum_{j=1, j \neq y_i}^n e^{s \cos \theta_j}}. \quad (\text{II-1})$$

Keterangan:

$L_3$  : *Additive Angular Margin Loss*

$N$  : *batch siz*

$m$  : *angular margin penalty*

$y_i$  : kelas ke- $y_i$

### 2.2.7 TensorFlow

TensorFlow merupakan *open-source library* pengembangan model jaringan syaraf tiruan yang dikembangkan oleh Google. TensorFlow bersifat fleksibel dan *scalable* untuk komputasi numerik menggunakan grafik aliran data. Pustaka ini memungkinkan pengguna memprogram dan melatih jaringan saraf dan model pembelajaran mesin lainnya secara efisien dan men-*deploy*-nya ke tahap produksi (Pang et al., 2020).

TensorFlow memungkinkan membangun model dalam tiga cara, yaitu model sekuensial (*sequential model*), *functional API*, dan model pra-latih (*pre-trained model*) (Sanchez et al., 2020).

### 2.2.8 Metrik Evaluasi

Evaluasi model klasifikasi dapat dilakukan menggunakan *Confusion matrix*. *Confusion matrix* digunakan sebagai alat visualisasi evaluasi model pada permasalahan *supervised learning*. Kolom matriks mewakili kelas prediksi (*predicted class*) dan baris mewakili kelas sebenarnya (*actual class*) (Novandya, 2017).

Pada Gambar II-7 menunjukkan bentuk dari *confusion matrix*.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

**Gambar II-7.** Confusion matrix

*Confusion matrix* terdiri dari 4 kategori nilai, yaitu:

- *True Positive* (TP)

Model memprediksi kelas dengan benar, prediksi positif dan nilai sebenarnya positif.

- *True Negative* (TN)

Model memprediksi kelas dengan benar, prediksi negatif dan nilai sebenarnya negatif.



- *False Positive (FP)*

Model salah prediksi kelas, prediksi positif dan nilai sebenarnya negatif.

- *False Negative (FN)*

Model salah prediksi kelas, prediksi negatif dan nilai sebenarnya positif.

Perhitungan *confusion matrix* menghasilkan empat keluaran yang dapat dijadikan tolak ukur mengukur performa model, yaitu *accuracy*, *recall*, *precision*, dan *F-1 Score* (Saputro & Sari, 2020). Pada penelitian yang dilakukan, penulis menggunakan keempat metrik tersebut.

- *Accuracy*

*Accuracy* merupakan jumlah model benar memprediksi kelas (*true positive* dan *true negative*) dibagi dengan total semua hasil prediksi.

Secara matematis, *Accuracy* dirumuskan seperti pada rumus II-2.

$$acc = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (II-2)$$

- *Recall*

*Recall* merupakan perhitungan dari kondisi ketika kelas aktualnya positif, seberapa sering model memprediksi positif. Secara matematis, *recall* dirumuskan seperti pada rumus II-3.

$$r = \frac{TP}{FN+TP} \times 100\% \quad (II-3)$$

- *Precision*

*Precision* merupakan hasil perhitungan dari ketika model memprediksi positif, seberapa sering prediksi tersebut bernilai benar. Secara matematis, *precision* dirumuskan seperti pada rumus II-4.

$$p = \frac{TP}{FP+TP} \times 100\% \quad (\text{II-4})$$

- *F-1 Score*

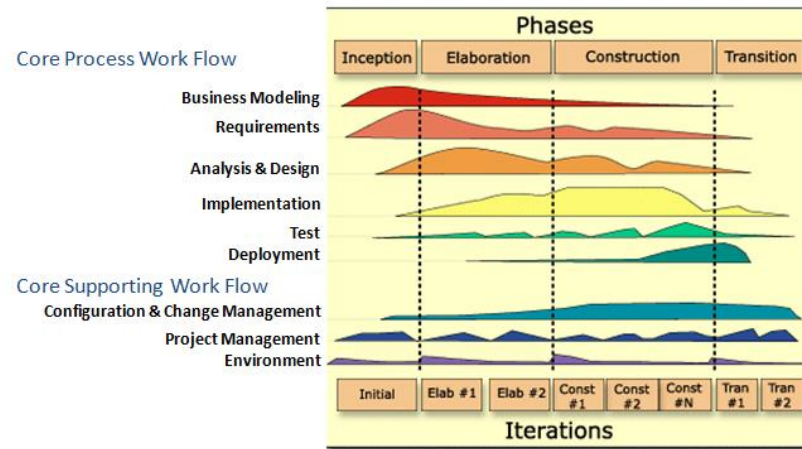
*F-1 Score* merupakan nilai rata-rata harmonic dari *precision* dan *recall*.

Secara matematis, *F-1 Score* dirumuskan seperti pada rumus II-5.

$$FM = 2 \times \frac{p \times r}{p + r} \times 100\% \quad (\text{II-5})$$

### 2.2.9 Rational Unified Process (RUP)

*Rational Unified Process* (RUP) adalah metode pengembangan perangkat lunak yang digunakan dalam proses pengembangan sistem pengenalan citra wajah bermasker. RUP adalah metode pengembangan perangkat lunak yang bersifat *architecture-centric* dan berorientasi *use case driven*. RUP dilakukan secara berulang-ulang (*iterative*). Dalam penerapannya, setiap iterasi RUP terbagi atas 4 fase pengembangan sistem, yaitu fase insepisi, fase elaborasi, fase konstruksi, dan fase transisi (Hakim & Rizky, 2020). Visualisasi fase-fase dalam RUP dapat dilihat pada Gambar II-8.



**Gambar II-8.** Fase-fase RUP

Dalam setiap iterasi, serangkaian aktivitas dilaksanakan. Aktivitas-aktivitas tersebut disebut disiplin (*disciplines*). Disiplin tersebut dibagi atas dua jenis, yaitu disiplin utama dan disiplin pendukung. Disiplin utama antara lain *business modelling*, *requirements*, *analysis and design*, *implementation*, dan *testing and deployment*. Sedangkan disiplin pendukung adalah manajemen konfigurasi dan perubahan, manajemen proyek, serta lingkungan. Secara umum, mayoritas aktivitas hanya dilakukan pada satu fase (Shafiee et al., 2020).

### 2.3 Penelitian Lain yang Relevan

Dalam proses penyusunan skripsi, terdapat beberapa-beberapa referensi penelitian-penelitian sebelumnya yang digunakan penulis sebagai pendukung dan penunjang proses penelitian, diantaranya:

### **2.3.1 Mandal, B., Okeukwu, A., & Theis, Y. (2021). Masked Face Recognition using ResNet-50.**

Penelitian pengenalan wajah bermasker pernah dilakukan oleh (Mandal et al., 2021) menggunakan salah satu arsitektur model pra-latih CNN, yaitu ResNet-50. Penelitian dilakukan bertujuan untuk menjawab tantangan ditengah Pandemi Covid-19 untuk membangun sistem pengenalan wajah yang dapat mengenali citra wajah bermasker. Dataset yang digunakan dalam penelitian ini adalah RMFD. Pengujian hasil penelitian menggunakan metrik akurasi dengan melakukan perbandingan terhadap model pengenalan wajah bermasker dan model pengenalan wajah tidak bermasker. Hasil penelitian menunjukkan bahwa tingkat akurasi dari model pengenalan wajah bermasker adalah 47,9%, sedangkan tingkat akurasi dari model pengenalan wajah tidak bermasker adalah 89,7%.

### **2.3.2 Li, Y., Guo, K., Lu, Y., & Liu, L. (2021). Cropping and Attention Based Approach for Masked Face Recognition.**

Topik pengenalan wajah bermasker pernah diteliti oleh (Y. Li et al., 2021) dengan menggunakan pendekatan *cropping-based* yang diintegrasikan dengan *Convolutional Block Attention Module* (CBAM). Pendekatan *cropping-based* digunakan penulis bertujuan untuk mencari titik pemotongan optimal (*optimal cropping*) dari masing-masing studi kasus yang diteliti. Penulis menggunakan 4 dataset untuk membangun model pengenalan wajah, yaitu *Simulated Masked Face Recognition Dataset* (SMFRD), *Webface*, *AR*, dan *Extend Yela B*. Pengujian pendekatan diajukan menggunakan metrik akurasi yang dibagi atas 4 studi kasus berbeda, yaitu *case 1*, *case 2*, *case 3*, *case 4*. Kemudian, hasil penelitian

dibandingkan dengan tingkat akurasi penelitian-penelitian terdahulu. Tingkat akurasi dari metode yang diajukan penulis pada masing-masing studi kasus antara lain 91,5%, 86,9%, 81,4%, dan 92,6%.

### **2.3.3 Hariri, W. (2021). Efficient Masked Face Recognition Method during The COVID-19 Pandemic.**

Penelitian (Hariri, 2022) menggunakan CNN dengan paradigma *Bag-of-Features* (BoF) untuk membangun model pengenalan wajah bermasker yang efisien. Penulis melaksanakan penelitian ini bertujuan untuk mengatasi masalah tingginya penggunaan memori dan kapasitas pemrosesan. Dalam proses penelitian, 3 arsitektur model CNN pra-latih digunakan, yaitu VGG-16, AlexNet, dan ResNet-50. Dua dataset digunakan pada penelitian, antara lain *Real-world Masked Face Recognition Dataset* (RMFRD) dan *Simulated Masked Face Recognition Dataset* (SMFRD). Pengujian hasil penelitian dilakukan menggunakan metrik akurasi pada masing-masing dataset. Hasil penelitian menunjukkan bahwa tingkat akurasi dari metode yang diajukan 91,3% pada dataset RMFRD dan 88,9% pada dataset SMFRD.

## **2.4 Kesimpulan**

Bab kajian literatur telah menjabarkan secara rinci terkait dasar-dasar teori dan penelitian-penelitian sebelumnya yang relevan untuk mendukung proses penelitian.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Pendahuluan**

Bab ini menguraikan setiap tahapan penelitian yang dilakukan dalam proses pengembangan perangkat lunak, mulai dari tahap pengumpulan data hingga manajemen proyek penelitian. Tahapan penelitian ini dijadikan acuan pada setiap tahap pengembangan perangkat lunak pengenalan citra wajah bermasker menggunakan CNN dengan teknik *Additive Angular Margin Loss*.

#### **3.2 Pengumpulan Data**

##### **3.2.1 Jenis Data**

Jenis dataset yang digunakan pada penelitian model klasifikasi citra wajah bermasker adalah data sekunder. Dataset tersebut adalah *Real-World Masked Face Dataset* (RMFD). RMFD terdiri dari 525 subjek (kelas) yang mencakup 90.000 citra wajah tanpa masker dan 5.000 citra wajah bermasker. Selain itu, Ukuran citra gambar yang tersedia pada dataset yang digunakan bervariasi.

##### **3.2.2 Sumber Data**

RMFD digunakan pada penelitian model klasifikasi citra wajah bermasker bersumber dari *National Engineering Research Center for Multimedia Software* (NERCMS), *School of Computer Science, Wuhan University*. Dataset tersebut

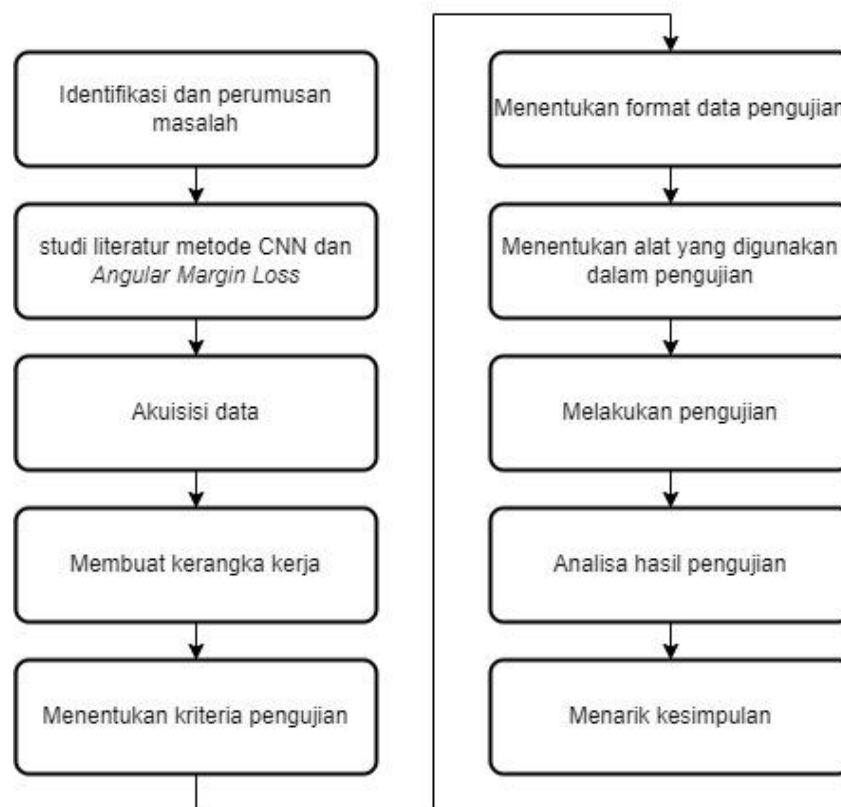
dapat diakses langsung secara publik melalui laman Kaggle, yaitu <https://www.kaggle.com/datasets/muhammeddalkran/masked-facerecognition>.

### 3.2.3 Metode Pengumpulan Data

Dalam penelitian ini, data dikumpulkan dengan cara mengunduh langsung dataset dari sumber data.

### 3.3 Tahapan Penelitian

Penelitian ini dilakukan berdasarkan tahapan-tahapan penelitian. Adapun tahapan penelitian dapat dilihat pada Gambar III-1.



