

**TESIS**  
**EVALUASI PENGENALAN WAJAH MENGGUNAKAN**  
**FACENET PADA PEGAWAI DINAS KOMUNIKASI DAN**  
**INFORMATIKA KOTA SAMARINDA**



disusun oleh

**AJI SAID WAHYUDI HIDAYAT**

**22.55.2314**

**Konsentrasi : Business Intelligence**

**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS AMIKOM YOGYAKARTA**  
**YOGYAKARTA**  
**2025**

**TESIS**  
**EVALUASI PENGENALAN WAJAH MENGGUNAKAN**  
**FACENET PADA PEGAWAI DINAS KOMUNIKASI DAN**  
**INFORMATIKA KOTA SAMARINDA**

**EVALUATION OF FACIAL RECOGNITION USING**  
**FACENET ON EMPLOYEES OF THE COMMUNICATION**  
**AND INFORMATICS DEPARTMENT OF SAMARINDA CITY**

Diajukan untuk memenuhi salah satu syarat mencapai derajat Pascasarjana  
Program Studi S2 PJJ Teknik Informatika



disusun oleh

**AJI SAID WAHYUDI HIDAYAT**

**22.55.2314**

**Konsentrasi : Business Intelligence**

**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS AMIKOM YOGYAKARTA**  
**YOGYAKARTA**

**2025**

**HALAMAN PERSETUJUAN**

**EVALUASI PENGENALAN WAJAH MENGGUNAKAN FACENET PADA  
PEGAWAI DINAS KOMUNIKASI DAN INFORMATIKA KOTA  
SAMARINDA**

**EVALUATION OF FACIAL RECOGNITION USING FACENET ON  
EMPLOYEES OF THE COMMUNICATION AND INFORMATICS  
DEPARTMENT OF SAMARINDA CITY**

yang disusun dan diajukan oleh

**Aji Sald Wahyudi Hidayat**

**22.55.2314**

telah disetujui oleh Dosen Pembimbing Tesis  
pada tanggal 7 November 2025

**Dosen Pembimbing,**

**Ainul Yaqin, M.Kom.**

**NIK. 190302255**



**HALAMAN PENGESAHAN**

**EVALUASI PENGENALAN WAJAH MENGGUNAKAN FACENET PADA  
PEGAWAI DINAS KOMUNIKASI DAN INFORMATIKA KOTA  
SAMARINDA**

**EVALUATION OF FACIAL RECOGNITION USING FACENET ON  
EMPLOYEES OF THE COMMUNICATION AND INFORMATICS  
DEPARTMENT OF SAMARINDA CITY**

yang disusun dan diajukan oleh

**Ajl Said Wahyudi Hidayat**

**22.55.2314**

Telah dipertahankan di depan Dewan Penguji  
pada tanggal 7 November 2025

**Susunan Dewan Penguji**

**Nama Penguji**

**Tanda Tangan**

Robert Marco, S.T., M.T., Ph.D.  
NIK. 190302228

Dr. Ferry Wahyu Wibowo, S.Si., M.Cs.  
NIK. 190302235

Arief Setyanto, S.Si., M.T., Ph.D.  
NIK. 190302036



Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer  
Tanggal 7 November 2025

**DEKAN FAKULTAS ILMU KOMPUTER**

Prof. Dr. Kusriani, M.Kom.  
NIK. 190302106



## HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Aji Said Wahyudi Hidayat  
NIM : 22.55.2314

Menyatakan bahwa Tesis dengan judul berikut:

**Evaluasi Pengenalan Wajah Menggunakan Facenet pada Pegawai Dinas Komunikasi dan Informatika Kota Samarinda**

Dosen Pembimbing 1 : Arief Setyanto, S.Si., M.T., Ph.D.

Dosen Pembimbing 2 : Aintil Yaqin, M.Kom.

1. Karya tulis ini adalah benar-benar **ASLI** dan **BELUM PERNAH** diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian **SAYA** sendiri, tanpa bantuan pihak lain kecuali arahan dari Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab **SAYA**, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini **SAYA** buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka **SAYA** bersedia menerima **SANKSI AKADEMIK** dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 7 November 2025

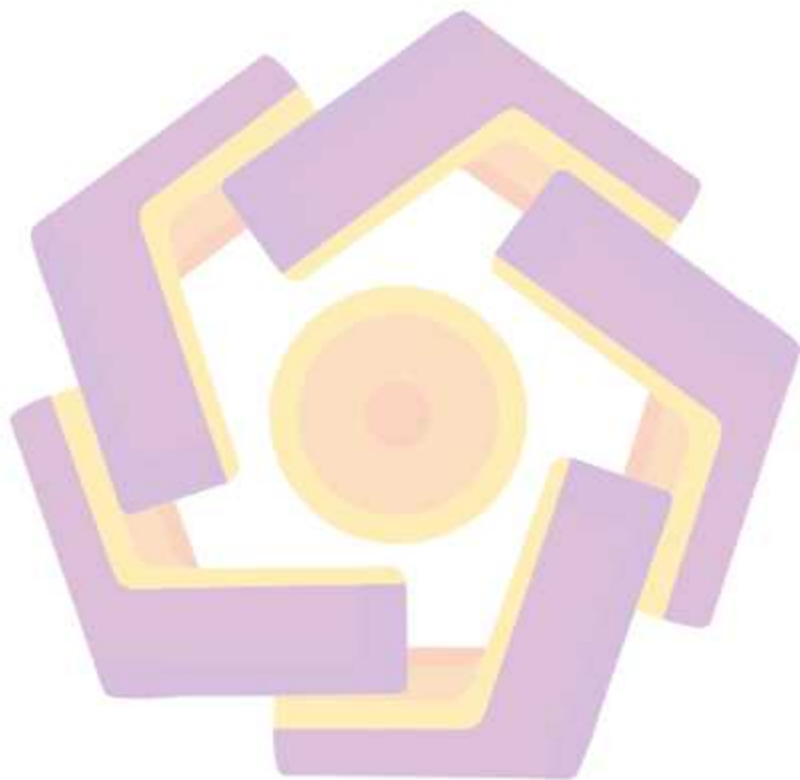
Yang Menyatakan,



Aji Said Wahyudi Hidayat

## **HALAMAN PERSEMBAHAN**

Thesis yang disusun ini dipersembahkan untuk keluarga, dan rekan sejawat yang telah mendukung dan menyukseskan penelitian hingga penulisan thesis ini.



## KATA PENGANTAR

*Assalamu'alaikum Warahmatullahi Wabarakatuh.*

Pertama-tama, puji syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan tesis berjudul "Evaluasi Pengenalan Wajah Menggunakan *FaceNet* pada Pegawai Dinas Komunikasi dan Informatika Kota Samarinda" dengan baik. Tesis ini bertujuan untuk memperoleh gelar Magister Teknik Informatika (S2) di Universitas Amikom Yogyakarta.

Penulis menyadari bahwa dalam proses penelitian hingga penyusunan tesis ini masih terdapat berbagai kekurangan. Namun demikian, penulis berharap karya ini dapat memberikan kontribusi positif serta menjadi referensi dan wawasan baru dalam penerapan sistem presensi berbasis pengenalan wajah.

Pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan, baik moril maupun teknis dalam penyusunan tesis ini, terutama kepada:

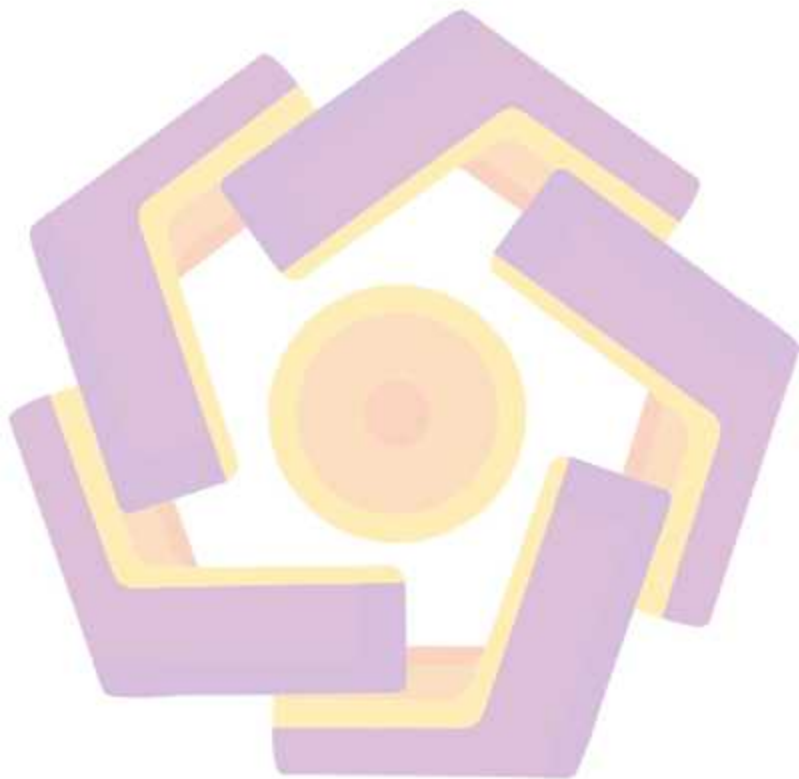
1. Keluarga tercinta atas doa, semangat, dan dukungan yang tiada henti selama proses penulisan tesis ini berlangsung.
2. Bapak Prof. Arief Setyanto, S.Si., M.T., Ph.D., dan bapak Ainul Yaqin, M.Kom., selaku dosen pembimbing yang turut membantu kelancaran dalam proses penulisan tesis ini.
3. Bapak Robert Marco, S.T., M.T., Ph.D., dan Bapak Prof. Dr. Ferry Wahyu Wibowo, S.Si., M.Cs., selaku dosen penguji yang turut memberikan masukan, kritik, dan saran, agar penelitian ini menjadi lebih baik.
4. Para karyawan Universitas Amikom Yogyakarta, yang turut membantu secara administrasi dalam menyelesaikan tesis ini.
5. Rekan kerja di Dinas Komunikasi dan Informatika Kota Samarinda yang turut membantu secara teknis dalam proses penelitian ini.

Akhir kata, penulis berharap tesis ini dapat bermanfaat bagi semua pihak yang membacanya. Segala kebaikan datang dari Allah SWT dan segala kekurangan adalah dari diri penulis sendiri.

*Billahi taufiq wal hidayah,  
Wassalamu 'alaikum warahmatullahi wabarakatuh.*

Yogyakarta, 27 November 2025

Penulis

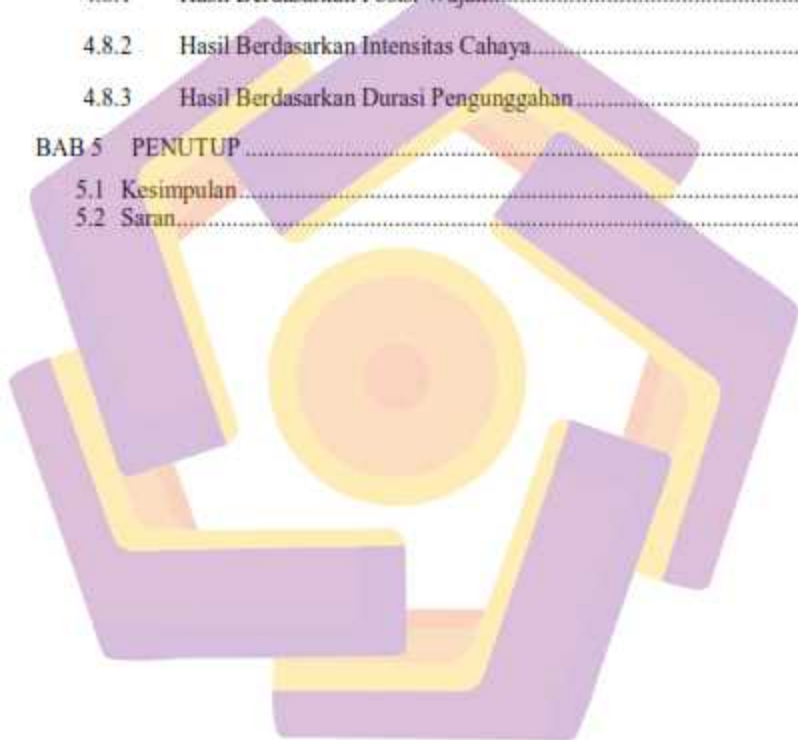


## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERSETUJUAN .....	ii
HALAMAN PENGESAHAN .....	iii
HALAMAN PERNYATAAN KEASLIAN TESIS .....	iv
HALAMAN PERSEMBAHAN .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI .....	viii
DAFTAR TABEL .....	xi
DAFTAR GAMBAR .....	xii
DAFTAR LAMPIRAN .....	xiii
DAFTAR LAMBANG DAN SINGKATAN .....	xiv
DAFTAR ISTILAH .....	xv
INTISARI .....	xvii
ABSTRACT .....	xviii
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	5
1.3 Batasan Masalah .....	6
1.4 Tujuan Penelitian .....	6
1.5 Manfaat Penelitian .....	6
1.6 Hipotesis .....	7
BAB 2 TINJAUAN PUSTAKA .....	8
2.1 Tinjauan Pustaka .....	8
2.2 Keaslian Penelitian .....	14
2.3 Landasan Teori .....	26
2.3.1 <i>Python</i> .....	26
2.3.2 <i>FaceNet</i> .....	27
2.3.3 <i>Android</i> .....	32
2.3.4 Klasifikasi Biner .....	34
BAB 3 METODE PENELITIAN .....	35
3.1 Jenis, Sifat, dan Pendekatan Penelitian .....	35
3.2 Metode Pengumpulan Data .....	35

3.2.1	Populasi dan Sampel.....	36
3.2.2	Teknik Pengumpulan Data.....	36
3.3	Metode Analisis Data.....	41
3.4	Alur Penelitian.....	41
3.4.1	Alat yang Digunakan.....	41
3.4.2	Bahasa Pemrograman.....	41
3.4.3	<i>Server</i> .....	42
3.4.4	Pengukuran Cahaya.....	42
3.4.5	Pengumpulan <i>Dataset</i> .....	42
3.4.6	Evaluasi model <i>FaceNet</i> pada <i>dataset</i> publik.....	42
3.4.7	Evaluasi model <i>FaceNet</i> pada <i>dataset custom</i> .....	43
3.4.8	Implementasi model <i>FaceNet</i> sebagai layanan.....	43
3.4.9	Implementasi <i>front end</i> menggunakan aplikasi <i>mobile</i> .....	43
3.4.10	Evaluasi sistem presensi pada kasus nyata di Dinas Komunikasi dan Informatika Kota Samarinda.....	44
<b>BAB 4 HASIL PENELITIAN DAN PEMBAHASAN.....</b>		<b>47</b>
4.1	Pengantar.....	47
4.2	Struktur <i>File</i> .....	47
4.3	Struktur Basis Data.....	48
4.3.1	Tabel <i>users</i> .....	49
4.3.2	Tabel <i>face_encodings</i> .....	51
4.3.3	Tabel <i>attendance</i> .....	51
4.3.4	Relasi antar tabel.....	52
4.4	Arsitektur.....	53
4.4.1	<i>Front End</i> .....	53
4.4.2	<i>Back End</i> .....	59
4.4.3	API (Application Programming Interface).....	62
4.5	Evaluasi Model <i>FaceNet</i> untuk Pengenalan Wajah.....	65
4.6	Implementasi Model <i>FaceNet</i> sebagai Service pada <i>Server</i> .....	68

4.7	Cara Kerja Sistem Presensi Berbasis Pengenalan Wajah.....	71
4.7.1	Autentikasi .....	72
4.7.2	Presensi .....	75
4.7.3	Halaman lain .....	76
4.8	Evaluasi Sistem Presensi pada Kasus Nyata di Dinas Komunikasi Dan Informatika Kota Samarinda.....	79
4.8.1	Hasil Berdasarkan Posisi Wajah.....	80
4.8.2	Hasil Berdasarkan Intensitas Cahaya.....	83
4.8.3	Hasil Berdasarkan Durasi Pengunggahan.....	84
<b>BAB 5</b>	<b>PENUTUP</b> .....	<b>87</b>
5.1	Kesimpulan.....	87
5.2	Saran.....	88



## DAFTAR TABEL

Tabel 2.1 Matriks literatur review dan posisi penelitian	14
Tabel 2.2 Beberapa kasus penggunaan khusus <i>FaceNet</i>	32
Tabel 3.1 Spesifikasi telepon genggam untuk mengambil data latih wajah	36
Tabel 3.2 Statistik Data Sampel Wajah	39
Tabel 4.1 Alur <i>file</i> project program	47
Tabel 4.2 Daftar kolom yang mengisi tabel <i>users</i>	49
Tabel 4.3 Daftar kolom yang mengisi tabel <i>face_encodings</i>	51
Tabel 4.4 Daftar kolom yang mengisi tabel <i>attendance</i>	52
Tabel 4.5 Daftar <i>endpoint API</i>	62
Tabel 4.6 Hasil pengujian dengan <i>dataset</i> publik	66
Tabel 4.7 Hasil pengujian dengan <i>dataset custom</i>	68

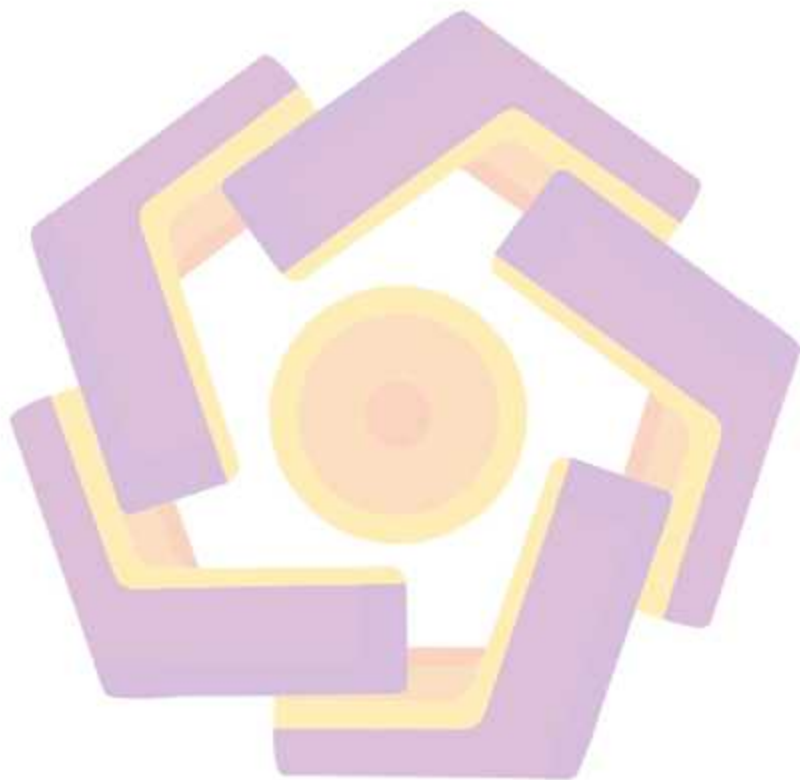


## DAFTAR GAMBAR

Gambar 3.1	Telepon genggam yang digunakan untuk memotret data latih wajah	39
Gambar 3.2	Beberapa sampel wajah yang terdiri dari (urutan dari kiri ke kanan) posisi tegap, kanan sudut $30^{\circ}$ , kanan sudut $60^{\circ}$ , kanan sudut $90^{\circ}$ , kiri sudut $30^{\circ}$ , kiri sudut $60^{\circ}$ , kiri sudut $90^{\circ}$ , atas, bawah, dan tegap.	40
Gambar 3.3	Contoh pose wajah yang dijadikan data uji, dengan tiga posisi wajah berbeda, dan berada pada cahaya terang	40
Gambar 3.4	Alur Sistem Presensi Berbasis Pengenalan Wajah	46
Gambar 4.1	Struktur Basis Data untuk sistem presensi berbasis pengenalan wajah	49
Gambar 4.2	Rancangan halaman autentikasi sistem presensi berbasis pengenalan wajah (dari kiri ke kanan; <i>login</i> , <i>register</i> , generasi kata sandi, dan daftar data latih wajah)	55
Gambar 4.3	Rancangan tampilan halaman dasbor	56
Gambar 4.4	Rancangan tampilan proses presensi berbasis pengenalan wajah (dari kiri ke kanan; halaman eksekusi presensi, dan hasil presensi)	57
Gambar 4.5	Rancangan tampilan <i>resume</i> presensi per pegawai	57
Gambar 4.6	Rancangan tampilan halaman profil	58
Gambar 4.7	Interaksi antar <i>front end</i>	58
Gambar 4.8	Alur <i>Endpoint API</i> berdasarkan hak akses	65
Gambar 4.9	Sampel citra wajah dari <i>dataset</i> publik	66
Gambar 4.10	Salah satu perbandingan citra wajah latih dan uji, dimana keluaran yang dihasilkan yaitu tidak cocok	67
Gambar 4.11	Tangkapan layar proses eksekusi <i>endpoint upload/test</i> pada saat pengujian melalui lingkungan lokal	69
Gambar 4.12	Respon registrasi berhasil	70
Gambar 4.13	Respon berhasil unggah data latih wajah (dua dari 10 respon yang ditampilkan)	70
Gambar 4.14	Respon keberhasilan presensi wajah	71
Gambar 4.15	Proses registrasi akun	74
Gambar 4.16	Proses login (kiri), dan tampilan halaman dasbor (kanan)	75
Gambar 4.17	Proses eksekusi pengenalan wajah	76
Gambar 4.18	Halaman <i>resume per user</i>	77
Gambar 4.19	Halaman profil	78
Gambar 4.20	Hubungan antara aplikasi dengan komponen <i>back end</i>	79
Gambar 4.21	Grafik Hasil Pengenalan Wajah Berdasarkan Posisi Wajah	80
Gambar 4.22	Tangkapan layar hasil presensi keluar	82
Gambar 4.23	Tangkapan layar ketika presensi gagal	82
Gambar 4.24	Wajah yang difoto dalam tiga kategori cahaya yang berbeda, dari kiri ke kanan antara lain cahaya gelap, remang, dan terang.	83
Gambar 4.25	Grafik Hasil Pengenalan Wajah Berdasarkan Intensitas Cahaya	84
Gambar 4.26	Grafik Hasil Pengenalan Wajah Berdasarkan Uji Latensi (lama proses pengenalan wajah)	85

## DAFTAR LAMPIRAN

Lampiran 1 Kode Program (utils/helpers.py)	92
Lampiran 2 Kode Program (app.py)	97



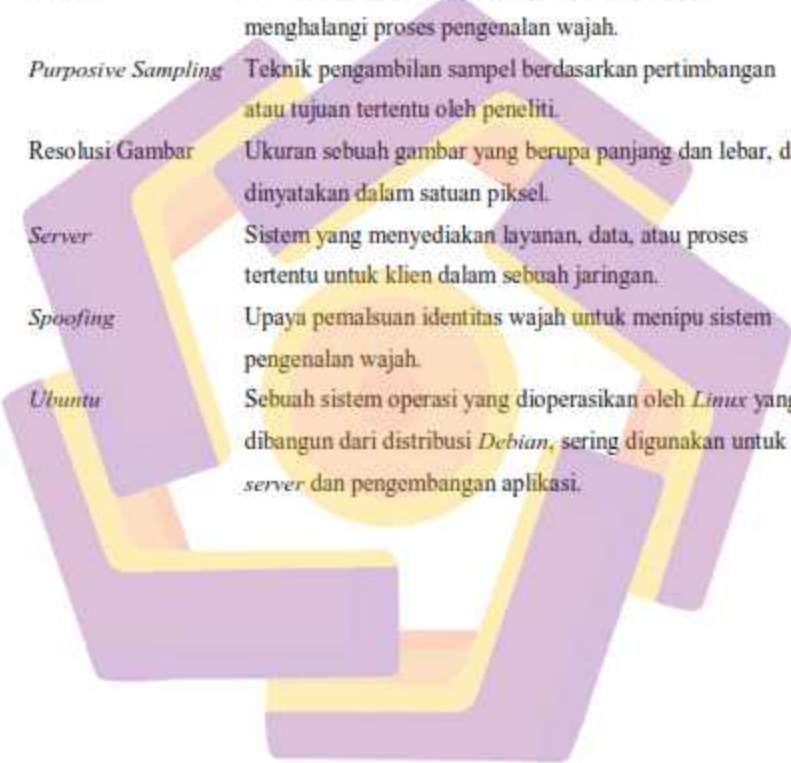
## DAFTAR LAMBANG DAN SINGKATAN



<i>API</i>	<i>Application Programming Interface</i>
<i>CNN</i>	<i>Convolutional Neural Network</i>
<i>FN</i>	<i>False Negative</i>
<i>FP</i>	<i>False Positive</i>
<i>HD</i>	<i>High Definition</i>
<i>LBP</i>	<i>Local Binary Pattern</i>
<i>LBPH</i>	<i>Local Binary Pattern Histogram</i>
<i>LDA</i>	<i>Linear Discriminant Analysis</i>
<i>LFW</i>	<i>Labeled Faces in the Wild</i>
<i>MTCNN</i>	<i>Multi-task Cascaded Convolutional Networks</i>
<i>PCA</i>	<i>Principal Component Analysis</i>
<i>TP</i>	<i>True Positive</i>
<i>TN</i>	<i>True Negative</i>
<i>UI</i>	<i>User Interface</i>

## DAFTAR ISTILAH

<i>Bandwidth</i>	Kapasitas maksimum data yang dapat di transfer melalui sebuah jaringan dalam jangka waktu tertentu.
<i>Dart</i>	Bahasa pemrograman yang digunakan untuk membuat sebuah program yang bersifat <i>multiplatform</i> , seperti <i>web</i> , <i>mobile</i> , <i>desktop</i> , dan <i>back end</i> .
<i>Dataset Custom</i>	<i>Dataset</i> yang telah di buat sendiri oleh peneliti atau pengembang untuk kebutuhan tertentu.
<i>Dataset Publik</i>	<i>Dataset</i> yang telah di publikasikan secara umum dan dapat di akses melalui sebuah halaman <i>website</i> atau repositori.
Deteksi Wajah	Proses untuk mendeteksi keberadaan wajah seseorang baik pada gambar maupun <i>video</i> .
Ekstraksi Fitur	Proses konversi data mentah menjadi sebuah fitur numerik yang lebih ringkas dan relevan untuk <i>machine learning</i> atau <i>deep learning</i> .
<i>Embedding</i>	Nilai numerik yang mewakili ciri khas wajah seseorang untuk tujuan Identifikasi dan verifikasi.
<i>Endpoint</i>	<i>URL</i> atau <i>path</i> dari <i>API</i> untuk melakukan proses komunikasi antara <i>client</i> dan <i>server</i> .
<i>Face Recognition</i>	Proses Identifikasi identitas seseorang secara otomatis berdasarkan fitur wajah yang terdeteksi dari citra atau <i>video</i> .
<i>Flutter</i>	<i>Framework</i> berbasis bahasa pemrograman <i>Dart</i> untuk membangun aplikasi yang bersifat <i>multiplatform</i> .
Kontrol Akses	Pengaturan hak akses pengguna untuk menentukan siapa yang berhak mengakses sistem.
<i>Lux / Lumen</i>	Satuan untuk mengukur intensitas cahaya ( <i>lux</i> mengukur cahaya yang diterima permukaan, sedangkan <i>lumen</i> mengukur total cahaya yang dipancarkan sumber cahaya).