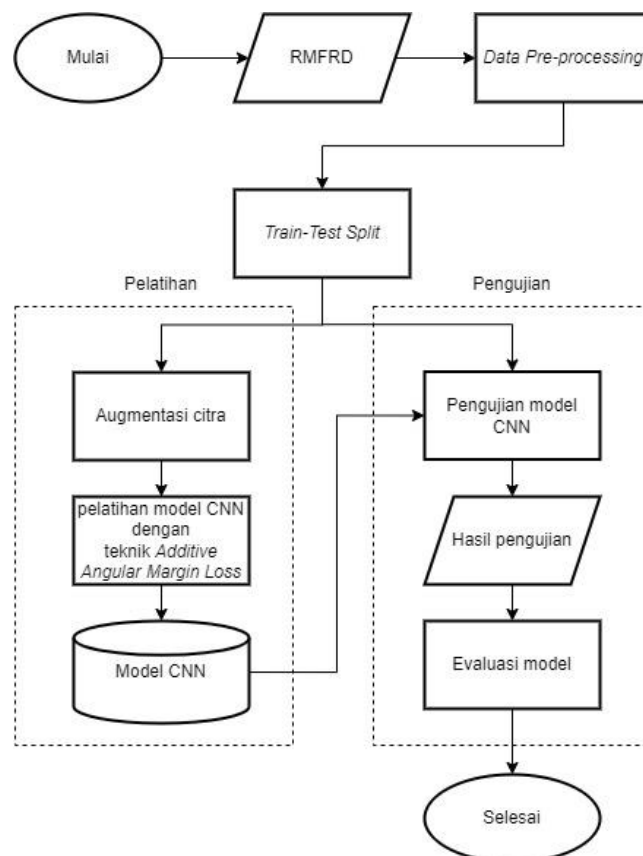
**Gambar III-1.** Tahapan-tahapan penelitian

### 3.3.1 Akuisisi Data

Pada tahapan akuisisi data, penulis mengambil dataset yang diperlukan sesuai dengan yang dijelaskan pada subbab 3.2. Dataset tersebut disimpan penulis di komputer dan penyimpanan awan, yaitu *Google Drive* pada akun pribadi pengguna.

### 3.3.2 Kerangka Kerja

Kerangka kerja penelitian model klasifikasi pengenalan citra wajah bermasker menggunakan CNN dengan teknik *Additive Angular Margin Loss* mengacu pada Gambar III-2.



**Gambar III-2.** Diagram Alir Klasifikasi Citra Wajah Bermasker Menggunakan CNN dengan Teknik *Additive Angular Margin Loss*



### **3.3.1.1 Data Pre-Processing**

*Data pre-processing* merupakan suatu proses mengolah data menjadi lebih efektif dan sesuai dengan kebutuhan pengembangan model pembelajaran. Pada proses ini, dilakukan *resizing image* dan *rescale* terhadap dataset citra wajah yang digunakan pada penelitian. Hal ini dilakukan bertujuan untuk mempercepat pemrosesan citra pada tahapan selanjutnya dan mengurangi sumber daya perangkat keras yang dipakai.

### **3.3.1.2 Train-Test Split**

Pada proses *Train-test split*, dataset dibagi menjadi 2 bagian yaitu data latih dan data uji. Perbandingan banyak data latih dan data uji sebesar 4:1, 80% untuk data latih dan 20% untuk data uji. 2 bagian data tersebut masing-masing secara berurutan digunakan untuk proses pelatihan model dan pengujian model.

### **3.3.1.3 Augmentasi Citra Wajah Data Latih**

Pada data latih yang telah dipisahkan dengan data uji, diterapkan augmentasi data. Hal ini dilakukan bertujuan untuk memperbanyak data latih dan meningkatkan variasi data latih. Augmentasi data yang diterapkan antara lain: *rotation*, *horizontal flipping*, *shifting*, dan *zooming*.

### **3.3.1.4 Pelatihan Model Pengenalan Wajah Bermasker**

Dataset yang telah dibagi digunakan untuk proses pelatihan model klasifikasi citra wajah bermasker. Bagian dataset yang digunakan untuk pelatihan adalah data

latih. Pelatihan model pengenalan wajah bermasker menggunakan CNN arsitektur ResNet-50 dan InceptionV3 dengan teknik *Additive Angular Margin Loss*.

### 3.3.1.5 Pengujian dan Evaluasi Model

Setelah melakukan pelatihan model, model tersebut diuji performanya menggunakan data uji. Kemudian hasil prediksi dari pengujian dibandingkan dengan hasil sebenarnya dari data uji. Perbandingan tersebut diukur menggunakan *accuracy*, *recall*, *precision*, dan *F-1 Score* yang dijadikan acuan dalam evaluasi model.

### 3.3.3 Kriteria Pengujian

Tahap kriteria pengujian dilakukan pengujian terhadap performa model klasifikasi citra wajah bermasker menggunakan CNN dan *Additive Angular Margin Loss*. Kriteria pengujian yang digunakan untuk melakukan evaluasi terhadap model adalah *confusion matrix* terhadap data latih dan data uji.

### 3.3.4 Format Data Pengujian

Proses pelatihan model menghasilkan nilai akurasi pelatihan, akurasi validasi, *loss*, dan *validation loss* dari *epoch* pertama sampai *epoch* terakhir. Keempat nilai tersebut dibandingkan dan divisualisasikan dalam bentuk plot garis menggunakan Matplotlib. Kemudian, model yang dihasilkan akan diuji dengan data uji. Data hasil pengujian tersebut disajikan dalam format tabel yang terdiri dari nilai evaluasi

berdasarkan kriteria pengujian. Format tabel tersebut akan disajikan sesuai dengan Tabel III-1.

**Tabel III-1.** Tabel Evaluasi Model CNN dengan *Additive Angular Margin Loss*

Konfigurasi Model	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F-1 Score</i>

### 3.3.5 Alat yang Digunakan dalam Pengujian

Perangkat keras dan perangkat lunak yang digunakan dalam penelitian adalah sebagai berikut:

- Prosesor : AMD Ryzen 7 2700U with Radeon Vega Mobile Gfx 2.20 GHz
- VGA : AMD Radeon(TM) RX Vega 10 Graphics
- Memori (RAM) : 8 GB
- Sistem Operasi : Windows 10 (64 bit)
- Bahasa Pemrograman: Python
- IDE : Google Colabs, Visual Studio Code

### 3.3.6 Pengujian Penelitian

Pengujian pada penelitian dilakukan menggunakan arsitektur dan *hyperparameter* CNN serta *loss function* yang diajukan. Pengujian tersebut

dilakukan menggunakan data uji berupa citra wajah yang bermasker. Hasil pengujian tersebut disajikan seperti pada Tabel III-1.

### 3.3.7 Analisis Hasil Pengujian

Analisis hasil pengujian dilakukan dengan berdasarkan data hasil yang disajikan menggunakan Tabel III-1. Model terbaik ditentukan berdasarkan hasil analisis tersebut. Setelah model terbaik ditentukan, penulis melakukan analisis terhadap visualisasi plot garis dari akurasi dan *loss* pada data latih dan data uji. Hal ini dilakukan untuk mengetahui model yang dihasilkan *good fitting*, *overfitting* atau *underfitting*.

## 3.4 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak yang digunakan pada penelitian adalah *Rational Unified Process* (RUP). Langkah-langkah yang dilakukan dalam pengembangan perangkat lunak pada penelitian ini dijelaskan secara detail pada sub bab 3.4.1 sampai sub bab 3.4.4.

### 3.4.1 Fase Insepsi

Fase insepsi berfokus tahapan *business modelling* dan *requirements*. Pada tahap *business modelling*, penulis memahami dan mengeksplorasi terkait ruang lingkup masalah yang diteliti. Pada tahap *requirements*, penulis menyusun *user requirements* dan *functional requirements* dari perangkat lunak yang dikembangkan.

### 3.4.2 Fase Elaborasi

Fase elaborasi berfokus pada tahap *requirements, analysis and design*, dan *implementation*. Pada tahap *requirements*, penulis menganalisa *requirements* dan *use case diagram* sederhana yang telah dibuat. Setelah itu, penulis menentukan arsitektur perangkat lunak, desain basis data, dan desain antarmuka yang berdasarkan pada *requirements*. Kemudian pada tahap *analysis and design*, penulis membuat *activity diagram*, *sequence diagram*, dan *class diagram*. Pada tahap *implementation*, penulis mendokumentasikan arsitektur perangkat lunak, desain basis data, desain antarmuka, *activity diagram*, *sequence diagram*, dan *class diagram*.

### 3.4.3 Fase Konstruksi

Fase konstruksi berfokus pada tahap *analysis and design, implementation*, dan *testing and deployment*. Pada tahap *analysis and design*, penulis membangun arsitektur CNN dengan model pra-latih ResNet-50 dengan teknik *additive angular margin loss*. Kemudian dilanjutkan tahap *implementation*, penulis melakukan pra-pengolahan terhadap dataset, membagi dataset menjadi data latih dan data uji, melakukan augmentasi citra terhadap data latih. Setelah itu, penulis melakukan pembelajaran model CNN dan membangun perangkat lunak sistem pengenalan wajah. Pada tahap *testing and deployment*, penulis menguji model CNN dengan data uji dan men-*deploy* model yang telah dibuat ke perangkat lunak yang telah dikembangkan.

### 3.4.4 Fase Transisi

Pada fase transisi hanya berfokus di tahapan *testing and deployment*. Pada tahap tersebut, penulis menyusun beberapa kenario pengujian. Setelah itu, penulis melakukan pengujian berdasarkan scenario tersebut. Hasil pengujian tersebut, didokumentasikan secara rapi oleh penulis untuk keperluan keberlanjutan pengembangan perangkat lunak.

### 3.5 Manajemen Proyek Penelitian

Manajemen proyek penelitian merupakan perencanaan kegiatan yang menjadi acuan penulis dalam menjalani proses penelitian. Adapun penjadwalan kegiatan-kegiatan yang dilakukan selama proses penelitian dirincikan menggunakan *Work Breakdown Structure* (WBS) dapat dilihat pada tabel III-2.

**Tabel III-2.** WBS Penelitian Pengenalan Citra Wajah Bermasker Menggunakan CNN dengan Teknik *Additive Angular Margin Loss*

ID	Nama Kegiatan	Durasi	Mulai	Selesai
	<b>Klasifikasi Citra Wajah Bermasker Menggunakan CNN dengan Teknik <i>Additive Angular Margin Loss</i></b>	<b>108 hari</b>	<b>09/08/22</b>	<b>23/11/22</b>
	<b>Menetapkan ruang lingkup penelitian</b>	<b>23 hari</b>	<b>09/08/22</b>	<b>05/09/22</b>
T01	Menentukan masalah penelitian	3 hari	09/08/22	11/08/22

ID	Nama Kegiatan	Durasi	Mulai	Selesai
T02	Mengumpulkan literatur dan melakukan studi literatur untuk penyusunan latar belakang masalah	7 hari	12/08/22	18/08/22
T03	Menyusun latar belakang, rumusan masalah, tujuan, dan manfaat penelitian	7 hari	19/08/22	25/08/22
T04	Menentukan batasan masalah	2 hari	26/08/22	27/08/22
T05	Membuat sistematika penulisan dan kesimpulan	1 hari	28/08/22	28/08/22
T06	Revisi	3 hari	29/08/22	31/08/22
	<b>Membuat kajian literature</b>	<b>9 hari</b>	<b>01/09/22</b>	<b>09/09/22</b>
T07	Menyusun dan menganalisis dasar-dasar teori yang mendukung penelitian	3 hari	01/09/22	03/09/22
T08	Menganalisis penelitian-penelitian sebelumnya yang relevan dengan penelitian	3 hari	04/09/22	06/09/22
T09	Revisi	3 hari	07/09/22	09/09/22
	<b>Menentukan Metodologi Penelitian</b>	<b>10 hari</b>	<b>10/09/22</b>	<b>19/09/22</b>
T10	Menentukan tahapan pengujian	4 hari	10/09/22	13/09/22

ID	Nama Kegiatan	Durasi	Mulai	Selesai
T11	Menentukan metode dan kriteria penelitian	3 hari	14/09/22	16/09/22
	Pengembangan Perangkat Lunak menggunakan RUP	3 hari	17/09/22	19/09/22
	Insepsi	4 hari	20/09/22	23/09/22
T12	Memahami masalah yang diteliti	1 hari	20/09/22	20/09/22
T13	Membuat <i>business modelling</i>	2 hari	21/09/22	22/09/22
T14	Menentukan <i>user requirements</i> dan <i>functional requirements</i>	1 hari	23/09/22	23/09/22
	Elaborasi	5 hari	24/09/22	28/09/22
T15	Melakukan analisa pada <i>requirements</i>	1 hari	24/09/22	24/09/22
T16	Membuat <i>usecase, activity diagram, sequence diagram, class diagram.</i>	2 hari	25/09/22	26/09/22
T17	Membuat desain tampilan antarmuka perangkat lunak	2 hari	27/09/22	28/09/22
	Konstruksi	31 hari	29/09/22	29/10/22
T18	Melakukan pra-pengolahan data	7 hari	29/09/22	05/10/22
T19	Membangun arsitektur CNN model pra-latih ResNet-50 dengan teknik <i>Additive Angular Margin Loss</i>	14 hari	06/10/22	19/10/22



ID	Nama Kegiatan	Durasi	Mulai	Selesai
T20	Membangun perangkat lunak pengenalan wajah bermasker.	7 hari	20/10/22	26/10/22
T21	Melakukan <i>deployment</i> model pembelajaran ke perangkat lunak	3 hari	27/10/22	29/10/22
	Transisi	8 hari	30/10/22	06/11/22
T22	Menyusun beberapa skenario pengujian	2 hari	30/10/22	31/10/22
T23	Melakukan pengujian terhadap pengguna.	3 hari	01/10/22	03/11/22
T24	Menyusun hasil pengujian	3 hari	04/11/22	06/11/22
	<b>Evaluasi dan Analisa Hasil Pengujian</b>	<b>18 hari</b>	<b>06/11/22</b>	<b>23/11/22</b>
T25	Evaluasi dan Analisis Hasil Pengujian	3 hari	06/11/22	08/11/22
T26	Menarik kesimpulan	1 hari	09/11/22	09/11/22
T27	Menyusun laporan	14 hari	10/11/22	23/11/22

### 3.6 Kesimpulan

Bab ini telah menjabarkan secara rinci tahapan-tahapan yang dilakukan penulis selama penelitian. Selain itu, bab ini juga menjelaskan juga metode pengembangan perangkat lunak yang digunakan oleh penulis untuk membangun perangkat lunak sistem pengenalan citra wajah bermasker.