<i>Machine Learning</i>	Cabang kecerdasan buatan yang memungkinkan komputer mempelajari pola dari data untuk membuat prediksi atau pengambilan keputusan.
<i>Mobile App</i>	Program perangkat lunak yang dirancang untuk berjalan pada perangkat telepon genggam.
Oklusi	Kondisi ketika ada objek yang tertutup, sehingga menghalangi proses pengenalan wajah.
<i>Purposive Sampling</i>	Teknik pengambilan sampel berdasarkan pertimbangan atau tujuan tertentu oleh peneliti.
Resolusi Gambar	Ukuran sebuah gambar yang berupa panjang dan lebar, dan dinyatakan dalam satuan piksel.
<i>Server</i>	Sistem yang menyediakan layanan, data, atau proses tertentu untuk klien dalam sebuah jaringan.
<i>Spoofing</i>	Upaya pemalsuan identitas wajah untuk menipu sistem pengenalan wajah.
<i>Ubuntu</i>	Sebuah sistem operasi yang dioperasikan oleh <i>Linux</i> yang dibangun dari distribusi <i>Debian</i> , sering digunakan untuk <i>server</i> dan pengembangan aplikasi.

## INTISARI

Penelitian ini bertujuan untuk melakukan evaluasi sistem presensi menggunakan pengenalan wajah berbasis aplikasi *mobile*. Sistem pengenalan wajah di bangun menggunakan pustaka *FaceNet* pada layanan yang disiapkan di *server*. Sistem tersebut di uji di lingkungan Dinas Komunikasi dan Informatika Kota Samarinda terhadap 40 subyek pegawai dengan sampel per pegawai sebanyak 10 foto. Pengujian dilakukan dalam berbagai kondisi pencahayaan dan pose. Evaluasi dilakukan terhadap ketepatan pengenalan wajah pada saat presensi masuk, dan presensi keluar. Namun, pengujian ini terbatas untuk pengguna telepon genggam dengan sistem operasi *Android*, dan tidak semua telepon genggam berbasis *Android* dapat menjalankan sistem ini dengan baik. Analisis dilakukan dengan metode kuantitatif, yaitu melihat jumlah eksekusi foto yang berjalan berdasarkan posisi wajah, pencahayaan, dan waktu eksekusi.

Dari 18 orang yang telah dilakukan pengujian, didapati hasil pengujian yaitu berdasarkan posisi wajah, posisi tegap dan kiri menghasilkan presensi masuk lebih baik. sedangkan berdasarkan intensitas cahaya, hasil yang lebih efektif yaitu pada intensitas antara 50-99 lux, dan waktu tercepat dalam proses presensi yaitu kurang dari satu menit.

Hasil penelitian menunjukkan bahwa sistem pengenalan wajah dapat berjalan efektif di berbagai posisi wajah, selain itu pencahayaan yang lebih memadai mendukung keberhasilan pengenalan wajah tersebut, dan proses pengunggahan dapat dilakukan kurang dari satu menit.

**Kata kunci:** Pengenalan Wajah, *FaceNet*, Aplikasi *Mobile*, Sistem Presensi.

## ABSTRACT

*This study aims to evaluate an attendance system using mobile application-based facial recognition. The facial recognition system was built using the FaceNet library on a service set up on a server. The system was tested in the Samarinda City Communication and Information Agency environment on 40 employee subjects with a sample of 10 photos per employee. Testing was carried out in various lighting conditions and poses. The evaluation was conducted on the accuracy of facial recognition during attendance entry and attendance exit. However, this test was limited to mobile phone users with the Android operating system, and not all Android-based mobile phones can run this system properly. The analysis was conducted using a quantitative method, namely by looking at the number of photo executions based on face position, lighting, and execution time.*

*From the 18 people who were tested, the test results showed that based on face position, an upright and left position produced better attendance, while based on light intensity, the most effective results were at an intensity of between 50-99 lux, and the fastest time for the attendance process was less than one minute.*

*The research results show that the facial recognition system can run effectively in various facial positions. In addition, adequate lighting supports the success of facial recognition, and the upload process can be done in less than a minute.*

**Keyword:** *Face Recognition, FaceNet, Mobile Applications, Attendance System.*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Kedisiplinan di dunia kerja merupakan cerminan untuk menaati peraturan dan kebijakan perusahaan, serta berperan sebagai proses pelatihan untuk mengembangkan pengendalian diri. Dengan disiplin kerja yang baik, karyawan diharapkan dapat bekerja lebih efektif, meningkatkan prestasi kerja, dan berkontribusi secara optimal bagi perusahaan (Anggraini dkk., 2023). Selain itu, kedisiplinan juga merupakan sebuah tuntutan agar dapat menjadi pribadi yang lebih baik, lebih dewasa, dan lebih beretika. Disiplin kerja yang baik akan cenderung mendorong kinerja yang baik pula, karena disiplin merupakan faktor yang menentukan kinerja, semakin baik kedisiplinan yang melekat pada suatu aparatur, maka kinerja yang diharapkan pun akan semakin baik (Zysman & Costinot, 2022), oleh karena itu, kedisiplinan harus diterapkan dimana saja, salah satunya adalah ketika di dunia kerja, dimana tidak hanya bekerja untuk mencari nafkah, menjalankan perintah pimpinan, maupun berinovasi di lingkungan kerja, tetapi juga bekerja untuk meningkatkan kedisiplinan diri, seperti harus turun tepat waktu, dilarang meninggalkan kantor tanpa izin, bersikap sopan santun, dan lain-lain. Disiplin kerja merupakan salah satu aspek yang berperan dalam menentukan kinerja karyawan. Sebagai faktor penting yang memengaruhi kinerja karyawan, disiplin kerja harus ditegakkan oleh semua pihak. Para pimpinan tidak hanya bertanggung

jawab menegakkan disiplin bagi bawahannya, tetapi juga harus menjadi teladan dalam menerapkan disiplin bagi diri mereka sendiri.

Untuk penegakkan kedisiplinan, para pegawai dituntut untuk melakukan hal-hal yang dapat meningkatkan kedisiplinan, salah satunya adalah harus hadir ke kantor tepat waktu, dan wajib menjalankan presensi sebelum dan sesudah jam kerja. Sistem presensi yang ada saat ini, dan yang sedang berjalan di Dinas Komunikasi dan Informatika Kota Samarinda adalah sistem presensi sidik jari, namun kendala yang dihadapi adalah masih banyaknya manipulasi data presensi, dimana pegawai dapat mengetahui nomor urut pegawai sehingga dapat melakukan presensi atas nama pegawai yang lain, selain itu alat presensi yang tersedia hanya satu dan pegawai harus mendatangi alat tersebut, sehingga mengakibatkan antrian yang panjang terutama menjelang batas akhir presensi yaitu pukul 08.00 WITA, dan juga sering tidak terbaca karena terkadang jari pegawai atau alat presensi tersebut kotor, dan apabila terjadi pemadaman listrik maka sistem presensi tidak dapat berjalan dan akan mempengaruhi kinerja pegawai.

Teknologi pengenalan wajah sendiri telah berkembang sejak tahun 1950-an, dengan penelitian terapan pertama kali dilakukan pada tahun 1964. Pada masa awal, pendekatan pengenalan wajah lebih banyak mengandalkan geometri wajah, meskipun belum dapat diterapkan secara praktis. Seiring berjalannya waktu, algoritma pengenalan wajah mengalami kemajuan melalui berbagai teknik, termasuk penggunaan fitur *manual* dan pembelajaran mendalam (*deep learning*). Salah satu metode yang terkenal adalah *Local Binary Pattern (LBP)*, yang diperkenalkan oleh Ojala dan rekan-rekannya. *LBP* diadaptasi untuk pengenalan

wajah pada tahun 2004 oleh Ahonen dan timnya, membantu ekstraksi fitur gambar wajah dengan lebih efektif. Dengan kemajuan teknologi, pendekatan *deep learning* mulai diadopsi secara luas. Model *deep learning* mampu menangani data dalam skala besar dan menyesuaikan metode pemrosesan untuk *dataset* kecil dalam konteks tertentu. Namun, pelatihan model ini menghadapi tantangan seperti waktu pelatihan yang lama dan kompleksitas dalam mencapai solusi optimal global. Penelitian terbaru berfokus pada pengenalan wajah dalam kondisi nyata, termasuk analisis faktor-faktor yang mempengaruhi pengenalan wajah, penggunaan representasi fitur baru, serta sumber data baru. Pendekatan ini penting untuk meningkatkan akurasi dan keandalan teknologi pengenalan wajah di berbagai situasi dunia nyata. Meskipun teknologi pengenalan wajah telah menunjukkan kemajuan signifikan, masih ada tantangan besar dalam aspek teknis, hukum, dan etika. Kekhawatiran khusus muncul terkait algoritma yang dapat menghasilkan keputusan yang tidak adil atau diskriminatif, serta perlunya transparansi dalam proses otomatisasi pengolahan informasi identitas. (Li dkk., 2020)

Secara umum, pengenalan wajah dapat dikategorikan ke dalam beberapa klasifikasi algoritma, antara lain algoritma klasik yang salah satunya adalah klasifikasi *K-Nearest Neighbor (KNN)*, ekstraksi fitur, *Convolutional Neural Network (CNN)*, dan pendekatan berbasis pembelajaran mesin (*machine learning*). (Oloyede dkk., 2020)

Dalam perkembangan teknologi informasi dan komunikasi yang sangat pesat, diperlukan suatu sistem presensi yang lebih canggih, efektif dan efisien. Salah satu solusinya adalah sistem presensi berbasis pengenalan wajah. Teknologi

ini dapat memberikan keamanan data yang lebih baik, mengurangi peluang terjadinya manipulasi, dan memudahkan identifikasi pribadi secara otomatis. Pengenalan wajah dilakukan menggunakan algoritma ini untuk menghitung dan menganalisis posisi, bentuk, dan sudut wajah yang diperoleh, kemudian membandingkannya dengan data wajah yang ada di dalam basis data untuk mengidentifikasi keaslian identitas pengguna. Secara garis besar, sistem ini mencakup tiga langkah, yaitu: deteksi wajah, ekstraksi fitur, dan pencocokan fitur. Pengenalan wajah merupakan teknologi yang dapat mengenali wajah seseorang dari rekaman *video* melalui basis data wajah. Teknologi ini biasanya digunakan untuk mengidentifikasi pengguna melalui layanan verifikasi *id*. Pemanfaatan teknologi pengenalan wajah dalam administrasi publik, khususnya di instansi pemerintah sangat relevan untuk memenuhi kebutuhan modernisasi infrastruktur teknologi informasi. Penerapan sistem pengenalan wajah dengan *FaceNet* menawarkan beberapa keunggulan, antara lain akurasi yang tinggi, komputasi yang efisien, serta ketahanan terhadap variasi pencahayaan dan ekspresi wajah. *FaceNet* yang dikenal dengan kesederhanaan komputasi dan kinerjanya dalam kondisi waktu nyata (*real time*) telah terbukti efektif dalam berbagai aplikasi biometrik. Hal ini menjadikannya pilihan ideal untuk di aplikasikan pada sistem presensi di instansi pemerintah, di mana skala besar dan biaya komputasi rendah menjadi pertimbangan utama. Dalam konteks ini, penerapan *FaceNet* (Schroff dkk., 2015) pada sistem presensi berbasis pengenalan wajah di Dinas Komunikasi dan Informatika Kota Samarinda diharapkan mampu meningkatkan efisiensi pengelolaan presensi, pengendalian akses, dan keamanan data pegawai.

Melalui penerapan teknologi ini, penelitian ini bertujuan untuk memberikan kontribusi yang signifikan terhadap pengembangan sistem informasi di tingkat daerah, memperluas pemahaman tentang teknologi pengenalan wajah dalam administrasi publik, dan mendukung upaya modernisasi infrastruktur teknologi di lingkungan pemerintahan.

Secara tata kelola, diharapkan penerapan presensi berbasis pengenalan wajah menggunakan algoritma *FaceNet* dapat menjadi langkah strategis menuju sistem administrasi yang lebih transparan, akuntabel, dan adaptif terhadap kemajuan teknologi.

Selain itu, sistem ini juga diharapkan dapat menjadi inovasi yang mendorong modernisasi infrastruktur digital, dan dapat membuka peluang pengembangan lebih lanjut, baik di Dinas Komunikasi dan Informatika Kota Samarinda maupun instansi lain, serta dapat meningkatkan kedisiplinan pegawai di lingkungan kerja.

Diharapkan pengenalan wajah menggunakan *FaceNet* dapat menjadi solusi inovatif yang mendukung kedisiplinan pegawai secara lebih efektif, efisien, dan aman.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang tersebut, maka rumusan masalah dirumuskan sebagai berikut.

1. Bagaimana cara kerja sistem presensi berbasis pengenalan wajah menggunakan algoritma *FaceNet* bekerja?
2. Apa saja faktor yang memengaruhi keberhasilan proses pengenalan wajah

menggunakan *FaceNet* dengan memerhatikan berbagai aspek?

### 1.3 Batasan Masalah

Berdasarkan rumusan masalah tersebut, maka batasan masalah dapat dikemukakan sebagai berikut:

1. Pengenalan wajah hanya dilakukan menggunakan ponsel yang menggunakan sistem operasi *Android*.
2. Tidak semua pegawai memiliki ponsel dengan sistem operasi *Android*, dan tidak semua ponsel *Android* dapat menjalankan presensi dengan semestinya.
3. Penelitian ini dibatasi menggunakan 10 sampel wajah per pegawai dengan pose yang berbeda.

### 1.4 Tujuan Penelitian

Tujuan penelitian ini adalah:

1. Mengatasi manipulasi data presensi pegawai di Dinas Komunikasi dan Informatika Kota Samarinda.
2. Memberikan kemudahan bagi pegawai Dinas Komunikasi dan Informatika Kota Samarinda dalam melakukan presensi, agar tetap menjaga kedisiplinan dan tidak merugikan pegawai.

### 1.5 Manfaat Penelitian

Manfaat penelitian ini adalah:

1. Dapat meningkatkan kualitas di bidang teknologi informasi pada lingkungan pemerintahan.

2. Dapat meningkatkan kedisiplinan pegawai.
3. Dapat memberikan solusi bagi pemerintah kota dalam mendisiplinkan pegawainya, namun tidak merugikan pegawai yang diakibatkan oleh sistem yang tidak efektif.
4. Mempermudah pegawai Dinas Komunikasi dan Informatika Kota Samarinda untuk melakukan presensi.

#### 1.6 Hipotesis

Cara kerja sistem presensi menggunakan algoritma *FaceNet* yang diharapkan antara lain para pegawai harus merekam wajahnya terlebih dahulu, kemudian sistem tersebut melakukan pra-proses perekaman wajah sebelum digunakan untuk melakukan pengenalan wajah pada saat presensi.

Proses pengenalan wajah yang diajukan diharapkan mendapat hasil yang optimal dalam pengenalan wajah, antara lain posisi wajah yang dapat berjalan dengan baik ketika wajah menghadap ke depan dan ke samping, dan juga dapat optimal di berbagai pencahayaan, dari terang hingga gelap.

## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 Tinjauan Pustaka

Beberapa penelitian terdahulu yang relevan dengan penelitian yang diangkat, dikumpulkan sebagai acuan dalam penelitian ini.

Menurut penelitian dari Li dkk. (2020), dan Oloyede dkk. (2020), beberapa algoritma pengenalan wajah yang dijelaskan antara lain:

1. *AdaBoost*

*AdaBoost* merupakan algoritma *boosting* orisinal yang diusulkan oleh Schapire yang digunakan untuk deteksi wajah, algoritma tersebut meningkatkan akurasi dengan menggabungkan berbagai klasifikasi menjadi satu klasifikasi akhir yang lebih kuat. *Booster* tersebut efisien dalam memperbaiki hasil dari klasifikasi wajah. Namun tantangannya terdapat dalam menyesuaikan *dataset* pelatihan dan menggabungkan pengklasifikasi lemah untuk membentuk pengklasifikasi yang kuat, yang mungkin memerlukan waktu dan sumber daya yang cukup besar.

2. *Local Binary Pattern (LBP)*

*LBP* merupakan algoritma yang efektif dalam ekstraksi fitur wajah, terutama dalam lingkup tekstur, dan dapat memberikan hasil yang baik dalam kondisi pencahayaan yang berbeda. *LBP* relatif sederhana untuk diimplementasikan dan sangat efisien, namun *LBP* dapat menjadi sensitif terhadap *noise* dan variasi dalam kondisi pencahayaan yang ekstrim, yang dapat mengurangi akurasi pengenalan wajah.

### 3. *Deep learning*

*Deep learning* merupakan salah satu cabang dari pembelajaran mesin yang dapat mengetahui fitur-fitur yang dibutuhkan untuk klasifikasi secara otomatis dalam proses pelatihan tanpa langkah-langkah ekstraksi fitur. Hal ini memaksa pembelajaran jaringan untuk memperoleh fitur-fitur yang lebih efektif untuk membedakan wajah yang berbeda. Sejumlah model untuk pengenalan wajah antara lain *Convolutional Neural Networks (CNN)*, dapat menangani *dataset* besar dan mampu belajar dari data tanpa perlu desain fitur *manual*. Ini memberikan akurasi yang tinggi dan kemampuan adaptasi yang baik terhadap pola wajah yang kompleks. Namun pelatihan model *deep learning* memerlukan waktu yang lama dan sumber daya komputasi yang signifikan. Selain itu, tidak ada jaminan bahwa solusi yang ditemukan adalah optimal global, dan model ini bisa kompleks serta sulit untuk diinterpretasikan.

### 4. *FaceNet*

*FaceNet* merupakan algoritma pengenalan wajah yang tengah dipakai saat ini, dimana algoritma tersebut mencapai akurasi tinggi pada *dataset* yang dikenal, seperti *Labeled Faces in the Wild (LFW)*, selain itu *dataset* hanya memerlukan 128 *bytes* untuk merepresentasikan setiap wajah, ia juga dapat mengeluarkan *embedding* wajah yang dapat digunakan dengan teknik pengelompokkan dan verifikasi standar, efektif dalam mengatasi perubahan yang mungkin terjadi, seperti pencahayaan yang berbeda, pose wajah yang bervariasi, dan umur, serta memperkenalkan metode pembelajaran yang inovatif, seperti *triplet loss* yang membedakan antara pasangan yang cocok dan tidak cocok dengan sangat baik. Hal

ini membuat sistem *FaceNet* bekerja lebih efektif dalam implementasi, efisien dalam penyimpanan, dan tidak memerlukan penanganan khusus dalam *embedding*, sehingga algoritma tersebut dapat digunakan untuk penelitian ini. (Schroff dkk., 2015)

Sejumlah literatur yang relevan dengan penelitian ini juga dikumpulkan sebagai acuan dalam penelitian presensi pengenalan wajah ini.

Penelitian (Alburaiki dkk., 2021) mengembangkan sistem presensi *mobile* berbasis pengenalan wajah dan deteksi lokasi menggunakan *Azure Face API*. Sistem ini mengidentifikasi wajah siswa secara otomatis melalui kamera ponsel pintar (*smartphone*) dengan *face landmarks* yang mengenali 27 titik penting. Hasilnya menunjukkan akurasi tinggi dan proses presensi yang cepat, meski terbatas oleh kondisi pencahayaan, waktu pendaftaran wajah, dan persetujuan *manual admin*. Untuk peningkatan, disarankan pengumpulan *dataset* lebih bervariasi, fitur penerimaan otomatis, dan algoritma yang lebih adaptif terhadap pencahayaan.

Penelitian (Mensah dkk., 2024) mengevaluasi performa algoritma *FaceNet* dalam menghadapi tantangan oklusi dan variasi ekspresi wajah. Studi ini menggunakan *dataset* gambar wajah sintetis dengan tingkat oklusi 30% dan 40%, serta variasi ekspresi. Metode imputasi diterapkan untuk memulihkan area wajah yang teroklusi. Hasil penelitian menunjukkan bahwa *FaceNet* mencapai tingkat pengenalan yang lebih baik pada gambar yang telah di imputasi dibandingkan dengan gambar yang teroklusi. Metode *MissForest* terbukti paling efektif dalam meningkatkan akurasi pengenalan. Penelitian ini menyoroti pentingnya pemilihan

*classifier* dan teknik augmentasi data untuk meningkatkan kinerja sistem pengenalan wajah, khususnya dalam kondisi nyata yang seringkali dihadapkan pada oklusi dan variasi ekspresi. Namun, penelitian ini mengakui keterbatasannya dalam hal variasi pencahayaan dan posisi kepala, serta waktu pemrosesan *FaceNet* yang relatif lebih lama dibandingkan metode berbasis *PCA*.

Penelitian (Dang & Tran, 2023) berfokus pada pengembangan sistem identifikasi pengguna yang aman dan multi-level, memanfaatkan estimasi pose kepala, *MTCNN*, dan *FaceNet*. Meskipun *dataset* yang digunakan tidak dijelaskan secara detail, penelitian ini melaporkan bahwa sistem diuji dengan berbagai citra wajah dalam beragam pose. Sistem mencapai akurasi pengenalan 90-95% dengan kecepatan pemrosesan 21-25 *fps* pada platform *Jetson Nano*. Penggunaan estimasi pose kepala terbukti meningkatkan akurasi secara signifikan. Namun, penelitian ini kurang membahas pengaruh pencahayaan dan variasi ekspresi wajah terhadap akurasi sistem.

Penelitian (Liu & Qu, 2024) mengembangkan model pengenalan wajah yang dioptimalkan untuk lingkungan perpustakaan universitas. Model *FaceNet-MMAR* (*MobileNet-Mish-Attention-Receptive Field*) dihasilkan melalui optimasi yang menggabungkan *MobileNet* sebagai jaringan fitur utama, fungsi aktivasi *Mish*, modul *attention*, dan *receptive field*. Model ini di latih dan di validasi menggunakan *dataset* wajah mahasiswa dan staf universitas, serta *dataset* umum *CASIA-WebFace*. Hasil penelitian menunjukkan akurasi pengenalan 99,05%, tingkat kesalahan 0,51%, dan tingkat kepuasan pengguna yang tinggi (97,6% pengajar dan 96,8% mahasiswa). Meskipun demikian, penelitian ini mengakui tantangan yang

masih ada, seperti pengaruh faktor eksternal (pencahayaan, ekspresi, postur) dan potensi kesalahan pengenalan pada wajah non-frontal atau dalam kondisi pencahayaan yang buruk.

Penelitian (Sunaryono dkk., 2021) menerapkan algoritma *Linear Discriminant Analysis (LDA)* untuk sistem presensi berbasis pengenalan wajah. Meskipun detail *dataset* tidak dijelaskan, penelitian ini menggunakan gambar wajah mahasiswa yang diambil menggunakan *smartphone*. Sistem yang diusulkan mencapai akurasi 97,29% dengan waktu pemrosesan 0,000096 detik per gambar. Namun, penelitian ini mencatat keterbatasan seperti ketergantungan pada *smartphone* mahasiswa, potensi penurunan akurasi pada kondisi pencahayaan buruk, dan kebutuhan pengembangan lebih lanjut untuk mencegah kecurangan.

Penelitian (Yang & Han, 2020) mengusulkan sistem kehadiran berbasis pengenalan wajah menggunakan *Convolutional Neural Network (CNN)* untuk mengatasi tantangan seperti *blur*, perubahan pose, dan oklusi. *Dataset* terdiri dari *video* kehadiran di kelas yang melibatkan 200 mahasiswa. Sistem mencapai akurasi 82%, dengan tingkat kegagalan sekitar 15% akibat *blur* dan masalah teknis lainnya. Tingkat kesalahan pengenalan sekitar 20% disebabkan oleh perubahan fitur wajah, pencahayaan, dan ketidakakuratan dalam perbandingan kode identifikasi.

Penelitian (Budiman dkk., 2022) melakukan tinjauan sistematis yang membandingkan algoritma *Local Binary Pattern Histogram (LBPH)* dan *Convolutional Neural Network (CNN)* untuk sistem kehadiran berbasis wajah. Hasil menunjukkan *CNN* memiliki akurasi 99%, lebih tinggi dan lebih stabil daripada *LBPH* (92%). Namun, *CNN* membutuhkan lebih banyak data dan sumber daya

komputasi. Kedua algoritma dipengaruhi oleh faktor eksternal seperti pencahayaan, posisi wajah, dan aksesoris.

Penelitian (Alhanaee dkk., 2021) mengusulkan sistem presensi berbasis pengenalan wajah menggunakan *transfer learning* dan *Convolutional Neural Networks (CNN)* dengan tiga model pra-terlatih: *AlexNet*, *GoogleNet*, dan *SqueezeNet*. Dataset terdiri dari 200 gambar wajah dari 10 individu. *AlexNet* mencapai akurasi validasi tertinggi (100%), diikuti *SqueezeNet* (98,33%) dan *GoogleNet* (93,33%). Keunggulan sistem ini adalah akurasi tinggi dan waktu pelatihan yang relatif singkat, namun *AlexNet* membutuhkan waktu pelatihan yang lebih lama.

Penelitian (Geetha dkk., 2021) bertujuan meningkatkan akurasi sistem pengenalan wajah dalam pemantauan ujian daring menggunakan *Support Vector Machine (SVM)* dan metode *Eigenface*. Sistem mencapai akurasi sekitar 61%, yang dapat ditingkatkan dengan menambah jumlah data pelatihan. Kelemahan utama adalah sensitivitas terhadap pencahayaan, kualitas gambar, dan variasi pose.

Penelitian (Rollon dkk., 2023) mengembangkan sistem presensi berbasis pengenalan wajah menggunakan jaringan saraf konvolusional (*CNN*) multi-tugas. Sistem ini mencapai tingkat akurasi tinggi dalam mendeteksi kehadiran siswa. Namun, penelitian ini mengidentifikasi beberapa kekurangan, termasuk kebutuhan peningkatan antarmuka pengguna, penggunaan kamera *HD*, integrasi dengan *API*, dan penanganan individu dengan fitur wajah yang mirip.