## **BAB IV**

### **PENGEMBANGAN PERANGKAT LUNAK**

#### **4.1 Pendahuluan**

Bab ini menjabarkan proses pengembangan perangkat lunak yang digunakan sebagai alat penelitian. Metode *Rational Unified Process* (RUP) digunakan dalam pengembangan perangkat lunak tersebut.

#### **4.2 Rational Unified Process**

RUP terdiri dari empat fase yaitu fase insepisi, fase elaborasi, fase transisi, dan fase konstruksi. Keempat fase tersebut dijelaskan secara rinci pada subbab 4.2.

##### **4.2.1 Fase Insepisi**

Pada tahap insepisi dilakukan beberapa aktivitas yaitu membuat pemodelan bisnis, menentukan kebutuhan pengguna, fungsional, dan non-fungsional.

##### **4.2.1.1 Pemodelan Bisnis**

Perangkat lunak yang dikembangkan dalam penelitian ini adalah sistem pengenalan citra wajah bermasker berbasis web. Data yang digunakan sebagai masukan pada perangkat lunak yaitu data gambar orang bermasker. Keluaran yang dihasilkan dari perangkat lunak ini adalah identitas orang dari citra gambar masukan.

#### 4.2.1.2 Kebutuhan Sistem

Kebutuhan sistem pada perangkat lunak terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional adalah layanan atau proses yang harus disediakan oleh perangkat lunak, sedangkan kebutuhan non-fungsional adalah kebutuhan pelengkap yang menekankan sifat perilaku perangkat lunak. Kebutuhan fungsional dan kebutuhan non-fungsional tersebut dapat dilihat pada Tabel IV-1 dan Tabel IV-2.

**Tabel IV-1. Kebutuhan Fungsional Perangkat Lunak**

No	Kebutuhan Fungsional
1	Perangkat lunak dapat mendeteksi wajah dan mengklasifikasi atau mengenali identitas orang.
2	Perangkat lunak dapat menerima masukan citra wajah yang dimasukkan pengguna berupa gambar.
3	Perangkat lunak dapat menampilkan hasil evaluasi model.

**Tabel IV-2. Kebutuhan Non-Fungsional Perangkat Lunak**

No	Kebutuhan Non-Fungsional
1	Perangkat lunak memiliki antarmuka yang simpel dan mudah digunakan.

#### 4.2.1.3 Analisis dan Desain


##### a. Analisis Data


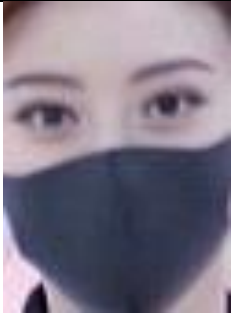


Data yang digunakan pada proses pembuatan model merupakan ata gambar memiliki format JPG dengan resolusi yang bervariasi. Namun, data gambar tersebut disamakan rasio resolusinya menjadi 180 x 180 menggunakan bantuan pustaka TensorFlow.


Dari 525 kelas yang tersedia di dataset, penelitian yang dilakukan hanya menggunakan 15 kelas. Hal ini dikarenakan 15 kelas tersebut terdapat di data wajah bermasker maupun data wajah tidak bermasker dan memiliki jumlah gambar minimal 20 gambar pada masing-masing kelas.

Tabel IV-3 dan Tabel IV-4 berikut secara berurutan menunjukkan sampel gambar dari masing-masing kelas citra wajah bermasker dan citra wajah tidak bermasker dari dataset RMFD.



**Tabel IV-3.** Sampel Citra Wajah Bermasker

No	Nama File	Identitas Wajah	Gambar
1	0_0_1.jpg	Duhaitao	


No	Nama File	Identitas Wajah	Gambar
2	0_0_0.jpg	Houminghao	
3	0_0_0.jpg	Jingtian	
4	0_0_2.jpg	Linxinru	
5	0_0_0.jpg	Luhan	

No	Nama File	Identitas Wajah	Gambar
6	0_0_3.jpg	Masu	
7	0_0_0.jpg	Matianyu	
8	0_0_0.jpg	Wuyifan	
9	0_0_2.jpg	Xuezhiqian	




No	Nama File	Identitas Wajah	Gambar
10	0_0_0.jpg	Xuweizhou	
11	0_0_0.jpg	Yangyang	
12	0_0_1.jpg	Yuanshanshan	
13	0_0_0.jpg	Zhangruoyun	

No	Nama File	Identitas Wajah	Gambar
14	0_0_0.jpg	Zhangyixing	
15	0_0_0.jpg	zhangyuxi	


**Tabel IV-4.** Sampel Citra Wajah Tidak Bermasker

No	Nama File	Identitas Wajah	Gambar
1	0_0_duhaitao_0010.jpg	Duhaitao	
2	0_0_houminghao_0008.jpg	Houminghao	



No	Nama File	Identitas Wajah	Gambar
3	0_0_jingtian_0002.jpg	Jingtian	
4	0_0_linxinru_0002.jpg	Linxinru	
5	0_0_luhan_0012.jpg	Luhan	
6	0_0_masu_0005.jpg	Masu	
7	0_0_matianyu_0004.jpg	Matianyu	
8	0_0_wuyifan_0010.jpg	Wuyifan	

No	Nama File	Identitas Wajah	Gambar
9	0_0_xuezhiqian_0003.jpg	Xuezhiqian	
10	0_0_xuweizhou_0011.jpg	Xuweizhou	
11	0_0_yangyang_0002.jpg	Yangyang	
12	0_0_yuanshanshan_0001.jpg	Yuanshanshan	
13	0_0_zhangruoyun_0009.jpg	Zhangruoyun	
14	0_0_zhangyixing_0002.jpg	Zhangyixing	

No	Nama File	Identitas Wajah	Gambar
15	0_0_zhangyuxi_0007.jpg	zhangyuxi	

b. Augmentasi Data

Augmentasi data dilakukan untuk menambah jumlah dan variasi data pada dataset. Augmentasi data juga bertujuan untuk meminimalisir terjadinya *overfitting*. Tabel IV-5 berikut menunjukkan konfigurasi parameter yang digunakan untuk melakukan augmentasi data.

**Tabel IV-5.** Konfigurasi Parameter Augmentasi Data

No	Jenis Parameter	Nilai Parameter
1	<i>Rescale</i>	1/255
2	<i>Rotation range</i>	20
3	<i>Zoom range</i>	0.05
4	<i>Horizontal flip</i>	<i>True</i>
5	<i>fill mode</i>	<i>nearest</i>

c. Pembagian Data

Dataset yang digunakan terdiri dari dataset citra wajah bermasker dan citra wajah tidak bermasker. Dalam pemakaiannya, proses pelatihan model menggunakan 2 jenis skenario, yaitu dataset citra wajah bermasker saja dan

gabungan dataset citra wajah bermasker dan citra wajah tidak bermasker. Dataset tersebut dibagi menjadi 3 bagian dengan bagian, yaitu mendekati 3:1:1. Tiga bagian tersebut secara berurutan mewakili data latih, data validasi, dan data uji. Tabel IV-6 berikut menunjukkan jumlah gambar pada masing-masing bagian data setelah mengalami pembagian.

**Tabel IV-6.** Pembagian Data

No	Jenis Dataset	Data Latih	Data Validasi	Data Uji
1	<i>Masked Face Dataset</i>	221	71	81
2	<i>Masked and Non Masked Face Dataset</i>	1962	649	668

d. Skenario Percobaan

Percobaan terhadap model yang dibangun dilakukan dalam 8 skenario. Skenario-skenario tersebut memiliki variasi pada konfigurasi parameternya. Tabel IV-7 berikut menunjukkan skenario-skenario yang digunakan dalam membangun model CNN.

**Tabel IV-7.** Skenario-Skenario Percobaan

No	Model	Arsitektur	Dataset
1	Model 1	Resnet50	<i>Masked Face Dataset</i>
2	Model 2	InceptionV3	<i>Masked Face Dataset</i>
3	Model 3	Resnet50+ArcFace	<i>Masked Face Dataset</i>
4	Model 4	InceptionV3+ArcFace	<i>Masked Face Dataset</i>
5	Model 5	Resnet50	<i>Masked and Non Masked Face Dataset</i>

No	Model	Arsitektur	Dataset
6	Model 6	InceptionV3	<i>Masked and Non Masked Face Dataset</i>
7	Model 7	Resnet50+ArcFace	<i>Masked and Non Masked Face Dataset</i>
8	Model 8	InceptionV3+ArcFace	<i>Masked and Non Masked Face Dataset</i>

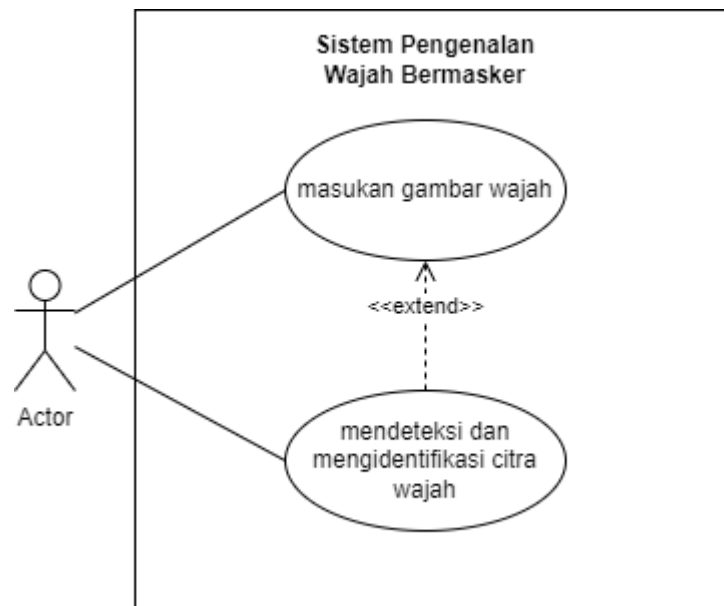
Selain perbedaan konfigurasi parameter yang berbeda, terdapat juga parameter yang sama yang digunakan dalam penelitian. Tabel IV-8 berikut menunjukkan parameter-parameter yang sama yang digunakan pada setiap skenario yang ada.

**Tabel IV-8.** Konfigurasi Parameter Tetap

No	Jenis Parameter	Nilai Parameter
1	<i>Optimizer</i>	Adam
2	<i>Learning rate</i>	0.002
3	<i>Batch Size</i>	32
4	<i>Loss</i>	<i>Categorical Crossentropy</i>
5	<i>Epoch</i>	25

e. Use Case Diagram

*Use case diagram* menggambarkan kegiatan yang dilakukan oleh aktor terhadap perangkat lunak. Use case diagram perangkat lunak sistem pengenalan citra wajah masker ditunjukkan pada Gambar IV-1.



**Gambar IV-1.** Use Case Diagram Perangkat Lunak

f. Definisi Aktor

Aktor pada perangkat lunak sistem pengenalan citra wajah bermasker ditunjukkan pada Tabel IV-9.

**Tabel IV-9.** Definisi Aktor

No	Aktor	Deskripsi
1	Pengguna	Orang yang dapat berinteraksi dengan perangkat lunak dan menggunakan semua fitur yang telah tersedia di dalam perangkat lunak.

g. Definisi Use Case

Use Case pada perangkat lunak sistem pengenalan citra wajah bermasker didefinisikan pada Tabel IV-10.

**Tabel IV-10.** Definisi *Use Case*

No	Use Case	Definisi
1	Masukan gambar wajah	Kegiatan ini digunakan untuk memuat data ke perangkat lunak dari data uji yang berupa gambar.
2	Mendeteksi dan mengidentifikasi citra wajah	Kegiatan ini digunakan untuk melakukan proses deteksi dan identifikasi citra wajah bermasker menggunakan arsitektur model ResNet-50 dengan teknik <i>additive angular margin loss</i> .

#### h. Skenario *Use Case*

Skenario *use case* menguraikan urutan spesifik dari aksi aktor dan reaksi sistem berdasarkan *use case diagram* yang telah dibuat pada gambar IV-1. Berikut skenario *use case* dari perangkat lunak sistem pengenalan citra wajah bermasker yang diuraikan pada Tabel IV-11 dan Tabel IV-12.

**Tabel IV-11.** Skenario *Use Case* Masukan Gambar Wajah

Identifikasi	
Nomor	101
Nama	Masukan Gambar Wajah
Tujuan	Proses ini digunakan untuk mendapatkan paket
Deskripsi	Use case menggambarkan proses dalam mendapatkan paket dalam lingkup area tertentu
Aktor	Pengguna
Kondisi Awal	Pengguna membuka aplikasi

Identifikasi	
<b>Pemicu</b>	Menekan tombol pilih gambar.
<b>Skenario Utama (Mengupload gambar)</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Menekan tombol pilih gambar	
	2. Menampilkan folder dan file gambar di direktori penyimpanan
3. Memilih gambar yang ingin diunggah	
4. Menekan tombol unggah	
<b>Kondisi Akhir</b>	Gambar berhasil diunggah

**Tabel IV-12.** Skenario *Use Case* Mendeteksi dan Mengidentifikasi Citra Wajah

Identifikasi	
<b>Nomor</b>	102
<b>Nama</b>	Mendeteksi dan Mengidentifikasi Citra Wajah
<b>Tujuan</b>	Proses ini digunakan untuk mendeteksi dan mengenali wajah pengguna
<b>Deskripsi</b>	Use case menggambarkan proses deteksi dan identifikasi citra wajah pengguna yang bermasker atau tidak bermasker
<b>Aktor</b>	Pengguna
<b>Kondisi Awal</b>	Pengguna telah mengunggah gambar wajah
<b>Pemicu</b>	Menekan tombol kenali



Identifikasi	
<b>Skenario Utama (Mendeteksi wajah)</b>	
Aksi Aktor	Reaksi Sistem
1. Menekan tombol kenali	
	2. Menampilkan citra wajah yang telah dibatasi kotak dan dilengkapi nama pengguna
<b>Kondisi Akhir</b>	Citra wajah berhasil diprediksi
<b>Skenario Alternatif (Tidak mendeteksi wajah)</b>	
Aksi Aktor	Reaksi Sistem
Melakukan no 1	
	2. mengirim pesan peringatan bahwa tidak terdapat wajah di gambar yang diunggah
<b>Kondisi Akhir</b>	Citra wajah gagal diprediksi

#### 4.2.2 Fase Elaborasi

Setelah fase insepri, pengembangan perangkat lunak dilanjutkan ke fase elaborasi. Pada fase ini dilakukan beberapa aktivitas yaitu membuat perangkat data perancangan antarmuka perangkat lunak, *activity diagram*, dan *sequence diagram*.