## 2.2 Keaslian Penelitian

Tabel 2.1 Matriks literatur review dan posisi penelitian  
Evaluasi Pengenalan Wajah Menggunakan FaceNet pada Pegawai Dinas Komunikasi dan Informatika Kota Samarinda

No	Judul Penelitian	Nama Peneliti, Tahun, Index	Metode Penelitian	Hasil	Keunggulan dan Kelemahan	Perbandingan
1	<i>Mobile Based Attendance System: Face Recognition and Location Detection using Machine learning</i>	Mubarak Salem, Muharak Alburaiqi, Md Gapur Md Johar, Rabab Alayham, Abbas Helmi, Mohammed Hazim Alkawaz; 2021; <i>Conference Proceeding</i>	Mengembangkan dan merancang aplikasi <i>mobile</i> yang memungkinkan mahasiswa untuk melakukan presensi dengan menggunakan pengenalan wajah dan deteksi lokasi, sehingga meningkatkan akurasi, keamanan, serta	Sistem yang diusulkan menunjukkan kemampuan yang tinggi dalam mengenali wajah siswa dan mendeteksi lokasi dengan akurasi lebih dari 85%, serta mendapatkan umpan balik positif dari pengguna mengenai kemudahan dan	Ketergantungan pada kondisi pencahayaan yang baik, potensi waktu yang dibutuhkan untuk proses pendaftaran yang memakan waktu akibat sistem persetujuan <i>manual</i> , serta kebutuhan akan <i>dataset</i> yang lebih beragam untuk meningkatkan akurasi. Disarankan untuk mengembangkan fitur penerimaan otomatis dan algoritma yang lebih	Penelitian mereka menggunakan algoritma pengenalan wajah yang diintegrasikan dengan <i>Azure Face Service</i> , saat siswa pertama kali masuk ke sistem, mereka melalui proses pendaftaran wajah di mana gambar wajah mereka diambil beberapa kali. Sedangkan untuk penelitian ini menggunakan algoritma <i>FaceNet</i> , dan pada saat pegawai pertama kali masuk ke sistem,

			efisiensi proses presensi di lingkungan pendidikan.	efektivitas proses presensi.	adaptif terhadap kondisi pencahayaan yang bervariasi.	wajahnya direkam hingga 10 kali dalam posisi yang berbeda.
2	<i>FaceNet recognition algorithm subject to multiple constraints: Assessment of the performance</i>	Joseph A. Mensah, Justice K. Appati, Elijah K.A Boateng, Eric Ocran, Louis Asiedu; <i>Scientific African</i> ; 2024; Q1	Mengevaluasi kinerja algoritma <i>FaceNet</i> dalam pengenalan wajah di bawah berbagai kendala, seperti oklusi (penutupan sebagian wajah) dan ekspresi wajah yang bervariasi. Penelitian ini juga menilai efektivitas metode imputasi statistik untuk memulihkan informasi wajah	Algoritma <i>FaceNet</i> menunjukkan peningkatan tingkat pengenalan wajah ketika menggunakan wajah yang diimputasi dibandingkan dengan wajah yang terhalang, dengan metode imputasi <i>MissForest</i> memberikan hasil yang lebih akurat dibandingkan metode lainnya. Selain itu, skema augmentasi data	Penelitian ini merekomendasikan penggunaan metode imputasi <i>MissForest</i> dan skema augmentasi data untuk meningkatkan kinerja pengenalan wajah ketika jumlah gambar per subjek terbatas. Selain itu ditemukan bahwa waktu komputasi untuk algoritma <i>FaceNet</i> relatif tinggi dibandingkan dengan teknik berbasis <i>PCA</i> , sehingga disarankan untuk meningkatkan metode berbasis <i>PCA</i> yang memiliki	Penelitian mereka menggunakan <i>dataset Japanese Female Facial Expression</i> , dan <i>AU-coded Cohn-Kanade</i> yang kemudian dilakukan imputasi, sedangkan penelitian ini menggunakan data wajah yang diambil dari kamera ponsel, namun tidak menggunakan metode imputasi.

			yang hilang akibat oklusi.	tertentu dapat lebih meningkatkan kinerja model <i>FaceNet</i> , yang juga mengungguli beberapa metode berbasis <i>PCA</i> yang ditingkatkan, dengan tingkat pengenalan rata-rata yang lebih tinggi pada tingkat oklusi 30% dan 40%.	kompleksitas komputasi lebih rendah dan waktu komputasi yang lebih singkat.	
3	<i>A Secured, Multilevel Face Recognition based on Head Pose Estimation, MTCNN and FaceNet</i>	Thai-Viet Dang, Hoai-Linh Tran; <i>Journal of Robotics and Control (JRC)</i> ; 2023; Q3	Mengembangkan sistem pengenalan wajah yang aman dan efisien yang memanfaatkan <i>MTCNN (Multi-task Cascaded Convolutional Networks)</i> untuk deteksi wajah dan	Sistem pengenalan wajah yang diusulkan mencapai tingkat akurasi sebesar 90-95% dengan kecepatan pemrosesan 21-25 <i>frame per second (fps)</i> , dan integrasi estimasi pose kepala	Perbaikan lebih lanjut dapat dilakukan dalam hal efisiensi komputasi dan kemampuan untuk menangani berbagai kondisi pencahayaan dan oklusi. Pekerjaan di masa mendatang dapat melibatkan peningkatan ketahanan model terhadap	Penelitian ini tidak menggunakan <i>MTCNN</i> untuk deteksi wajah.

			<p>lokalisasi <i>landmark</i>, <i>FaceNet</i> untuk menghasilkan vektor fitur dari gambar wajah, dan estimasi pose kepala untuk meningkatkan akurasi pengenalan wajah dengan memastikan bahwa sistem dapat mengenali wajah dari berbagai sudut.</p>	<p>secara signifikan meningkatkan kemampuan sistem untuk mendeteksi dan mengenali wajah, dan mengurangi risiko input palsu.</p>	<p>berbagai jenis serangan <i>spoofing</i> dan mengoptimalkannya untuk penerapan pada perangkat berdaya rendah.</p>	
4	<p><i>Construction of a smart face recognition model for university libraries</i></p>	<p>Yan Liu, Yan Qu; <i>PLOS ONE</i>; 2024; Q1</p>	<p>Mengoptimalkan teknologi pengenalan wajah untuk digunakan</p>	<p>Algoritma <i>FaceNet-MMAR</i> yang diusulkan menunjukkan kinerja</p>	<p>Peningkatan lebih lanjut dalam teknologi pengenalan wajah untuk mengatasi tantangan</p>	<p>Penelitian mereka memadukan <i>FaceNet</i> dengan <i>MobileNet</i> sebagai fitur jaringan utama, modul</p>

	<p><i>based on FaceNet-MMAR algorithm</i></p>	<p>dalam sistem identifikasi kontrol akses perpustakaan pintar, meningkatkan akurasi dan efisiensi identifikasi kontrol akses perpustakaan, mengembangkan model pengenalan wajah cerdas yang mengatasi masalah keamanan dan meningkatkan efisiensi layanan di perpustakaan universitas.</p>	<p>yang unggul dalam tugas pengenalan wajah dibandingkan dengan metode tradisional, dengan akurasi pengenalan mencapai 99,05% dengan tingkat kesalahan identifikasi yang rendah sebesar 0,51%. Penerapan model ini dapat secara signifikan meningkatkan manajemen keamanan dan pengalaman pengguna di perpustakaan universitas.</p>	<p>seperti perubahan pencahayaan, ekspresi, dan postur, yang dapat memengaruhi akurasi pengenalan untuk meningkatkan ketahanan sistem pengenalan wajah dalam aplikasi dunia nyata.</p>	<p><i>Attention</i>, dan <i>Receptive field block (RFB)</i>, serta menggunakan fungsi aktivasi <i>Mish</i> untuk optimasi <i>FaceNet</i>.</p>
--	---	---	---	--	---

5	<i>An Android Based Course Attendance System Using Face Recognition</i>	Dwi Sunaryono, Joko Siswanto, Radityo Anggoro; <i>Journal of King Saud University – Computer and Information Sciences</i> ; 2021: Q1	Mengembangkan sistem kehadiran mahasiswa berbasis <i>Android</i> yang menerapkan sistem pengenalan wajah. Sistem ini dirancang untuk meningkatkan efisiensi dalam mencatat kehadiran mahasiswa, mengurangi kemungkinan kecurangan dalam proses kehadiran, meningkatkan akurasi dan kecepatan dalam pengenalan wajah.	Sistem kehadiran yang diusulkan mencapai akurasi pengenalan wajah sebesar 97.29% dengan menggunakan analisis diskriminan linier ( <i>LDA</i> ) dengan hanya memerlukan waktu 0.000096 detik untuk proses pengenalan wajah di <i>server</i> , dan telah diimplementasikan untuk memproses kehadiran di 21 mata kuliah selama tiga semester tanpa masalah signifikan.	Disarankan untuk mengeksplorasi penggunaan perangkat jarak antara <i>smartphone</i> mahasiswa dan <i>Raspberry Pi</i> yang terletak di ruang kelas, guna meminimalkan kemungkinan kecurangan dalam proses kehadiran. Kelemahan yang diidentifikasi yaitu potensi masalah dengan intensitas cahaya rendah yang dapat mempengaruhi kualitas gambar wajah yang diambil.	Secara prosedur, disebutkan bahwa presensi dilakukan dengan mengambil gambar wajah dan kode <i>QR</i> di depan kelas, kemudian diunggah ke <i>server</i> . Sedangkan pada penelitian ini, presensi dilakukan dengan mengambil gambar wajah, kemudian diunggah ke <i>server</i> dan menunggu hingga proses pengunggahan berhasil.
---	---	--	--	---	--	--

6	<i>Face Recognition Attendance System Based on Real-Time Video Processing</i>	Hao Yang, Xiaofeng Han; <i>IEEE Access</i> ; 2020; Q1	Merancang sistem presensi berbasis pengenalan wajah yang menggunakan pemrosesan <i>video real time</i> . Fokus utama dari penelitian ini meliputi akurasi sistem pengenalan wajah dalam proses presensi, stabilitas sistem presensi berbasis pengenalan wajah, tingkat ketidakhadiran yang dapat dikurangi dengan sistem, dan pengaturan	Sistem presensi berbasis pengenalan wajah menunjukkan akurasi sekitar 82% dalam pengenalan wajah, dibandingkan dengan metode presensi tradisional, sistem ini dapat mengurangi tingkat ketidakhadiran hingga 60%, dan dapat meningkatkan efisiensi dan mengurangi masalah presensi yang disebabkan oleh metode <i>manual</i> .	Ketidakakuratan yang disebabkan oleh kabur gambar, perubahan postur, dan oklusi (penutupan wajah), dan keterbatasan dalam pengenalan wajah yang dapat dipengaruhi oleh faktor-faktor seperti pencahayaan dan ekspresi wajah.	Penelitian mereka merancang aplikasi pengenalan wajah dengan pemrosesan <i>video</i> , sedangkan penelitian ini, proses presensi dilakukan dengan swafoto kemudian mengunggahnya ke <i>server</i> .
---	---	---	--	--	--	---

			antarmuka sistem presensi.			
7	<i>Student attendance with face recognition (LBPH or CNN): Systematic literature review</i>	Andre Budiman, Fabian, Ricky Aryatama Yaputera, Said Achmad, Aditya Kurmiawan; <i>Procedia Computer Science</i> ; 2023; <i>Conference Proceeding</i>	Menganalisis dan membandingkan algoritma pengenalan wajah yang sesuai untuk diterapkan dalam sistem kehadiran mahasiswa di universitas, khususnya algoritma <i>Convolutional Neural Network (CNN)</i> dan <i>Local Binary Pattern Histogram (LBPH)</i> .	Algoritma <i>CNN</i> memiliki akurasi yang lebih baik dibandingkan dengan <i>LBPH</i> dalam konteks pengenalan wajah untuk kehadiran. <i>CNN</i> juga menunjukkan stabilitas akurasi yang lebih baik ketika terpengaruh oleh faktor eksternal.	Penelitian ini merekomendasikan penggunaan algoritma <i>CNN</i> untuk sistem kehadiran mahasiswa karena keakuratannya yang lebih tinggi. Penelitian ini juga mencatat bahwa banyak algoritma pengenalan wajah yang ada membuat sulit untuk menentukan algoritma terbaik yang harus diterapkan, dan ada kebutuhan untuk penelitian lebih lanjut untuk mengatasi masalah ini.	Penelitian mereka membahas perbandingan algoritma <i>LBPH</i> dan <i>CNN</i> .
8	<i>Face Recognition Smart Attendance</i>	Khawla Alhanace,	Mengembangkan sistem kehadiran	Pendekatan yang diusulkan	Penelitian ini menyarankan untuk	Penelitian ini menggunakan <i>dataset</i> yang

	<p><i>System using Deep Transfer Learning</i></p>	<p>Mitha Alhammadi, Nahla Almenhali, Maad Shatnawi; <i>Procedia Computer Science</i>; 2021; <i>Conference Proceeding</i></p>	<p>berbasis pengenalan wajah menggunakan jaringan saraf konvolusional (CNN) dan teknik <i>transfer learning</i>. Sistem ini dirancang untuk meningkatkan efisiensi dan keamanan dalam manajemen kehadiran dengan memanfaatkan teknologi pengenalan wajah sebagai metode identifikasi biometrik.</p>	<p>menggunakan tiga model CNN yang telah dilatih sebelumnya (<i>SqueezeNet</i>, <i>GoogleNet</i>, dan <i>AlexNet</i>) dapat mencapai akurasi prediksi yang sangat tinggi. Model <i>AlexNet</i> mencapai akurasi 100%, sedangkan <i>SqueezeNet</i> dan <i>GoogleNet</i> masing-masing mencapai akurasi 98.33% dan 93.33%. Hasil menunjukkan bahwa sistem pengenalan wajah dapat diandalkan untuk</p>	<p>melakukan pengujian lebih lanjut dengan <i>dataset</i> yang lebih besar dan beragam untuk meningkatkan generalisasi model. Kelemahan yang diidentifikasi antara lain waktu pelatihan yang lebih lama untuk model <i>AlexNet</i> dibandingkan dengan model lainnya, yang mungkin menjadi kendala dalam aplikasi waktu nyata.</p>	<p>hanya dari pegawai Dinas Komunikasi dan Informatika Kota Samarinda untuk pengujian, dan tidak menggunakan model yang disebutkan, sehingga dapat mengidentifikasi kinerja algoritma <i>FaceNet</i> secara murni.</p>
--	---	--	---	---	--	--

				aplikasi manajemen kehadiran, dengan waktu pelatihan yang wajar dan akurasi yang tinggi.		
9	<i>Design of face detection and recognition system to monitor students during online examinations using Machine learning algorithms</i>	M. Geetha, R.S. Latha, S.K. Nivetha, S.Hariprasath, S.Gowtham, C.S.Deepak; 2021 <i>International Conference on Computer Communication and Informatics; 2021; Conference Proceeding</i>	Penelitian ini bertujuan untuk meningkatkan akurasi sistem pengenalan wajah yang ada dengan menggunakan algoritma <i>Support Vector Machine (SVM)</i> dan <i>Eigenface</i> . Fokus utama adalah untuk merancang model deteksi dan pengenalan wajah yang dapat memantau	Sistem yang diusulkan menunjukkan bahwa penggunaan algoritma <i>SVM</i> untuk deteksi wajah dapat meningkatkan kecepatan dan akurasi dalam memantau siswa selama ujian <i>online</i> . Penelitian ini juga mencatat bahwa meskipun ada kemajuan dalam sistem deteksi wajah, tantangan seperti	Penelitian ini mengakui bahwa meskipun sistem yang diusulkan efektif, masih ada ruang untuk perbaikan. Disarankan untuk menerapkan algoritma lain seperti <i>LBPH</i> , <i>Fisher faces</i> , <i>SIFT</i> , dan <i>SURF</i> untuk membangun model pengenalan yang lebih efisien. Penelitian lebih lanjut diperlukan untuk mengatasi masalah pencahayaan dan kualitas gambar yang dapat	Berdasarkan saran dari penelitian mereka, penelitian ini mempertimbangkan kondisi cahaya dan kualitas gambar, dimana gambar wajahnya merupakan resolusi penuh.

			<p>aktivitas siswa selama ujian <i>online</i>, terutama dalam konteks pembelajaran jarak jauh yang meningkat akibat pandemi.</p>	<p>variasi pencahayaan dan <i>noise</i> dalam gambar wajah masih ada. Penelitian ini merekomendasikan penggunaan algoritma lain seperti <i>Convolutional Neural Networks (CNN)</i> untuk meningkatkan akurasi lebih lanjut.</p>	<p>mempengaruhi akurasi deteksi wajah.</p>	
10	<p><i>Attendance System Based on Facial Recognition Using Multi-Task Convolutional Neural Network</i></p>	<p>Lance Andrew Rollon, Caye Angelica Lusing, John Bryan Aquino, Criselle Centeno, Ariel Antwan, Rolando Sison; <i>World Journal</i></p>	<p>Menciptakan sistem kehadiran yang memudahkan dan mempercepat pencatatan kehadiran menggunakan teknologi pengenalan wajah.</p>	<p>Penelitian ini berhasil mengembangkan aplikasi web yang signifikan bagi fakultas dan mahasiswa, dengan kualitas produk yang baik dalam hal fungsionalitas,</p>	<p>Penelitian ini mungkin menghadapi tantangan dalam mendeteksi individu dengan fitur wajah yang mirip, seperti kembar identik. Keterbatasan dalam pengujian yang dilakukan, yang mungkin tidak</p>	<p>Penelitian mereka juga memasukkan penilaian kualitas ISO 25010 dari berbagai aspek.</p>

		<i>of Advanced Research and Reviews; 2023; Non-index</i>	Mengatasi masalah yang ada pada metode pencatatan kehadiran tradisional yang sering kali tidak efisien dan rentan terhadap kesalahan manusia.	keandalan, efisiensi kinerja, dan kegunaan. Sistem ini terbukti bermanfaat untuk membantu fakultas dan mahasiswa dalam memeriksa kehadiran secara akurat dan otomatis.	mencakup semua skenario penggunaan.	
--	--	--	---	--	-------------------------------------	--

## 2.3 Landasan Teori

Bagian ini memuat rangkuman teori-teori yang diambil dari buku atau literatur yang mendukung penelitian, serta memuat penjelasan tentang konsep dan prinsip dasar yang diperlukan untuk pemecahan permasalahan. Landasan teori dapat berbentuk uraian kualitatif, model matematis, atau tools yang langsung berkaitan dengan permasalahan yang diteliti. Sumber teori yang dirujuk pada bagian ini harus dicantumkan dalam kalimat atau pernyataan yang diacu dan dicantumkan dalam Daftar Pustaka.

Pada sub bab ini, uraian mengenai teori dasar yang berhubungan dengan tesis ini di sajikan sebagai berikut.

### 2.3.1 *Python*

*Python* adalah bahasa pemrograman tingkat tinggi dan serbaguna yang banyak digunakan, awalnya bahasa pemrograman ini dikembangkan oleh Guido van Rossum pada awal 1990-an di Belanda. Bahasa ini dikelola oleh komunitas pengembang inti yang secara aktif terlibat dalam pertumbuhan dan kemajuannya. Meskipun logo resmi *Python* menunjukkan dua ular yang saling terkait, bahasa ini tidak dinamakan menurut ular mana pun. Van Rossum menamakan bahasa ini yang terinspirasi dari acara komedi tahun 1970-an yang berjudul '*Monty Python's Flying Circus*'.

*Python* memiliki tiga versi utama; versi awal, *Python 1.0*, dirilis pada Januari 1994. Versi utama kedua, *Python 2.0*, dirilis pada tahun 2000, dan versi utama ketiga, *Python 3.0*, dirilis pada tahun 2008. *Python 3* tidak kompatibel dengan *Python 2*, ini berarti bahwa kode yang ditulis dalam *Python 2* mungkin tidak

berfungsi seperti yang diharapkan dalam *Python 3* tanpa melakukan beberapa modifikasi. (Srivastava, 2024)

*Python* adalah bahasa pemrograman yang luar biasa. Sifatnya yang open source, serbaguna, dan tidak bergantung pada *platform* telah menghasilkan komunitas pengembang yang sangat besar, bersama dengan ekosistem yang luar biasa yang mencakup puluhan ribu pustaka (*library*) yang tersedia secara gratis untuk pengembangan *web*, pembelajaran mesin (*machine learning*), ilmu data, dan banyak domain lainnya.

Dalam ekosistem *Python*, umumnya menggunakan alat *Python* khusus domain, seperti kerangka kerja *web* dan pustaka *ML*, untuk menyelesaikan berbagai tugas dalam suatu pekerjaan. Penggunaan alat-alat ini bukanlah hal yang sepele, karena memerlukan keterampilan *Python* yang penting, seperti memproses teks, menangani data terstruktur, membuat alur kontrol, dan menangani *file*. Programmer *Python* dapat menulis solusi yang berbeda untuk menangani tugas yang sama. Di antara solusi-solusi ini, satu solusi umumnya lebih baik daripada yang lain karena solusi tersebut mungkin lebih ringkas, lebih mudah dibaca, atau lebih efisien, yang secara kolektif disebut sebagai *Pythonic*, yaitu gaya pengodean idiomatis yang ingin dikuasai oleh semua Programmer *Python*. (Cui, 2023)

### 2.3.2 *FaceNet*

*FaceNet* adalah sistem yang secara langsung belajar memetakan gambar wajah ke dalam ruang *Euclidean* yang kompak, di mana jarak antar titik merepresentasikan kesamaan wajah. Dengan menggunakan pemodelan ini, *FaceNet* dapat melakukan berbagai tugas termasuk verifikasi wajah (menentukan

apakah dua gambar wajah berasal dari orang yang sama), pengenalan wajah (mengidentifikasi seseorang berdasarkan wajahnya), dan pengelompokan wajah (mencari dan mengelompokkan wajah yang serupa dalam sekumpulan gambar). *FaceNet* menggunakan jaringan syaraf konvolusional (*CNN*) dalam pelatihan yang dirancang untuk mengoptimalkan *embedding* wajah secara langsung tanpa memerlukan lapisan *bottleneck* dalam prosesnya. Sistem ini dilatih menggunakan *triplet loss* yang memanfaatkan tiga gambar wajah: satu gambar sebagai *anchor* (gambar utama sebagai acuan), satu gambar positif yang serupa, dan satu gambar negatif yang berbeda. Tujuan *triplet loss* adalah untuk meminimalisir jarak antara *anchor* dan gambar positif, serta memaksimalkan jarak antara *anchor* dan gambar negatif. Dengan cara ini, sistem dapat membedakan wajah dengan lebih akurat.

Dalam implementasinya, *FaceNet* mengeksplorasi dua arsitektur jaringan syaraf, antara lain model *Zeiler&Fergus* dan *Inception*. Kedua arsitektur tersebut telah terbukti berhasil dalam banyak aplikasi *computer vision*. Pada *dataset Labeled Faces in the Wild (LFW)*, *FaceNet* mencapai akurasi baru sebesar 99.63%, dan pada *YouTube Faces DB*, sistem ini mencapai akurasi 95.12%. Ini menunjukkan peningkatan signifikan dibandingkan dengan hasil terbaik yang dipublikasikan sebelumnya, mengurangi tingkat kesalahan sekitar 30%. Dengan hanya menggunakan 128 *bytes* per wajah, *FaceNet* menunjukkan efisiensi representasi yang jauh lebih baik, memungkinkan aplikasi dalam skala besar di bidang pengenalan wajah. *FaceNet* juga efektif dalam mengelompokkan foto-foto pribadi pengguna ke dalam kelompok berdasarkan identitas yang sama. Hal ini

menunjukkan ketahanan terhadap oklusi, pencahayaan yang berbeda, pose dan perubahan usia. (Schroff dkk., 2015)

*FaceNet* adalah sebuah arsitektur jaringan syaraf dalam (*deep neural network*) yang dikembangkan oleh peneliti dari *Google* untuk pengenalan wajah dan pengelompokan wajah. *FaceNet* dirancang untuk menghasilkan representasi vektor dari wajah, yang memungkinkan komparasi wajah yang efektif untuk keperluan identifikasi dan verifikasi. Dengan menggunakan representasi ini, sistem dapat membandingkan kesamaan antara wajah dalam satu ruang. Salah satu kunci dari *FaceNet* adalah penggunaan *triplet loss* dalam proses pelatihannya. Dalam pengaturan ini, model dilatih untuk memastikan bahwa jarak antara vektor wajah yang sama (misalnya, wajah A dan B) lebih kecil dibandingkan dengan jarak vektor wajah yang berbeda (misalnya, wajah A dan C). Dengan cara ini, model belajar untuk menghasilkan vektor representasi wajah yang lebih terpisah berdasarkan kesamaan atau perbedaan di antara wajah tersebut. *FaceNet* dapat beroperasi dengan sangat efektif di latar belakang *database* besar, membuatnya ideal untuk aplikasi praktis seperti sistem keamanan dan pengenalan wajah secara *real time*. Arsitektur ini mencapai hasil yang sangat baik di berbagai *benchmark* pengenalan wajah, mengungguli banyak metode lainnya.

*FaceNet* telah digunakan dalam berbagai aplikasi berharga, seperti verifikasi identitas, pencarian wajah di *database*, dan penyaringan gambar untuk mengidentifikasi orang dalam foto dan *video*. Sejak diperkenalkan, *FaceNet* telah menginspirasi banyak penelitian lebih lanjut dalam bidang pengenalan wajah dan

pengembangan teknologi lainnya yang memanfaatkan vektor representasi untuk pengklasifikasian dan *clustering* wajah yang lebih efisien. (Parkhi dkk., 2015)

*FaceNet* adalah model *deep learning* yang dikembangkan untuk pengenalan wajah dan kesamaan wajah. Dikembangkan oleh *Google*, arsitektur ini digunakan untuk mengidentifikasi individu dan membandingkan wajah berdasarkan tingkat kesamaannya. *FaceNet* terkenal karena penggunaan teknik pembelajaran fitur (*embedding*) dan menggunakan serangkaian jaringan saraf konvolusional (*CNN*) untuk mencapai akurasi tinggi dalam pengenalan wajah.

*FaceNet* merepresentasikan wajah menggunakan vektor berdimensi rendah (*embedding*), yang merangkum fitur wajah dan menyederhanakan perbandingan wajah yang berbeda. Ia juga menggunakan fungsi loss yang disebut "*triplet loss*" untuk mengukur kesamaan wajah.

*FaceNet* mengikuti langkah-langkah mendasar berikut:

1. Unggah Gambar: Gambar wajah dimasukkan ke dalam model. Namun ukuran gambar masih bisa diubah ke dimensi tertentu (misalnya 160x160 piksel).
2. Lapisan Konvolusional: Gambar input melewati serangkaian lapisan konvolusional yang mengekstraksi fitur wajah dan membuat representasi yang lebih dalam.
3. Membuat Embedding: Model merepresentasikan setiap wajah menggunakan vektor berdimensi rendah (*embedding*) yang menyimpan fitur wajah dalam bentuk yang ringkas.

4. Menggunakan Triplet loss: *FaceNet* belajar menggunakan triplet, setiap triplet terdiri dari "anchor" (referensi), "positive" (mirip), dan "negative" (berbeda). Model mencoba meminimalkan jarak antara *anchor* dan positif, dan memaksimalkan jarak antara *anchor* dan negatif.
5. Hasil Keluaran: Model yang dilatih digunakan untuk mengenali wajah dan mengukur kesamaannya. Setiap wajah direpresentasikan oleh *embedding* 128 dimensi, dan kesamaan diukur berdasarkan jarak antara vektor-vektor tersebut.