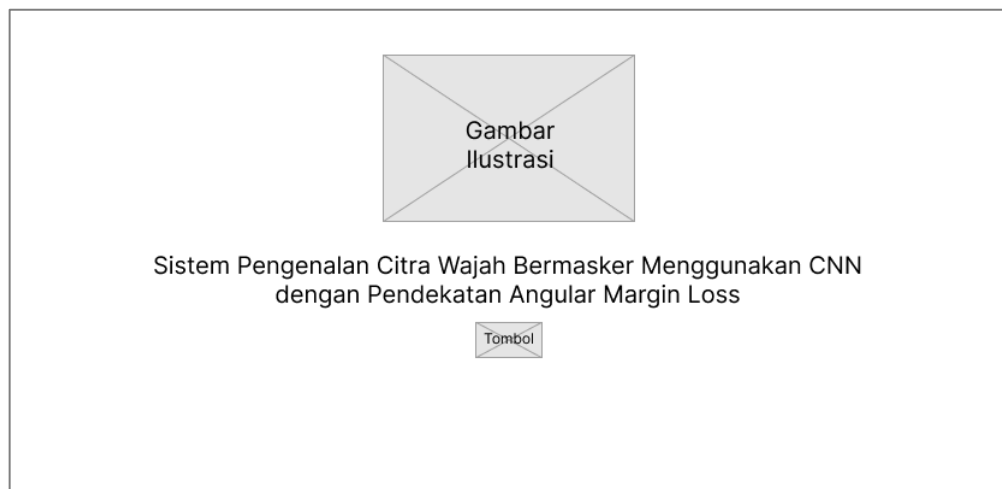
#### 4.2.2.1 Pemodelan Bisnis

##### a. Perancangan Data

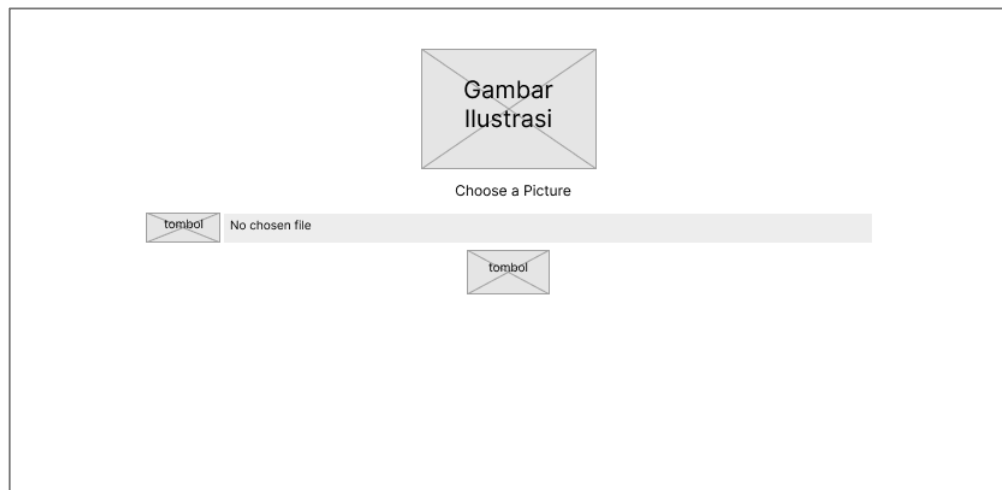
Perangkat lunak yang dibangun memiliki kemampuan untuk mengidentifikasi identitas dari citra wajah yang diunggah. Data yang akan melalui proses pengenalan adalah citra wajah yang termasuk dalam bagian data uji pada dataset.

##### b. Perancangan Antarmuka

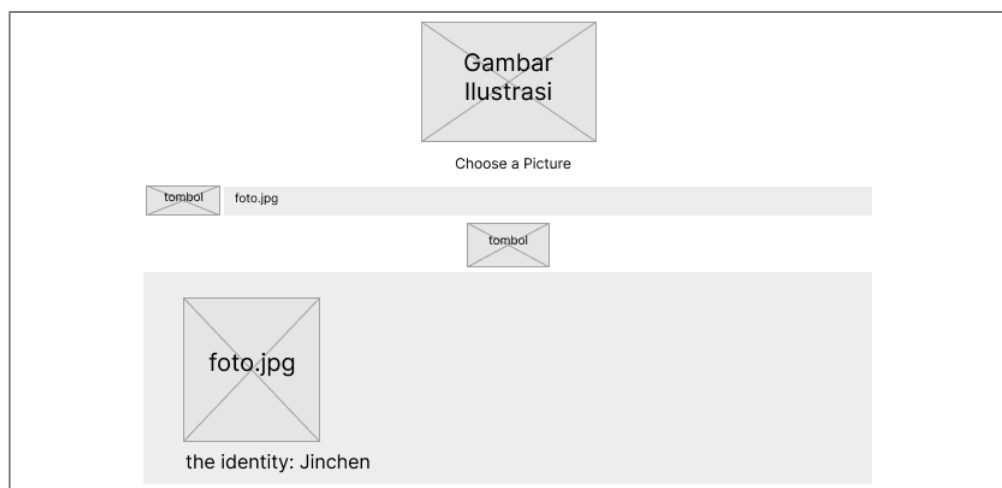
Perancangan antarmuka dilakukan bertujuan memberikan tampilan perangkat lunak yang sesuai dengan kebutuhan pengguna. Gambar IV-2 sampai dengan Gambar IV-4 merupakan rancangan antarmuka perangkat lunak yang dibangun.



**Gambar IV-2.** Rancangan Tampilan Halaman Depan



**Gambar IV-3.** Rancangan Tampilan Halaman Pengenalan Wajah



**Gambar IV-4.** Rancangan Tampilan Halaman Pengenalan Wajah Ketika Menampilkan Hasil Identifikasi

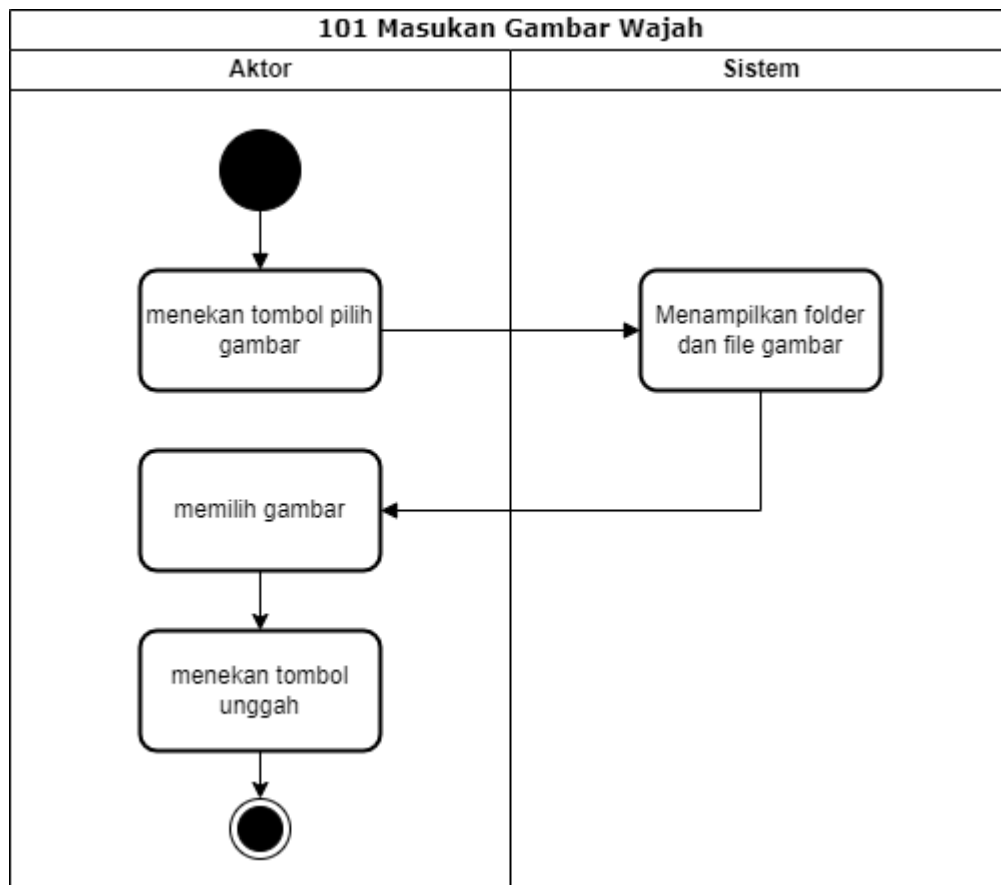
#### 4.2.2.2 Kebutuhan Sistem

Perangkat keras, perangkat lunak, dan Bahasa pemrograman diperlukan dalam pengembangan perangkat lunak pada penelitian ini. Bahasa pemrograman yang digunakan yaitu Python, sedangkan perangkat lunak yang digunakan adalah Google Colab dan Figma.

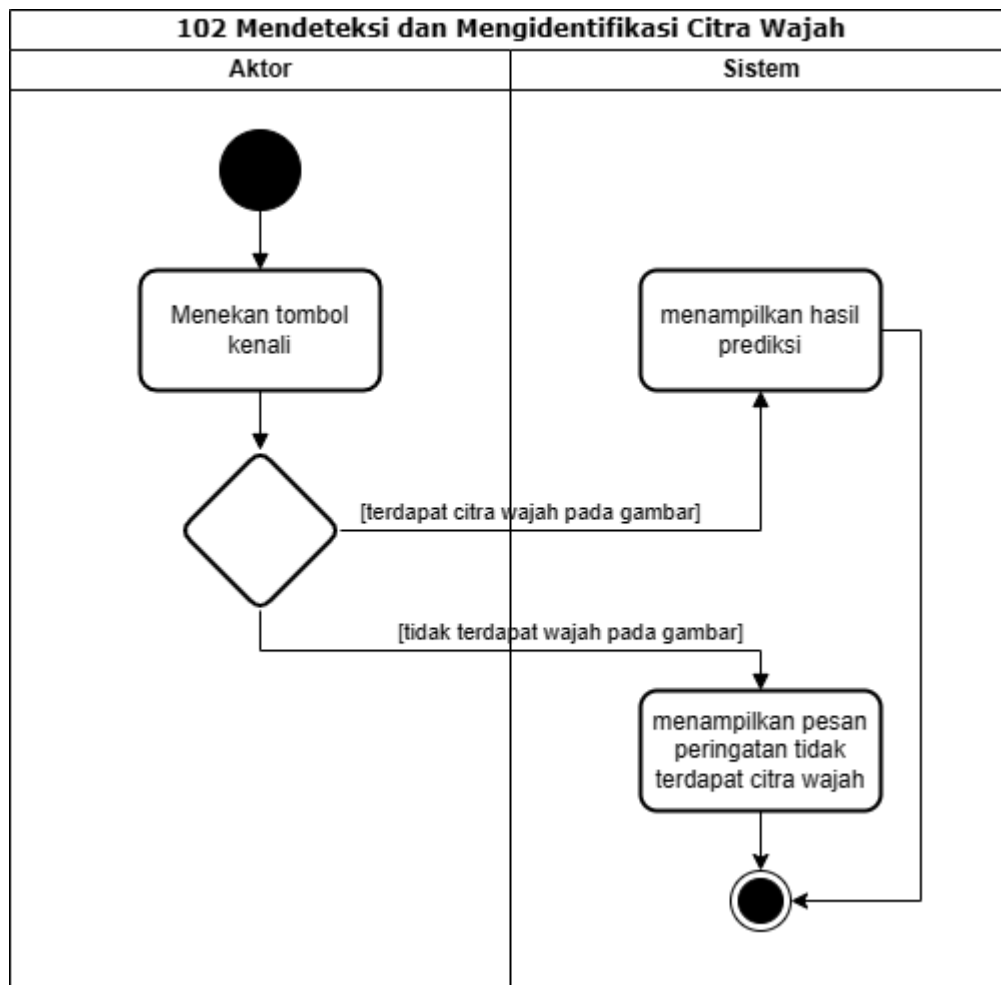
#### 4.2.2.3 Analisis dan Desain

##### a. Activity Diagram

*Activity diagram* menggambarkan urutan aktivitas proses yang berjalan dalam sistem berdasarkan *use case* dan skenario *use case* yang telah dibuat. *Activity diagram* dari perangkat lunak sistem pengenalan wajah bermasker dapat dilihat pada Gambar IV-5 dan Gambar IV-6.



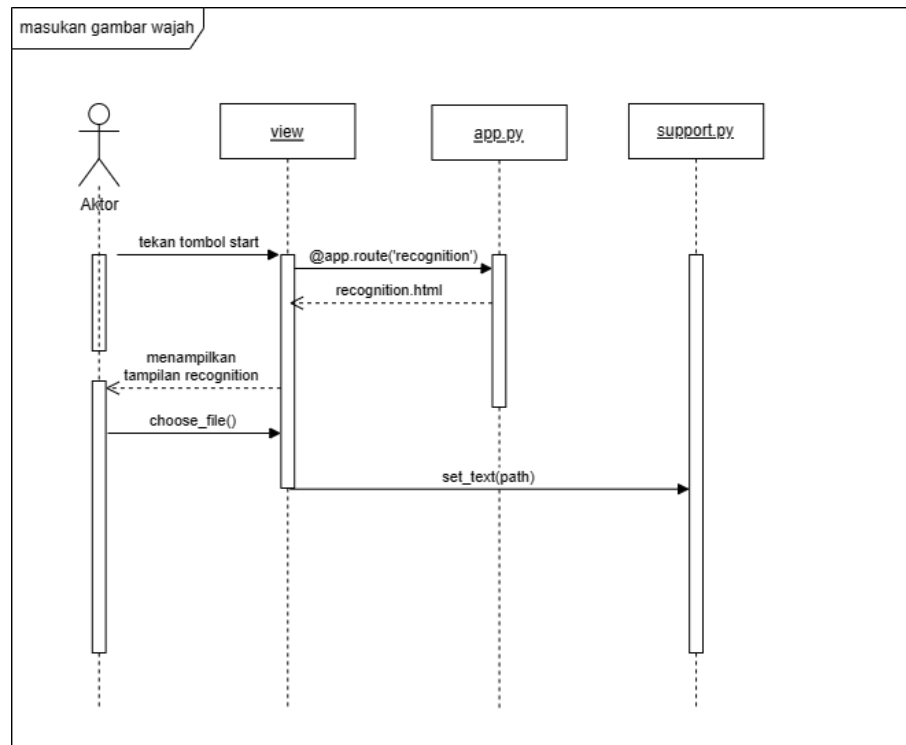
**Gambar IV-5.** Activity Diagram Masukan Gambar Wajah



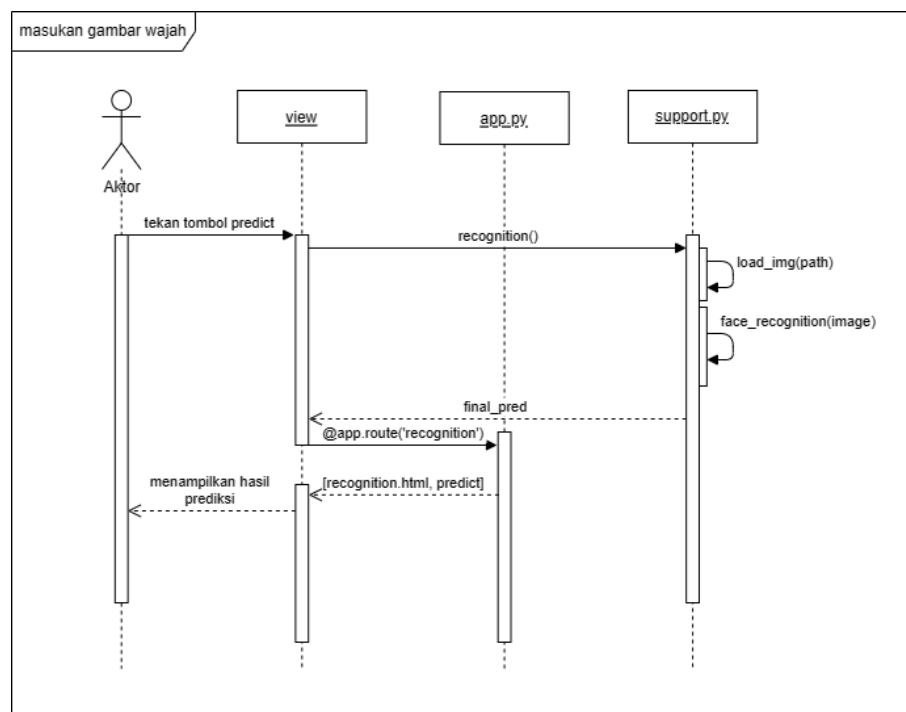
**Gambar IV-6.** Activity Diagram Deteksi dan Identifikasi Citra Wajah

b. Sequence Diagram

*Sequence diagram* menggambarkan alur interaksi antar objek di dalam sistem. Berdasarkan *use case* pada Gambar IV-1, terdapat dua *sequence diagram* yaitu masukan gambar wajah pada Gambar IV-7 dan deteksi dan identifikasi citra wajah pada Gambar IV-8.



**Gambar IV-7.** *Sequence Diagram* Masukan Gambar Wajah



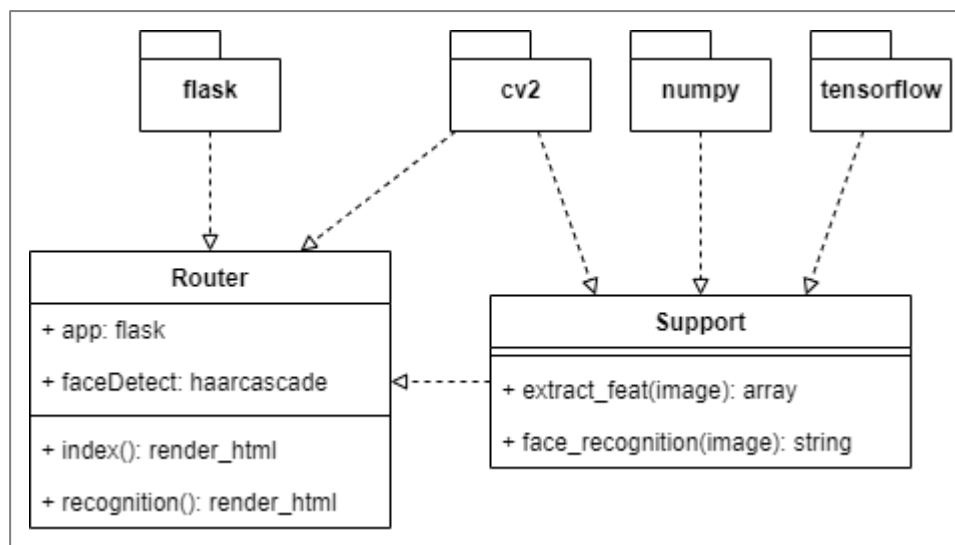
**Gambar IV-8.** *Sequence Diagram* Deteksi dan Identifikasi Citra Wajah

### 4.2.3 Fase Kontruksi

Pada fase ini dilakukan beberapa aktivitas yang berkait dengan pembuatan perangkat lunak, yaitu merancang pemodelan *class diagram*, mengimplementasikan kelas-kelas tersebut pada perangkat lunak, dan mengimplementasikan antarmuka yang telah dirancang.

#### 4.2.3.1 Kebutuhan Sistem

Pada tahap ini dijelaskan kelas yang dibangun pada perangkat lunak dan hubungannya satu sama lain melalui pemodelan diagram kelas. Diagram kelas perangkat lunak pada penelitian ini ditunjukkan oleh Gambar IV-9 berikut.



**Gambar IV-9.** Diagram Kelas

#### 4.2.3.2 Implementasi

Pada fase kontruksi tahap implementasi, rancangan yang telah dibuat diimplementasikan menjadi sebuah perangkat lunak. Dalam subbab ini,

implementasi dibahas dalam dua bagian, yaitu implementasi kelas dan implementasi antarmuka. Alat pendukung dalam proses implementasi yaitu bahasa pemrograman Python dan pustaka Flask.

a. Implementasi Kelas

Kelas yang telah dirancang sebelumnya diimplementasikan menggunakan bahasa pemrograman Python. Tabel IV-13 berikut menunjukkan hasil implementasi dari kelas-kelas tersebut.

**Tabel IV-13.** Keterangan Implementasi Kelas

No	Nama Kelas	Nama File	Keterangan
1	Router	app.py	Kelas yang mengatur <i>routing</i> saat pengguna ingin membuka laman tertentu.
2	Support	support.py	Kelas yang melakukan proses deteksi dan identifikasi wajah dari gambar yang dimasukkan pengguna.

b. Implementasi Antarmuka

Pada tahap ini, antarmuka yang telah dirancang pada fase elaborasi diimplementasikan ke perangkat lunak. Implementasi antarmuka tersebut menggunakan *flask*.

Tabel IV-14 adalah daftar file HTML sebagai bentuk implementasi antarmuka perangkat lunak.

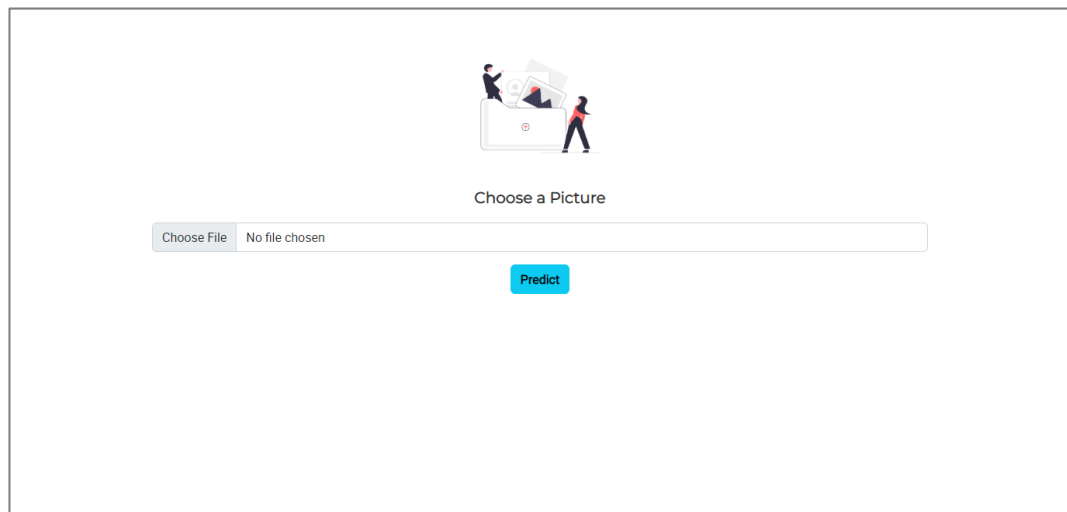


**Tabel IV-14.** Daftar File HTML

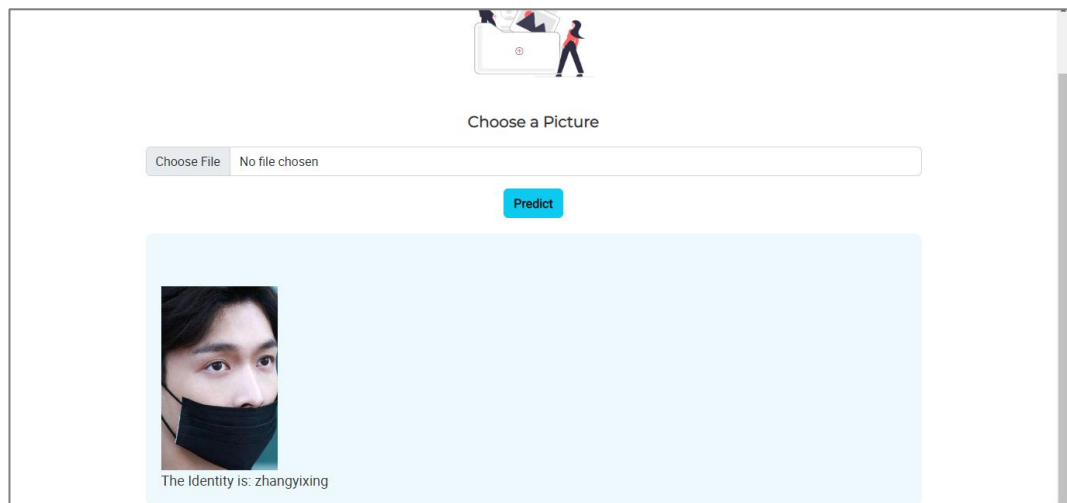
No	Nama File	Keterangan
1	base.html	Sebagai kerangka tampilan dari setiap halaman perangkat lunak.
2	index.html	Sebagai tampilan dari halaman depan.
3	recognition.html	Sebagai tampilan dari halaman unggah gambar wajah dan identifikasi wajah.

Gambar IV-10 sampai Gambar IV-12 berikut menunjukkan tampilan antarmuka dari file HTML pada Tabel IV-14 ketika dijalankan.

**Gambar IV-10.** Antarmuka Halaman Depan



**Gambar IV-11.** Antarmuka Halaman Pengenalan Wajah



**Gambar IV-12.** Antarmuka Halaman Pengenalan Wajah setelah Ditekan Tombol

*Predict*

#### 4.2.4 Fase Transisi

Fase transisi merupakan fase terakhir dari pengembangan perangkat lunak dengan metode RUP. Pada fase ini, beberapa aktivitas dilakukan seperti menyusun scenario pengujian, melakukan pengujian, dan mendokumentasikan hasil pengujian.

#### 4.2.4.1 Pemodelan Bisnis

Pada tahap pemodelan bisnis, perangkat lunak yang telah dibangun diuji dengan menggunakan *Black Box Testing*. Pengujian diawali dengan pembuatan rencana pengujian.

#### 4.2.4.2 Analisis dan Desain

Tahap analisis dan desain membahas tentang rencana pengujian. Rencana pengujian disusun berdasarkan dari diagram *use case* yang ada pada Gambar IV-1. Rencana pengujian untuk *use case* masukan gambar wajah ditunjukkan oleh Tabel IV-15. Kemudian, rencana pengujian untuk *use case* deteksi dan identifikasi wajah ditunjukkan oleh Tabel IV-16.

**Tabel IV-15.** Rencana Pengujian *Use Case* Masukan Gambar Wajah

No	ID	Pengujian	Tingkat Pengujian
1	UC-101-1	Mengunggah gambar wajah dari data uji.	Pengujian unit

**Tabel IV-16.** Rencana Pengujian *Use Case* Deteksi dan Identifikasi Wajah

No	ID	Pengujian	Tingkat Pengujian
1	UC-102-1	Melakukan proses deteksi dan identifikasi wajah terhadap gambar/video yang diunggah.	Pengujian unit

#### 4.2.4.3 Implementasi

Pada tahap implementasi, dibahas hasil implementasi dari rencana pengujian pada Tabel IV-11 dan Tabel IV-12 terhadap perangkat lunak yang dibangun. Pengujian untuk *use case* masukan gambar wajah ditunjukkan oleh Tabel IV-17. Kemudian, pengujian untuk *use case* deteksi dan identifikasi wajah ditunjukkan oleh Tabel IV-18.