*FaceNet* dapat digunakan dalam berbagai aplikasi. Berikut adalah beberapa kasus penggunaan umum:

1. Pengenalan Wajah: Untuk memverifikasi atau mengidentifikasi individu. Contohnya sistem keamanan atau proses verifikasi identitas.
2. Pengukuran Kesamaan Wajah: Untuk mengevaluasi kesamaan antara dua wajah. Contohnya *platform* media sosial untuk sistem rekomendasi wajah.
3. Pengenalan Ekspresi Wajah: Untuk menganalisis dan mengenali ekspresi wajah. Contohnya permainan atau aplikasi interaktif untuk analisis emosi.
4. Keamanan dan Pengawasan: Untuk mendeteksi perilaku abnormal dalam kamera pengintai. Contohnya sistem pelacakan dan analisis ancaman.
5. Transformasi dan Manipulasi Wajah: Dalam aplikasi *Virtual Reality* atau *Augmented Reality* untuk transformasi wajah. Contoh: Menggunakan *filter* dan efek wajah.

Beberapa kasus penggunaan khusus *FaceNet* dijelaskan pada tabel 2.2.

Tabel 2.1 Beberapa kasus penggunaan khusus *FaceNet*

No	Kasus	Masukan	Keluaran
1	Pengenalan Wajah dalam Sistem Keamanan	Gambar wajah dari kamera keamanan	identifikasi individu
2	Sistem Rekomendasi Kesamaan Wajah Media Sosial	Gambar wajah yang diunggah oleh pengguna	Rekomendasi wajah yang mirip dengan pengguna
3	Aplikasi Pengenalan Emosi	Gambar yang berisi ekspresi wajah	Analisis keadaan emosional wajah
4	Pelacakan Wajah dalam Permainan	Gambar wajah pemain	Perubahan ekspresi karakter secara langsung dalam permainan
5	Sistem Pengawasan	Gambar wajah dari tempat umum	Identifikasi dan pelacakan potensi ancaman

*FaceNet* adalah alat yang ampuh dalam pengenalan wajah dan pengukuran kesamaan. Dengan menggunakan teknik pembelajaran mendalam, alat ini mengenali dan membandingkan wajah secara akurat. Aplikasinya sangat luas, memainkan peran penting di banyak industri, mulai dari keamanan dan media sosial hingga aplikasi permainan dan medis. Dengan pengembangan lebih lanjut, *FaceNet* akan terus menjadi komponen utama sistem pengenalan wajah di masa mendatang. (Topsakal, 2024)

### 2.3.3 *Android*

*Android* adalah sistem operasi yang umumnya dibuat untuk perangkat seluler seperti ponsel dan *tablet*. Sistem operasi ini telah ada selama 15 tahun dan saat ini tidak hanya berjalan di ponsel saja, namun juga perangkat lain seperti jam tangan, kaca mata pintar, peralatan rumah tangga, mobil, kamera, *TV* pintar, dan konsol *game*. *Android* adalah *platform* telepon pintar yang paling banyak digunakan, dan menurut *TechJury*, per April 2021, *Android* masih mendukung sekitar 75% dari semua telepon pintar dan tablet. Sistem operasi ini bersifat *open*

*source* dan karenanya banyak perangkat lain yang menggunakannya sebagai sistem operasi.

Semua alat dan kerangka kerja yang diperlukan untuk pengembangan *Android* disediakan dengan baik oleh *Google* dan komunitas pengembangnya yang lebih besar. Untuk pengembangan, *source code* dapat dimodifikasi untuk membangun aplikasi dan dapat dijadikan aplikasi publik atau menyimpannya sendiri.

Aplikasi yang berfungsi penuh harus mencakup setidaknya dua dari elemen berikut:

1. *User interface*

*User interface* merupakan elemen yang dilihat pengguna di layar yang memperlihatkan komponen atau menu yang jelas untuk berfungsi sebagai titik interaksi antara pengguna dan aplikasi. Di *Android*, ada dua cara untuk mengimplementasikan antarmuka pengguna, yaitu menggunakan *XML (Extensible Markup Language)*, atau *Jetpack Compose*.

2. *Interactivity*

*Interactivity* adalah elemen untuk menampilkan elemen *UI* di dalam perangkat itu sendiri saat aplikasi dijalankan, dengan menghubungkannya ke kelas aktivitas.

3. *Storage options*

*Storage options* merupakan elemen yang sebagian besar aplikasi berfungsi untuk membuat, membaca, memperbarui, atau menghapus data saat sedang digunakan, atau disebut dengan manajemen *CRUD*, di mana ada kebutuhan untuk

mengelola data antara pengguna, aplikasi, dan entitas penyimpanan. Sistem *Android* menyediakan banyak opsi untuk menyimpan data di perangkat, bisa dilakukan secara lokal di kartu *SD*, memori *internal* perangkat, atau daring menggunakan penyimpanan *cloud*. (Panjuta & Nwokike, 2024)

#### 2.3.4 Klasifikasi Biner

Dalam klasifikasi biner, metrik evaluasi digunakan untuk menilai kinerja model klasifikasi dengan membandingkan hasil prediksi dengan hasil aktual. Istilah-istilah yang umum digunakan untuk menggambarkan hasil klasifikasi dapat dijelaskan sebagai berikut:

1. *True Positive (TP)*: merupakan kasus-kasus di mana model dapat memprediksi kelas positif dengan tepat. Dengan kata lain, objek yang sebenarnya positif dan model mengidentifikasinya dengan tepat sebagai positif.
2. *False Positive (FP)*: merupakan kasus-kasus di mana model salah memprediksi kelas positif. Ini merujuk pada objek yang sebenarnya negatif, tetapi model salah mengidentifikasinya sebagai positif.
3. *True Negative (TN)*: merupakan kasus-kasus di mana model dapat memprediksi kelas negatif dengan benar. Ini merepresentasikan objek yang sebenarnya negatif, dan model mengidentifikasinya dengan tepat sebagai negatif.
4. *False Negative (FN)*: merupakan kasus-kasus di mana model salah memprediksi kelas negatif. Ini merujuk pada objek yang sebenarnya positif, tetapi model salah mengidentifikasinya sebagai negatif. (Verma dkk., 2025)

## **BAB 3**

### **METODE PENELITIAN**

#### **3.1 Jenis, Sifat, dan Pendekatan Penelitian**

Tujuan utama dari penelitian ini adalah untuk mengembangkan dan menguji sistem pengenalan wajah menggunakan *FaceNet* pada pegawai Dinas Komunikasi dan Informatika Kota Samarinda. Pendekatan ini memungkinkan peneliti untuk mengevaluasi kinerja *FaceNet* dalam konteks pengenalan wajah secara sistematis dan terstruktur.

Jenis penelitian yang digunakan adalah deskriptif terapan, karena berfokus pada pembangunan dan pengembangan sistem pengenalan wajah menggunakan *FaceNet* bagi pegawai Dinas Komunikasi dan Informatika Kota Samarinda, yang di dalamnya terdapat penjelasan secara rinci mengenai proses deteksi dan pengenalan wajah.

Pendekatan yang diterapkan dalam penelitian ini menggunakan pendekatan kuantitatif, karena data yang diperoleh berupa citra wajah, dan terdapat pengerjaan teknis seperti penulisan kode dan eksekusi ke *server*, dan bersifat eksperimental karena melibatkan pengujian sistem pengenalan wajah untuk keperluan presensi pegawai.

#### **3.2 Metode Pengumpulan Data**

Rincian mengenai metode pengumpulan data dalam penelitian ini diuraikan sebagai berikut.

### 3.2.1 Populasi dan Sampel

Penelitian ini dilakukan di Dinas Komunikasi dan Informatika Kota Samarinda, namun di fokuskan kepada bidang *e-government* yang berjumlah 18 pegawai.

### 3.2.2 Teknik Pengumpulan Data

Data yang dibutuhkan dalam penelitian ini meliputi:

#### 1. Data Citra Wajah

Pengambilan citra wajah pegawai dilakukan secara kolektif dengan menggunakan kamera ponsel dari salah satu ponsel dengan resolusi kamera depan sebesar 32.0 MP, yang dilakukan pada berbagai posisi wajah. Ponsel yang digunakan untuk mengambil data citra wajah yaitu *Samsung Galaxy A55*, dimana spesifikasinya dijelaskan pada tabel 3.1.

<sup>1</sup>Tabel 3.1 Spesifikasi telepon genggam untuk mengambil data latih wajah

Ikhtisar	
Prosesor	2.75GHz, 2GHz
Berat (gram)	213
Prosesor	
CPU Speed	2.75GHz, 2GHz
Tipe CPU	Octa-Core
Display Utama	
Ukuran	168.3mm (6.6" full rectangle) / 163.7mm (6.4" rounded corners)
Resolusi	1080 x 2340 (FHD+)
Teknologi	Super AMOLED
Kedalaman Warna	16M
Kamera	
Rear Camera - Resolution (Multiple)	50.0 MP + 12.0 MP + 5.0 MP
Rear Camera - F Number (Multiple)	F1.8 , F2.2 , F2.4
Main Camera - Auto Focus	Yes
Rear Camera - OIS	Yes
Rear Camera Zoom	Digital Zoom up to 10x

<sup>1</sup> (<https://www.samsung.com/id/smartphones/galaxy-a/galaxy-a55-5g-awesome-navy-256gb-sm-a556ezkcxid/>)

<i>Front Camera - Resolution</i>	32.0 MP
<i>Front Camera - F Number</i>	F2.2
<i>Front Camera - Auto Focus</i>	No
<i>Front Camera - OIS</i>	No
<i>Main Camera - Flash</i>	Yes
<i>Front Camera - Flash</i>	No
<i>Video Recording Resolution</i>	UHD 4K (3840 x 2160) @30fps
<i>Slow Motion</i>	240fps @HD
<b>Penyimpanan</b>	
<i>Memori</i>	8 GB
<i>Penyimpanan</i>	256 GB
<i>Penyimpanan Tersedia</i>	235.5 GB
<i>Dukungan Penyimpanan Eksternal</i>	MicroSD (Up to 1TB)
<b>Network/Bearer</b>	
<i>Number of SIM</i>	Dual-SIM
<i>SIM size</i>	Nano-SIM (4FF), Embedded-SIM
<i>SIM Slot Type</i>	SIM 1 + Hybrid (SIM or MicroSD) or Embedded SIM
<i>Infra</i>	2G GSM, 3G WCDMA, 4G LTE FDD, 4G LTE TDD, 5G Sub6 FDD, 5G Sub6 TDD
<i>2G GSM</i>	GSM850, GSM900, DCS1800, PCS1900
<i>3G UMTS</i>	B1(2100), B2(1900), B4(AWS), B5(850), B8(900)
<i>4G FDD LTE</i>	B1(2100), B2(1900), B3(1800), B4(AWS), B5(850), B7(2600), B8(900), B12(700), B13(700), B17(700), B20(800), B25(1900), B26(850), B28(700), B66(AWS-3)
<i>4G TDD LTE</i>	B38(2600), B40(2300), B41(2500)
<i>5G FDD Sub6</i>	N1(2100), N3(1800), N5(850), N7(2600), N8(900), N26(850), N28(700), N66(AWS-3)
<b>Konektivitas</b>	
<i>USB Interface</i>	USB Type-C
<i>USB Version</i>	USB 2.0
<i>Location Technology</i>	GPS, Glonass, BeiDou, Galileo, QZSS
<i>Earjack</i>	USB Type-C
<i>MHL</i>	No
<i>Wi-Fi</i>	802.11a/b/g/n/ac/ax 2.4GHz+5GHz, HE80, MIMO, 1024-QAM
<i>Wi-Fi Direct</i>	Yes
<i>Bluetooth Version</i>	Bluetooth v5.3
<i>NFC</i>	Yes
<b>Sistem Operasi</b>	
<i>Android</i>	
<b>Informasi Umum</b>	
<i>Warna</i>	Awesome Navy
<i>Form Factor</i>	Touch Bar

<b>Sensor</b>	
<i>Accelerometer, Fingerprint Sensor, Gyro Sensor, Geomagnetic Sensor, Hall Sensor, Light Sensor, Virtual Proximity Sensing</i>	
<b>Spesifikasi fisik</b>	
<i>Dimension (HxWxD, mm)</i>	<i>161.1 x 77.4 x 8.2</i>
<i>Weight (g)</i>	<i>213</i>
<b>Baterai</b>	
<i>Video Playback Time (Hours, Wireless)</i>	<i>Up to 28</i>
<i>Battery Capacity (mAh, Typical)</i>	<i>5000</i>
<b>Audio dan video</b>	
<i>Stereo Support</i>	<i>Yes</i>
<i>Video Playing Format</i>	<i>MP4, M4V, 3GP, 3G2, AVI, FLV, MKV, WEBM</i>
<i>Video Playing Resolution</i>	<i>UHD 4K (3840 x 2160) @30fps</i>
<i>Audio Playing Format</i>	<i>MP3, M4A, 3GA, AAC, OGG, OGA, WAV, AMR, AWB, FLAC, MID, MIDI, XMF, MXMF, IMY, RTTTL, RTX, OTA</i>
<b>Layanan dan Aplikasi</b>	
<i>Gear Support</i>	<i>Galaxy Buds2 Pro, Galaxy Buds Pro, Galaxy Buds Live, Galaxy Buds+, Galaxy Buds2, Galaxy Buds, Galaxy Buds FE, Galaxy Fit3, Galaxy Fit2, Galaxy Fit e, Galaxy Fit, Galaxy Watch6, Galaxy Watch5, Galaxy Watch4, Galaxy Watch3, Galaxy Watch, Galaxy Watch Active2, Galaxy Watch Active</i>
<i>Samsung DeX Support</i>	<i>No</i>
<i>SmartThings Support</i>	<i>Yes</i>
<i>Mobile TV</i>	<i>No</i>

Bentuk ponsel yang digunakan pada penelitian ini ditampilkan pada gambar

3.1.