**Tabel IV-17.** Pengujian *Use Case* Masukan Gambar/Video Wajah

ID	Prosedur Pengujian	Keluaran yang Diharapkan	Keluaran yang Didapatkan	Kesimpulan
UC-101-1	Tekan tombol “pilih gambar”, kemudian pilih gambar.	Menampilkan gambar yang telah diunggah	Menampilkan gambar yang telah diunggah	Terpenuhi

**Tabel IV-18.** Pengujian *Use Case* Deteksi dan Identifikasi Wajah

ID	Prosedur Pengujian	Keluaran yang Diharapkan	Keluaran yang Didapatkan	Kesimpulan
UC-102-1	Tekan tombol “kenali”	Menampilkan identitas dari citra wajah yang diunggah	Menampilkan identitas dari citra wajah yang diunggah	Terpenuhi

### 4.3 Kesimpulan

Metode RUP digunakan untuk pengembangan perangkat lunak pada penelitian ini. RUP terdiri dari 4 fase, yaitu fase insepasi, fase elaborasi, fase konstruksi, dan fase transisi. Fase insepasi melakukan analisis data dan rancangan perangkat lunak. Fase elaborasi membuat rancangan antarmuka perangkat lunak dan *sequence diagram*. Fase konstruksi mengimplementasi rancangan yang telah dibuat pada fase sebelumnya. Terakhir, fase transisi melakukan pengujian terhadap perangkat lunak yang telah dibangun.

## BAB V

### HASIL DAN ANALISIS PENELITIAN

#### 5.1 Pendahuluan

Bab ini membahas hasil penelitian pengenalan citra wajah bermasker menggunakan CNN dengan pendekatan *additive angular margin loss* yang telah dikembangkan berdasarkan parameter terbaik. Performa dari model CNN yang dilatih dan diuji diukur menggunakan metrik pengukuran berupa *confussion matrix*, *accuracy*, *precision*, *recall*, *F-1 Score*.

#### 5.2 Data Hasil Penelitian

Percobaan dilakukan dengan melakukan konfigurasi model yang bervariasi. Konfigurasi-konfigurasi yang digunakan mengacu pada Tabel IV-5, Tabel IV-6, Tabel IV-7, dan Tabel IV-8. Tabel V-1 berikut menunjukkan ringkasan informasi dari keempat tabel yang disebutkan sebelumnya.

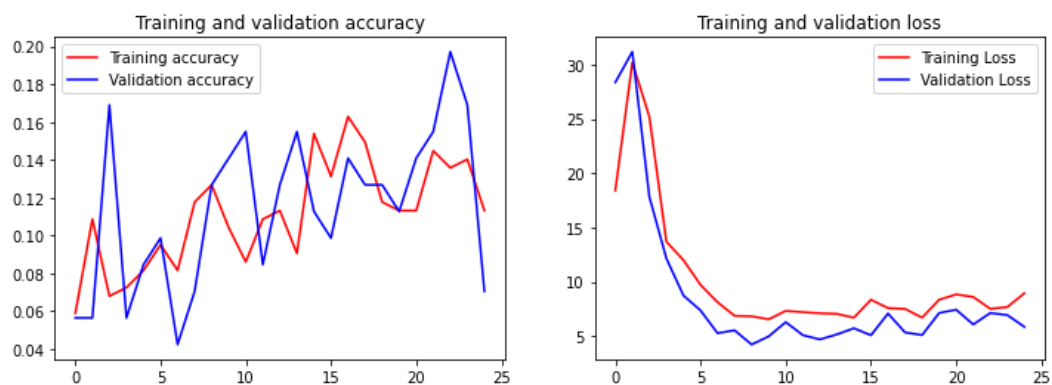
**Tabel V-1.** Daftar Variasi Konfigurasi Model

No	Arsitektur	Dataset	Rasio Data
1	Resnet50	<i>Masked Face Dataset</i>	221:71:81
2	InceptionV3	<i>Masked Face Dataset</i>	221:71:81
3	Resnet50 + ArcFace	<i>Masked Face Dataset</i>	221:71:81
4	InceptionV3 + ArcFace	<i>Masked Face Dataset</i>	221:71:81

No	Arsitektur	Dataset	Rasio Data
5	Resnet50	<i>Masked and Non Masked Face Dataset</i>	1962:649:668
6	InceptionV3	<i>Masked and Non Masked Face Dataset</i>	1962:649:668
7	Resnet50 + ArcFace	<i>Masked and Non Masked Face Dataset</i>	1962:649:668
8	InceptionV3 + ArcFace	<i>Masked and Non Masked Face Dataset</i>	1962:649:668

### 5.2.1 Data Hasil Konfigurasi Model 1

Hasil performa model 1 terhadap data latih dan data validasi ditunjukkan pada Gambar V-1. Gambar V-1 menunjukkan grafik akurasi dan *loss* model 1 pada data latih dan data validasi terhadap jumlah *epoch*.



**Gambar V-1.** Grafik Akurasi Model 1

Berdasarkan Gambar V-1, didapatkan akurasi Model 1 dari data latih dan data validasi masing-masing hanya mencapai 16,29% dan 14,08%. Kemudian, *loss* yang

didapatkan terhadap data latih dan data validasi masing-masing adalah 7,5822 dan 7,0848.

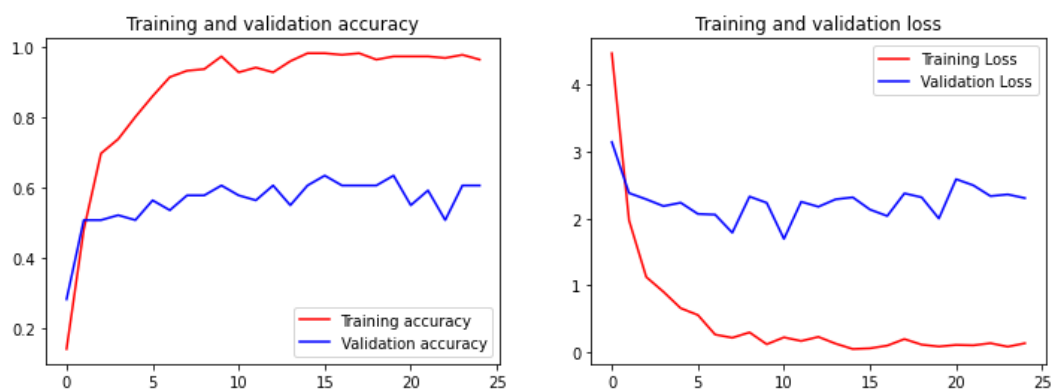
Hasil pengujian performa model terhadap data uji ditunjukkan pada Tabel V-2. Tabel V-2 menunjukkan nilai dari masing-masing metrik evaluasi performa model.

**Tabel V-2.** Hasil *Testing* Model 1

Konfigurasi Model	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F-1 Score</i>
Model 1	0,1358	0,6471	0,1467	0,2392

### 5.2.2 Data Hasil Konfigurasi Model 2

Hasil performa model 2 terhadap data latih dan data validasi ditunjukkan pada Gambar V-2. Gambar V-2 menunjukkan grafik akurasi dan *loss* model 2 pada data latih dan data validasi terhadap jumlah *epoch*.



**Gambar V-2.** Grafik Akurasi Model 2

Berdasarkan Gambar V-2, didapatkan akurasi Model 2 dari data latih dan data validasi masing-masing hanya mencapai 96,38% dan 60,56%. Kemudian, *loss* yang



didapatkan terhadap data latih dan data validasi masing-masing adalah 0,1357 dan 2,3069.

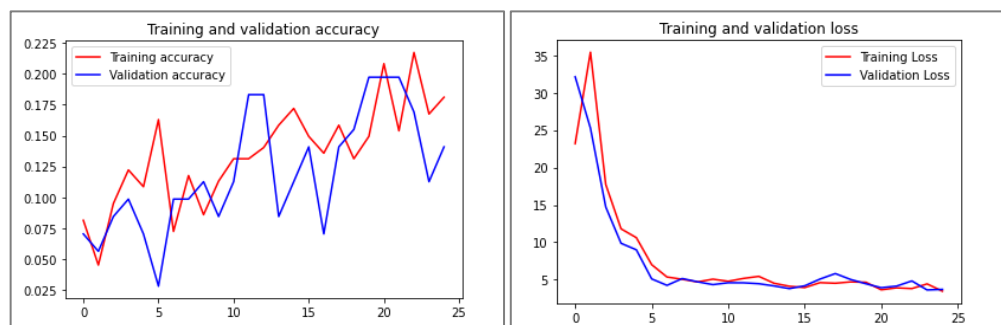
Hasil pengujian performa model terhadap data uji ditunjukkan pada Tabel V-3. Tabel V-3 menunjukkan nilai dari masing-masing metrik evaluasi performa model.

**Tabel V-3.** Hasil *Testing* Model 2

Konfigurasi Model	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F-1 Score</i>
Model 2	0,7901	0,9142	0,8533	0,8827

### 5.2.3 Data Hasil Konfigurasi Model 3

Hasil performa model 3 terhadap data latih dan data validasi ditunjukkan pada Gambar V-3. Gambar V-3 menunjukkan grafik akurasi dan *loss* model 3 pada data latih dan data validasi terhadap jumlah *epoch*.



**Gambar V-3.** Grafik Akurasi Model 3

Berdasarkan Gambar V-3, didapatkan akurasi Model 3 dari data latih dan data validasi masing-masing hanya mencapai 18,10% dan 14,08%. Kemudian, *loss* yang didapatkan terhadap data latih dan data validasi masing-masing adalah 3,4628 dan 3,7043.

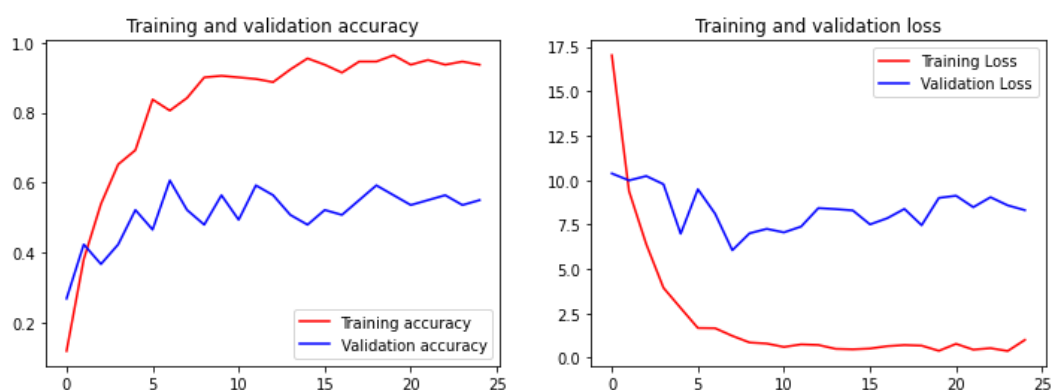
Hasil pengujian performa model terhadap data uji ditunjukkan pada Tabel V-4. Tabel V-4 menunjukkan nilai dari masing-masing metrik evaluasi performa model.

**Tabel V-4.** Hasil *Testing* Model 3

Konfigurasi Model	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F-1 Score</i>
Model 3	0,1111	0,4500	0,1323	0,2044

#### 5.2.4 Data Hasil Konfigurasi Model 4

Hasil performa model 4 terhadap data latih dan data validasi ditunjukkan pada Gambar V-4. Gambar V-4 menunjukkan grafik akurasi dan *loss* model 4 pada data latih dan data validasi terhadap jumlah *epoch*.



**Gambar V-4.** Grafik Akurasi Model 4

Berdasarkan Gambar V-4, didapatkan akurasi Model 4 dari data latih dan data validasi masing-masing hanya mencapai 93,67% dan 54,93%. Kemudian, *loss* yang didapatkan terhadap data latih dan data validasi masing-masing adalah 1,0081 dan 8,3073.

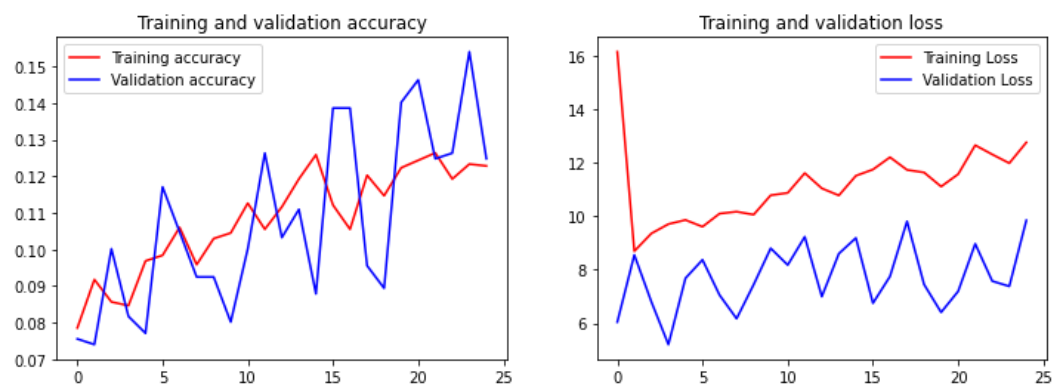
Hasil pengujian performa model terhadap data uji ditunjukkan pada Tabel V-5. Tabel V-5 menunjukkan nilai dari masing-masing metrik evaluasi performa model.

**Tabel V-5.** Hasil *Testing* Model 4

Konfigurasi Model	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F-1 Score</i>
Model 4	0,7283	0,8428	0,8676	0,8550

### 5.2.5 Data Hasil Konfigurasi Model 5

Hasil performa model 5 terhadap data latih dan data validasi ditunjukkan pada Gambar V-5. Gambar V-5 menunjukkan grafik akurasi dan *loss* model 5 pada data latih dan data validasi terhadap jumlah *epoch*.



**Gambar V-5.** Grafik Akurasi Model 5

Berdasarkan Gambar V-5, didapatkan akurasi Model 5 dari data latih dan data validasi masing-masing hanya mencapai 12,28% dan 12,48%. Kemudian, *loss* yang didapatkan terhadap data latih dan data validasi masing-masing adalah 12,7712 dan 9,8533.

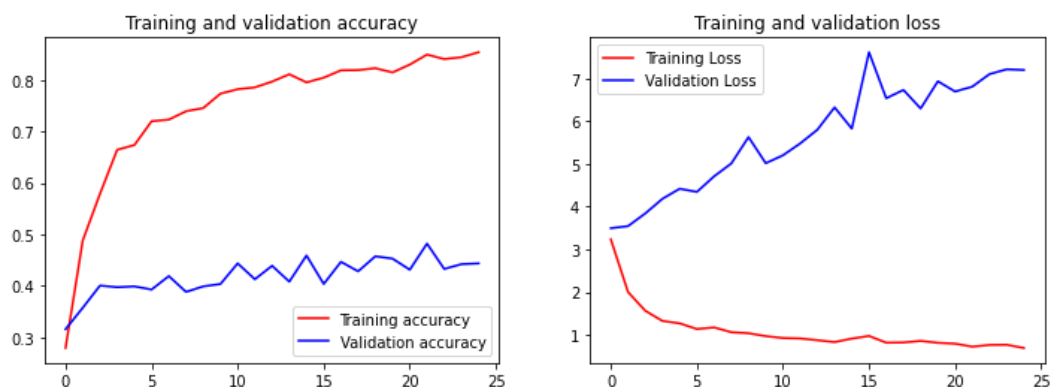
Hasil pengujian performa model terhadap data uji ditunjukkan pada Tabel V-6. Tabel V-6 menunjukkan nilai dari masing-masing metrik evaluasi performa model.

**Tabel V-6.** Hasil *Testing* Model 5

Konfigurasi Model	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F-1 Score</i>
Model 5	0,3652	0,4656	0,3824	0,4199

### 5.2.6 Data Hasil Konfigurasi Model 6

Hasil performa model 6 terhadap data latih dan data validasi ditunjukkan pada Gambar V-6. Gambar V-6 menunjukkan grafik akurasi dan *loss* model 6 pada data latih dan data validasi terhadap jumlah *epoch*.



**Gambar V-6.** Grafik Akurasi Model 6

Berdasarkan Gambar V-6, didapatkan akurasi Model 6 dari data latih dan data validasi masing-masing hanya mencapai 85,42% dan 44,38%. Kemudian, *loss* yang didapatkan terhadap data latih dan data validasi masing-masing adalah 0,6880 dan 7,2055.

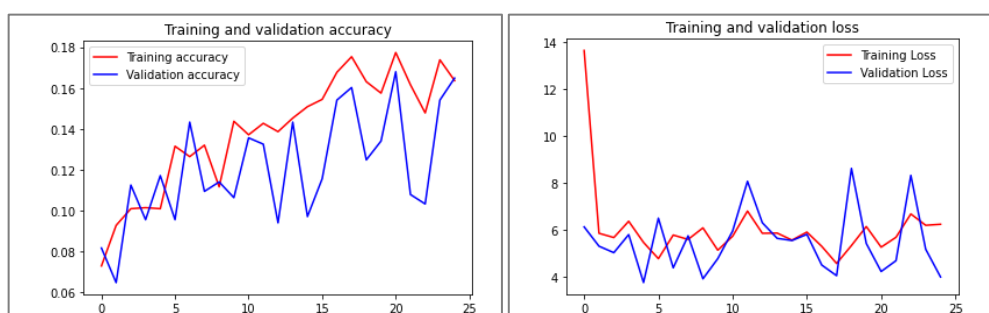
Hasil pengujian performa model terhadap data uji ditunjukkan pada Tabel V-7. Tabel V-7 menunjukkan nilai dari masing-masing metrik evaluasi performa model.

**Tabel V-7.** Hasil *Testing* Model 6

Konfigurasi Model	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F-1 Score</i>
Model 6	0,5898	0,9563	0,6176	0,7505

### 5.2.7 Data Hasil Konfigurasi Model 7

Hasil performa model 7 terhadap data latih dan data validasi ditunjukkan pada Gambar V-7. Gambar V-7 menunjukkan grafik akurasi dan *loss* model 7 pada data latih dan data validasi terhadap jumlah *epoch*.



**Gambar V-7.** Grafik Akurasi Model 7

Berdasarkan Gambar V-7, didapatkan akurasi Model 7 dari data latih dan data validasi masing-masing hanya mencapai 16,36% dan 16,49%. Kemudian, *loss* yang didapatkan terhadap data latih dan data validasi masing-masing adalah 6,2522 dan 4,0044.

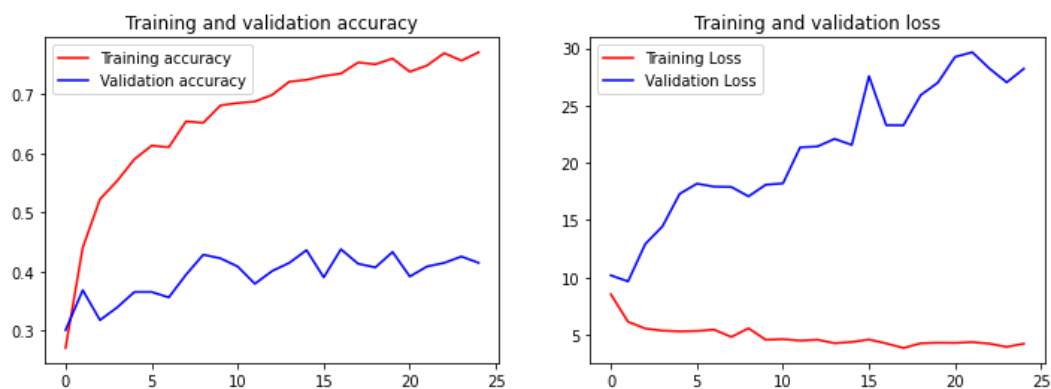
Hasil pengujian performa model terhadap data uji ditunjukkan pada Tabel V-8. Tabel V-8 menunjukkan nilai dari masing-masing metrik evaluasi performa model.

**Tabel V-8.** Hasil *Testing* Model 7

Konfigurasi Model	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F-1 Score</i>
Model 7	0,0164	0,2391	0,1410	0,1774

### 5.2.8 Data Hasil Konfigurasi Model 8

Hasil performa model 8 terhadap data latih dan data validasi ditunjukkan pada Gambar V-8. Gambar V-8 menunjukkan grafik akurasi dan *loss* model 8 pada data latih dan data validasi terhadap jumlah *epoch*.



**Gambar V-8.** Grafik Akurasi Model 8

Berdasarkan Gambar V-8, didapatkan akurasi Model 8 dari data latih dan data validasi masing-masing hanya mencapai 77,17% dan 41,45%. Kemudian, *loss* yang didapatkan terhadap data latih dan data validasi masing-masing adalah 4,2042 dan 28,2249.

Hasil pengujian performa model terhadap data uji ditunjukkan pada Tabel V-9. Tabel V-9 menunjukkan nilai dari masing-masing metrik evaluasi performa model.

**Tabel V-9.** Hasil *Testing* Model 8

<b>Konfigurasi Model</b>	<b><i>Accuracy</i></b>	<b><i>Recall</i></b>	<b><i>Precision</i></b>	<b><i>F-1 Score</i></b>
Model 8	0,4655	0,9120	0,5065	0,6513

### 5.3 Analisis Hasil Penelitian

Setiap variasi konfigurasi model CNN memiliki performa pelatihan, validasi dan pengujian yang berbeda-beda dalam klasifikasi citra wajah bermasker. Tabel V-10 berikut menunjukkan perbandingan performa model berdasarkan data latih dan data validasi. Kemudian, Tabel V-11 menunjukkan perbandingan performa model pada data uji.

**Tabel V-10.** Perbandingan Hasil *Training-Validation* Model

<b>No</b>	<b>Konfigurasi Model</b>	<b><i>Train_Acc</i></b>	<b><i>Val_Acc</i></b>	<b><i>Train_Loss</i></b>	<b><i>Val_Loss</i></b>
0	ResNet50 (Mandal et al., 2021)	60,05%	47,91%	1,5005	2,4092
1	Model 1	16,29%	14,08%	7,5822	7,0848
<b>2</b>	<b>Model 2</b>	<b>96,38%</b>	<b>60,56%</b>	<b>0,1357</b>	<b>2,3069</b>
3	Model 3	18,10%	14,08%	3,4628	3,7043
4	Model 4	93,67%	54,93%	1,0081	8,3073
5	Model 5	12,28%	12,48%	12,7712	9,8533
6	Model 6	85,42%	44,38%	0,6880	7,2055

No	Konfigurasi Model	<i>Train_Acc</i>	<i>Val_Acc</i>	<i>Train_Loss</i>	<i>Val_Loss</i>
7	Model 7	16,36%	16,49%	6,2522	4,0044
8	Model 8	77,17%	41,45%	4,2042	28,2249

Variasi dari setiap konfigurasi model CNN memiliki performa pelatihan dan validasi yang berbeda – beda dalam klasifikasi citra wajah bermasker. Berdasarkan Tabel V-10, Model 2 dan 4 merupakan dua model terbaik. Namun, Model 2 masih lebih baik dibandingkan model 4 dikarenakan nilai akurasi data validasi lebih baik dan memiliki grafik yang lebih baik. Selain itu, Model 2 memiliki akurasi dan *loss* yang lebih baik dibandingkan model acuan, yaitu Model ResNet50 (Mandal et al., 2021). Model 2 memiliki akurasi data latih lebih besar 36,33% dibandingkan model acuan dan memiliki akurasi data validasi lebih besar 12,65% dibandingkan model acuan.

**Tabel V-11.** Perbandingan Hasil *Testing* Model

No	Konfigurasi Model	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F-1 Score</i>
0	ResNet50 (Mandal et al., 2021)	0,4791	0,4719	0,4613	0,4473
1	Model 1	0,1358	0,6471	0,1467	0,2392
<b>2</b>	<b>Model 2</b>	<b>0,7901</b>	<b>0,9142</b>	<b>0,8533</b>	<b>0,8827</b>
3	Model 3	0,1111	0,4500	0,1323	0,2044
4	Model 4	0,7283	0,8428	0,8676	0,8550
5	Model 5	0,3652	0,4656	0,3824	0,4199



No	Konfigurasi Model	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F-1 Score</i>
6	Model 6	0,5898	0,9563	0,6176	0,7505
7	Model 7	0,0164	0,2391	0,1410	0,1774
8	Model 8	0,4655	0,9120	0,5065	0,6513

Model 2 memiliki performa yang lebih baik juga dibandingkan dengan model acuan berdasarkan metrik evaluasi yang digunakan. Model 2 memiliki nilai yang lebih stabil dibandingkan dengan model lain. Model 2 yang dihasilkan memiliki kinerja yang baik berdasarkan metrik evaluasi berupa *accuracy* sebesar 0,7901, *recall* sebesar 0,9142, *precision* sebesar 0,8533, dan *f-1 score* sebesar 0,8827. Model 2 juga memiliki kinerja yang lebih baik secara keseluruhan dibandingkan dengan model acuan.

#### 5.4 Kesimpulan

Terdapat 8 model CNN yang dibangun berdasarkan 8 skenario percobaan dengan konfigurasi berbeda untuk menyelesaikan masalah pengenalan citra wajah bermasker. Variasi konfigurasi dilakukan untuk mencari model dengan performa terbaik. Hasil dari perbandingan performa model menunjukkan Model 2 adalah model terbaik. Akurasi Model 2 dari data latih dan data validasi masing-masing hanya mencapai 96,38% dan 60,56%. Kemudian, *loss* yang didapatkan terhadap data latih dan data validasi masing-masing adalah 0,1357 dan 2,3069. Selain itu, Model 2 yang dihasilkan juga memiliki kinerja yang baik dibandingkan model acuan berdasarkan metrik evaluasi berupa *accuracy* sebesar 79,01%, *recall* sebesar

91,42%, *precision* sebesar 85,33%, dan *f-1 score* sebesar 88,27%. Namun, model terbaik tersebut merupakan model yang tidak menggunakan *additive angular margin loss*.

## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Kesimpulan yang diperoleh setelah dilakukan analisis pada hasil penelitian pada bab sebelumnya adalah sebagai berikut:

1. Perangkat lunak sistem pengenalan citra wajah bermasker menggunakan berbagai model CNN, yaitu InceptionV3 dan ResNet50 dengan dan tanpa pendekatan *additive angular margin loss* berhasil dikembangkan, walaupun model yang dihasilkan tidak lebih baik dibandingkan metode lain tanpa menggunakan *additive angular margin loss*.
2. Berdasarkan hasil penelitian, model terbaik yang dihasilkan adalah Model InceptionV3 tanpa menggunakan *additive angular margin loss* yang dihasilkan memiliki kinerja yang baik dibandingkan model acuan berdasarkan metrik evaluasi berupa *accuracy* sebesar 79,01%, *recall* sebesar 91,42%, *precision* sebesar 85,33%, dan *f-1 score* sebesar 88,27%.

#### 6.2 Saran

Berikut saran untuk penelitian selanjutnya adalah sebagai berikut:

1. Gunakan dataset yang lebih besar dan lebih banyak agar model CNN dapat mempelajari fitur lebih banyak.
2. Diperlukan langkah tambahan untuk mengoptimalkan hasil ekstraksi fitur wajah.

3. Diperlukan studi yang lebih luas pada konfigurasi *layer* sehingga mendapat model CNN yang *fine-tuning* atau model dengan kurva akurasi dan *loss* yang lebih baik.

## DAFTAR PUSTAKA

- Alzu'bi, A., Albalas, F., Al-Hadhrami, T., Younis, L. B., & Bashayreh, A. (2021). Masked face recognition using deep learning: A review. In *Electronics (Switzerland)* (Vol. 10, Issue 21). MDPI. <https://doi.org/10.3390/electronics10212666>
- Anwar, A., & Raychowdhury, A. (2020). *Masked Face Recognition for Secure Authentication*. <http://arxiv.org/abs/2008.11104>
- Boutros, F., Damer, N., Kirchbuchner, F., & Kuijper, A. (2021). *Self-restrained Triplet Loss for Accurate Masked Face Recognition*. <http://arxiv.org/abs/2103.01716>
- Chandra, B. S., Sastry, C. S., Jana, S., & Patidar, S. (2017). Atrial fibrillation detection using convolutional neural networks. *Computing in Cardiology*, 44, 1–4. <https://doi.org/10.22489/CinC.2017.163-226>
- Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (n.d.). *ArcFace: Additive Angular Margin Loss for Deep Face Recognition*. <https://github.com/>
- Ejaz, M. S., Islam, M. R., Sifatullah, M., & Sarker, A. (2019, May 1). Implementation of Principal Component Analysis on Masked and Non-masked Face Recognition. *1st International Conference on Advances in Science, Engineering and Robotics Technology 2019, ICASERT 2019*. <https://doi.org/10.1109/ICASERT.2019.8934543>
- Hakim, Z., & Rizky, R. (2020). Analisis Perancangan Sistem Informasi Pembuatan Paspor Di Kantor Imigrasi Bumi Serpong Damai Tangerang Banten Menggunakan Metode Rational Unified Process. *Jutis (Jurnal Teknik Informatika)*, 6(2), 103–112. <http://ejournal.unis.ac.id/index.php/jutis/article/view/135>
- Hariri, W. (2022). Efficient masked face recognition method during the COVID-19 pandemic. *Signal, Image and Video Processing*, 16(3), 605–612. <https://doi.org/10.1007/s11760-021-02050-w>
- Khan, H. A., Jue, W., Mushtaq, M., & Mushtaq, M. U. (2020). Brain tumor classification in MRI image using convolutional neural network. *Mathematical Biosciences and Engineering*, 17(5), 6203–6216. <https://doi.org/10.3934/MBE.2020328>
- Li, L., Mu, X., Li, S., & Peng, H. (2020). A Review of Face Recognition Technology. *IEEE Access*, 8, 139110–139120. <https://doi.org/10.1109/ACCESS.2020.3011028>
- Li, Y., Guo, K., Lu, Y., & Liu, L. (2021). Cropping and attention based approach for masked face recognition. *Applied Intelligence*, 51(5), 3012–3025. <https://doi.org/10.1007/s10489-020-02100-9>
- Lin, C., Li, L., Luo, W., Wang, K. C. P., & Guo, J. (2019). Transfer learning based

- traffic sign recognition using inception-v3 model. *Periodica Polytechnica Transportation Engineering*, 47(3), 242–250. <https://doi.org/10.3311/PPtr.11480>
- Mandal, B., Okeukwu, A., & Theis, Y. (2021). *Masked Face Recognition using ResNet-50*. <http://arxiv.org/abs/2104.08997>
- Montero, D., Nieto, M., Leskovsky, P., & Aginako, N. (2021). *Boosting Masked Face Recognition with Multi-Task ArcFace*. <http://arxiv.org/abs/2104.09874>
- Mukti, I. Z., & Biswas, D. (2019). Transfer Learning Based Plant Diseases Detection Using ResNet50. *2019 4th International Conference on Electrical Information and Communication Technology, EICT 2019, December*, 1–6. <https://doi.org/10.1109/EICT48899.2019.9068805>
- Novandya, A. (2017). Penerapan Algoritma Klasifikasi Data Mining C4.5 pada Dataset Cuaca Wilayah Bekasi. *KNiST*, 368–372.
- Nurmaini, S., Tondas, A. E., Darmawahyuni, A., Rachmatullah, M. N., Umi Partan, R., Firdaus, F., Tutuko, B., Pratiwi, F., Juliano, A. H., & Khoirani, R. (2020). Robust detection of atrial fibrillation from short-term electrocardiogram using convolutional neural networks. *Future Generation Computer Systems*, 113, 304–317. <https://doi.org/10.1016/j.future.2020.07.021>
- Pang, B., Nijkamp, E., & Wu, Y. N. (2020). Deep Learning With TensorFlow: A Review. *Journal of Educational and Behavioral Statistics*, 45(2), 227–248. <https://doi.org/10.3102/1076998619872761>
- Sanchez, S. A., Romero, H. J., & Morales, A. D. (2020). A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework. *IOP Conference Series: Materials Science and Engineering*, 844(1). <https://doi.org/10.1088/1757-899X/844/1/012024>
- Saputro, I. W., & Sari, B. W. (2020). Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa. *Creative Information Technology Journal*, 6(1), 1. <https://doi.org/10.24076/citec.2019v6i1.178>
- Shafiee, S., Wautelet, Y., Hvam, L., Sandrin, E., & Forza, C. (2020). Scrum versus Rational Unified Process in facing the main challenges of product configuration systems development. *Journal of Systems and Software*, 170, 110732. <https://doi.org/10.1016/j.jss.2020.110732>
- Tyagi, V. (2018). Understanding Digital Image Processing. *Understanding Digital Image Processing, November*. <https://doi.org/10.1201/9781315123905>
- Wu, Y., & Ji, Q. (2019). Facial Landmark Detection: A Literature Survey. *International Journal of Computer Vision*, 127(2), 115–142. <https://doi.org/10.1007/s11263-018-1097-z>
- Xiong, Z., Stiles, M. K., & Zhao, J. (2017). Robust ECG signal classification for detection of atrial fibrillation using a novel neural network. *Computing in Cardiology*, 44(Figure 1), 1–4. <https://doi.org/10.22489/CinC.2017.066-138>
- Zhang, H., Li, Q., Sun, Z., & Liu, Y. (2018). Combining Data-Driven and Model-

Driven Methods for Robust Facial Landmark Detection. *IEEE Transactions on Information Forensics and Security*, 13(10), 2409–2422.  
<https://doi.org/10.1109/TIFS.2018.2800901>

## LAMPIRAN

### Lampiran 1. Kode Program

#### app.py

```
from flask import Flask, render_template, Response, request
from support import Support
import cv2 as cv

app = Flask(__name__)

app.config['SEND_FILE_MAX_AGE_DEFAULT'] = 1

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/recognition', methods=['POST', 'GET'])
def recognition():
    predict = None
    print(faceDetect)
    if request.method == 'POST':
        img = request.files['face']
        img.save('static/uploaded.jpg', 0)

        image = cv.imread('static/uploaded.jpg',
cv.IMREAD_COLOR)

        predict = Support().face_recognition(image)
        return render_template('recognition.html',
predict=predict)

    return render_template('recognition.html',
predict=predict)

app.run(debug=True)
```

#### support.py

```
import cv2 as cv
import numpy as np
import dlib
from tensorflow.keras.models import load_model

class Support(object):

    def extract_feat(self, image):
        detector = dlib.get_frontal_face_detector()
        predictor =
dlib.shape_predictor('model/shape_predictor_68_face_landmarks
.dat')
        shape = 0
```



```

        rects = detector(image, 1)

        for rect in rects:
            shape = predictor(image, rect)
            shape_np = np.zeros((68, 2), dtype="int")
            for i in range(0, 68):
                shape_np[i] = (shape.part(i).x,
shape.part(i).y)
            shape = shape_np

            for i, (x, y) in enumerate(shape):
                cv.circle(image, (x, y), 1, (0, 0, 255), -1)
        return image

    def face_recognition(self, image):
        IMG_SIZE = 150
        img_feat = self.extract_feat(image)
        fix_img = cv.resize(img_feat, (IMG_SIZE, IMG_SIZE),
interpolation=cv.INTER_AREA)
        fix_img = fix_img / 255
        fix_img = fix_img.reshape(-1, IMG_SIZE, IMG_SIZE, 3)
        model = load_model('model/mobilenet_model.h5')
        pred = model.predict(fix_img)
        label_map = ['duhaitao', 'houminghao', 'jingtian',
'linxinru', 'luhan', 'masu', 'matianyu', 'wuyifan',
'xuezhiquan', 'xuweizhou', 'yangyang', 'yuanshanshan',
'zhangruoyun', 'zhangyixing', 'zhangyuxi']
        pred = np.argmax(pred)
        final_pred = label_map[pred]
        return final_pred

```

#### base.html

```

<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1">
        <title>Masked-Face Recognition App</title>

        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/b
ootstrap.min.css" rel="stylesheet">
        <link rel="stylesheet" href="{ url_for('static',
filename='css/style.css') }}">

        <script src="https://kit.fontawesome.com/1d7a98a84b.js"
crossorigin="anonymous"></script>
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/boo
tstrap.bundle.min.js"></script>

    </head>

```

```

<body>

  <div>
    <div class="text-center">
      {% block heading %}

      {% endblock %}
    </div>
  </div>

  <div class="container">
    {% block content %}

    {% endblock %}
  </div>

  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
  <script
src="https://unpkg.com/aos@next/dist/aos.js"></script>
  <script src="{{url_for('static',
filename='js/script.js')}}" ></script>

  </body>
</html>

```

### index.html

```

{% extends 'base.html' %}

{% block heading %}
  
{% endblock %}

{% block content %}
<h2 class="text-center">Sistem Pengenalan Citra Wajah
Bermasker Menggunakan CNN dengan Pendekatan Angular Margin
Loss</h2>
<div class="text-center"><a class="btn btn-primary" href="{{
url_for('recognition') }}">Start</a></div>
{% endblock %}

```

### recognition.html

```

{% extends 'base.html' %}

{% block heading %}
  
{% endblock %}

{% block content %}

```

```

<div class="container">
  <div>
    <h1 class="title-two">Choose a Picture</h1>
    <form action="" method="POST"
enctype="multipart/form-data">
      <input class="form-control" type="file"
name="face" accept="images/*" required />
      <div class="text-center"><button
type="submit" class="btn btn-info mt-3 fw-
bold">Predict</button></div>
    </form>
  </div>

  <div id="menu">
    {% if predict != None %}
      <div>
        <div class="box-content primary">
          
          <br> The Identity is: {{ predict }}
        </div>
      </div>
    {% elif predict == None %}
    {% endif %}
  </div>
</div>
{% endblock %}

```

## style.css

```

:root {
  --blue1: #EDF9FC;
  --blue2: #D1F4FA;
  --blue3: #53CDE2;
  --black: #333333;
  --white: #FFFFFF;
  --success: #99FF99;
  --fail: #990000;
  --font1: "roboto";
  --font2: "montserrat";
}

/*Font*/
@font-face {
  font-family: 'roboto';
  src: url('font/roboto.woff') format('woff');
  font-weight: normal;
  font-style: normal;
}

@font-face {
  font-family: 'montserrat';
  src: url('font/montserrat.woff') format('woff');
  font-weight: normal;
}

```

```

    font-style: normal;
}

@media (min-width: 577px){
    body {
        margin: 0;
        padding: 0;
        font-family: var(--font1);
        font-size: 16px;
        color: var(--black);
    }

    .head{
        background-image: linear-gradient(180deg, var(--
blue2), var(--blue3));
    }

    .head img{
        width: 300px !important;
        margin-bottom: 48px;
    }

    .content{
        position: relative;
        top: -20px;
        background-color: var(--white);
        border-top-left-radius: 25px;
        border-top-right-radius: 25px;
        overflow: auto;
    }

    .title-one, .title-two{
        font-family: var(--font2);
        font-weight: bold;
        text-align: center;
        margin-top: 30px;
    }

    .title-two{
        font-size: 20px;
    }

    .box-content{
        background-color: var(--blue1);
        border-radius: 10px;
        padding: 20px;
        text-align: justify;
        margin: 20px auto;
        font-size: 18px;
        width: 1000px;
    }

    .padding-content{
        padding: 90px !important;
    }

```

```

}

.padding-content2{
    padding: 50px !important;
}

.box-content.important{
    background-color: var(--blue2);
}

.box-content.success{
    background-color: var(--success);
    font-weight:bold;
}

.box-content.fail{
    background-color: var(--fail);
    font-weight:bold;
    color: var(--white);
}

.box-content img{
    width: 50px;
}

.box-content.icon-tips{
    width:auto !important;
}

.box-content.icon-tips img{
    width:100px !important;
}

.item-number{
    font-size: 100px;
    text-align: center;
    color: var(--blue3);
}

.icon-detection{
    width: 100px !important;
}

#menu img{
    width: 150px !important;
}

form{
    width: 1000px;
    margin: 20px auto;
}

.img-tips{
    height: 112.27px !important;
}

```

```

        line-height: 112.27px !important;
    }
}

@media (max-width: 576px) {
    body {
        margin: 0;
        padding: 0;
        font-family: 'roboto';
        font-size: 14px;
        color: var(--black);
    }

    /* NAVBAR */
    .navbar {
        background-color: var(--blue2);
        margin: 0 auto;
        border-top-left-radius: 15px;
        border-top-right-radius: 15px;
        padding: 15px 0;
    }

    .navbar .nav-item .nav-link {
        color: var(--white);
        font-size: 25px;
        background-color: var(--blue3);
        border-radius: 10px;
        width: 50px;
        margin: 0 auto !important;
    }

    .main-menu {
        position: absolute;
        top: -20px;
        font-size: 30px !important;
    }

    .navbar .nav-item .nav-link:hover,
    .navbar .nav-item .nav-link.active {
        color: var(--blue3);
        background-color: var(--blue1);
    }

    .head {
        background-image: url('../images/bg-head.png');
        background-size: cover;
        height: 220px;
    }

    .content {
        position: relative;
        top: -20px;
        height: 460px;
    }
}

```

```

        background-color: var(--white);
        border-top-left-radius: 25px;
        border-top-right-radius: 25px;
        overflow: auto;
    }

    .title-one{
        font-family: var(--font2);
        font-weight:bold;
        text-align:center;
        margin-top: 20px;
    }

    .title-two{
        font-family: var(--font2);
        font-weight:bold;
        text-align:left;
        margin-top: 20px;
    }

    .box-content{
        background-color: var(--blue1);
        border-radius: 10px;
        padding: 10px;
        text-align: justify;
        margin: 20px 0;
    }

    .box-content.important{
        background-color: var(--blue2);
    }

    .box-content.success{
        background-color: var(--success);
        font-weight:bold;
    }

    .box-content.fail{
        background-color: var(--fail);
        font-weight:bold;
        color: var(--white);
    }

    .item-number{
        font-size: 60px;
        text-align: center;
        color: var(--blue3);
    }
}

```