<sup>2</sup>Gambar 3.1 Telepon genggam yang digunakan untuk memotret data latih wajah

Untuk data latih, jumlah pegawai yang diambil sampel data latihnya sebanyak 40 pegawai, setiap pegawai memiliki jumlah maksimal 10 wajah dengan posisi wajah yang berbeda, sehingga menghasilkan 400 data wajah, seperti dijelaskan pada tabel 3.2.

Tabel 3.2 Statistik Data Sampel Wajah

No	NIM	Jumlah
1	Jumlah Pegawai yang Sampelnya Diambil	40
2	Jumlah Foto Maksimal per Pegawai	10
3	Jumlah Gambar Sampel yang Diunggah	400
4	Rata-rata Jumlah Sampel per Pegawai	10

Dalam proses pengambilan wajah untuk dijadikan data latih, 10 wajah tersebut diambil dari sudut yang berbeda, mulai dari tegap, kanan, kiri, atas, dan bawah seperti yang dijabarkan di gambar 3.2.

<sup>2</sup> (<https://www.samsung.com/id/smartphones/galaxy-a/galaxy-a55-5g-awesome-iceblue-256gb-sm-a556elbcxid/buy/>)



Gambar 3.2 Beberapa sampel wajah yang terdiri dari (urutan dari kiri ke kanan) posisi tegap, kanan sudut  $30^{\circ}$ , kanan sudut  $60^{\circ}$ , kanan sudut  $90^{\circ}$ , kiri sudut  $30^{\circ}$ , kiri sudut  $60^{\circ}$ , kiri sudut  $90^{\circ}$ , atas, bawah, dan tegap.

## 2. Data Uji

Pengambilan citra wajah pegawai pada berbagai kondisi pencahayaan, dan pose wajah untuk dijadikan data uji pada sistem pengenalan wajah, seperti dicontohkan pada gambar 3.3.



Gambar 3.3 Contoh pose wajah yang dijadikan data uji, dengan tiga posisi wajah berbeda, dan berada pada cahaya terang

Misalnya melalui observasi, wawancara, eksperimen, atau penyebaran angket. Jika metode penyebaran angket digunakan, maka blangko angket harus

dilampirkan dalam proposal. Untuk setiap metode pengumpulan data, harus dijelaskan tentang jenis data yang dikumpulkan dengan metode-metode yang terkait. Bagian ini juga memuat penjelasan secara lengkap dan terinci tentang jenis data yang diperlukan untuk analisis dalam pembahasan.

### **3.3 Metode Analisis Data**

Analisis data yang akan diteliti adalah hasil perbandingan antara data wajah yang telah difoto dengan hasil deteksi menggunakan *FaceNet* dengan berbagai kondisi pencahayaan dan posisi wajah berbeda, serta durasi proses pengenalan wajah yang disimpan ke *server*.

### **3.4 Alur Penelitian**

Penelitian yang dijalankan dijelaskan secara detail sebagai berikut.

#### **3.4.1 Alat yang Digunakan**

Dalam pembuatan aplikasi presensi berbasis pengenalan wajah, dibutuhkan berbagai alat untuk pembuatannya, antara lain:

1. *Laptop* dengan spesifikasi tinggi agar dapat melakukan komputasi dengan baik dan cepat.
2. Telepon genggam pintar untuk melakukan percobaan aplikasi, dengan mengaktifkan mode pengembangan (*debugging*).

#### **3.4.2 Bahasa Pemrograman**

Bahasa pemrograman yang digunakan antara lain:

1. *Python*: untuk menjalankan proses pengenalan wajah, dengan berbagai pustaka pendukung.
2. *Dart*: untuk membuat aplikasi *mobile* melalui *framework Flutter*.

### 3.4.3 Server

Server yang digunakan menggunakan layanan *Amazon Web Service*, perangkat lunak *Ubuntu* versi 24.04 LTS, CPU sebesar 8 Core, memori sebesar 32 GB, dan *bandwidth* 33 T. Server tersebut digunakan untuk proses penyambungan ke halaman *endpoint*.

### 3.4.4 Pengukuran Cahaya

Untuk melakukan pengukuran cahaya pada saat proses pengenalan wajah, dibutuhkan aplikasi untuk mengukur intensitas cahaya, yaitu *Lux*, yang dapat dijalankan melalui telepon genggam. Patokan yang diambil untuk menentukan intensitas cahaya dalam pengambilan pose wajah yaitu berdasarkan posisi kamera depan ketika pengguna mengunggah wajahnya.

### 3.4.5 Pengumpulan Dataset

Pada tahap awal penelitian, dilakukan pengumpulan *dataset* yang dibagi menjadi dua jenis, yaitu *dataset* publik dan *dataset custom*. *Dataset* publik yang digunakan adalah *Labeled Faces in the Wild (LFW)* yang diperoleh melalui situs *Kaggle*, kemudian di seleksi dengan mempertimbangkan jumlah foto latih setiap orangnya, namun terbatas pada posisi wajah dan distraksi dengan objek non-wajah. Sementara itu, *dataset custom* diperoleh dari hasil perekaman wajah pegawai yang telah dilakukan sebelumnya, dimana masing-masing pegawai diambil fotonya sebanyak sepuluh kali dalam posisi wajah yang berbeda.

### 3.4.6 Evaluasi model FaceNet pada dataset publik

Pada tahap ini, model *FaceNet* dievaluasi menggunakan *dataset* publik. Proses evaluasi dilakukan dengan memanfaatkan sejumlah foto latih (*training*) dari *dataset LFW* yang diperoleh melalui laman *Kaggle*, kemudian dilakukan pengujian

terhadap foto uji (*testing*) dengan mengambil sejumlah foto dari laman *Google* dengan melihat bentuk wajah yang berbeda dari foto latih. Hasil evaluasi diukur berdasarkan jumlah eksekusi pengenalan wajah yang berhasil.

#### **3.4.7 Evaluasi model *FaceNet* pada *dataset custom***

Setelah pengujian menggunakan *dataset* umum, tahap selanjutnya adalah mengevaluasi kinerja model *FaceNet* menggunakan *dataset custom*. Sejumlah foto latih yang diperoleh dari perekaman wajah pegawai digunakan dalam proses pelatihan, kemudian dilakukan pengujian dengan melakukan foto terhadap pegawai yang fotonya sudah di latih menggunakan telepon genggam pintar. Sama seperti pada tahap sebelumnya, hasil evaluasi diukur berdasarkan jumlah eksekusi pengenalan wajah yang berhasil.

#### **3.4.8 Implementasi model *FaceNet* sebagai layanan**

Setelah pengujian menggunakan *dataset* selesai, tahap selanjutnya adalah model *FaceNet* diimplementasikan dalam bentuk layanan (*service*) berbasis *Application Programming Interface (API)* menggunakan *framework Flask*. Implementasi ini bertujuan agar proses pengenalan wajah dapat di akses secara terpusat melalui *server*, sehingga memungkinkan integrasi dengan aplikasi lain. Layanan ini menerima input berupa citra wajah, kemudian melakukan proses ekstraksi fitur menggunakan model *FaceNet*, dan mengembalikan hasil verifikasi atau Identifikasi wajah kepada pengguna.

#### **3.4.9 Implementasi *front end* menggunakan aplikasi *mobile***

Tahap ini berfokus pada pengembangan *front end* berupa aplikasi *mobile* yang berfungsi sebagai antarmuka pengguna. Aplikasi dikembangkan menggunakan *framework Flutter* dan di integrasikan dengan *service FaceNet* yang

telah dibangun. Melalui aplikasi ini, pengguna dapat melakukan perekaman wajah, mengunggah citra ke *server*, serta menerima hasil proses pengenalan wajah secara langsung. Dengan demikian, aplikasi *mobile* berperan sebagai penghubung antara pengguna dan sistem pengenalan wajah berbasis *FaceNet*.

#### **3.4.10 Evaluasi sistem presensi pada kasus nyata di Dinas Komunikasi dan Informatika Kota Samarinda**

Tahap ini dilakukan untuk mengevaluasi kinerja sistem presensi berbasis *FaceNet* pada kondisi nyata di lingkungan Dinas Komunikasi dan Informatika Kota Samarinda. Evaluasi difokuskan pada dua aspek utama, yaitu:

##### **1. Variasi Pencahayaan**

Pengujian dilakukan dengan mengambil data wajah pegawai dalam kondisi pencahayaan yang berbeda-beda, misalnya terang, redup, maupun cahaya tidak merata. Tujuannya adalah untuk menilai sejauh mana sistem mampu mempertahankan akurasi pengenalan wajah pada kondisi pencahayaan yang bervariasi.

##### **2. Variasi Pose atau Arah Wajah**

Pengujian juga dilakukan dengan memperhatikan variasi pose wajah, seperti wajah tegak lurus, miring ke kanan atau kiri, serta menunduk atau mendongak. Evaluasi ini bertujuan untuk mengetahui tingkat keandalan sistem dalam mengenali wajah dengan perubahan sudut pandang.

##### **3. Durasi**

Selain memperhatikan variasi pose wajah dan pencahayaan, pengujian wajah juga memperhatikan lamanya proses pengenalan wajah. Evaluasi ini

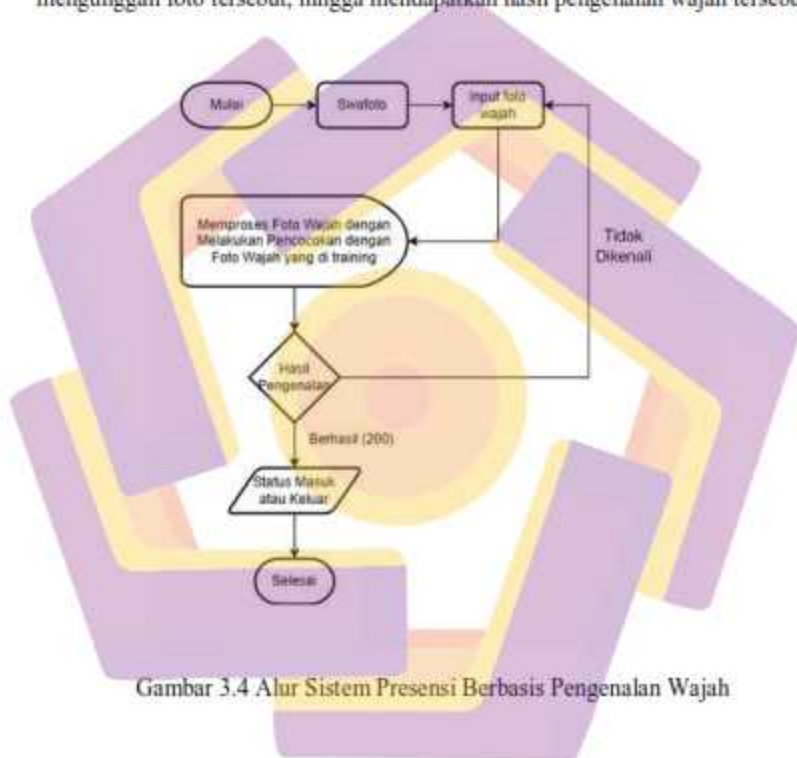
bertujuan untuk mengetahui tingkat realibilitas latensi *server* terhadap proses eksekusi pengenalan wajah.

Rangkaian alur penelitian disusun ke dalam berbagai tahap sebagai berikut:

1. Pengambilan gambar sampel: Para pegawai melakukan pemotretan wajah menggunakan satu kamera telepon genggam untuk dijadikan sampel gambar, pengambilan dilakukan dengan berbagai sudut wajah yang berbeda, mulai dari kanan, kiri, atas, dan bawah.
2. Pemrosesan Gambar Sampel: Setelah gambar sampel terkumpul, selanjutnya membuat akun masing-masing pegawai dan mengunggah gambar sampel tersebut, kemudian di *server* dilakukan penyimpanan gambar sampel sesuai dengan nama pegawai terkait, dan mengonversi gambar sampel tersebut menjadi sebuah fitur.
3. Pengujian: Setelah pembuatan akun, kemudian dilakukan pengujian dengan *login* ke akun masing-masing pegawai terlebih dahulu menggunakan ponsel masing-masing pegawai, setelah itu mereka melakukan swafoto dengan posisi wajah dan pencahayaan yang berbeda, kemudian mengunggah foto tersebut dan menunggu hasil pengenalan wajah tersebut. Untuk proses pemotretan wajah, posisi wajah yang diambil adalah posisi tegap, kanan dengan sudut  $90^{\circ}$ , dan kiri dengan sudut  $90^{\circ}$ , sedangkan untuk pencahayaan, dibagi menjadi tiga kategori, yaitu gelap (intensitas cahaya di bawah 50 *lumen*), remang-remang (antara 50-99 *lumen*), dan terang (minimal 100 *lumen*). Pengukuran cahaya dilakukan menggunakan salah

satu aplikasi pengukuran cahaya, yaitu 'Lux', dengan berpatok pada posisi kamera ketika melakukan swafoto.

Alur proses presensi berbasis pengenalan wajah dijelaskan dalam bentuk diagram alir seperti pada gambar 3.4., mulai dari proses swafoto, kemudian mengunggah foto tersebut, hingga mendapatkan hasil pengenalan wajah tersebut.



Gambar 3.4 Alur Sistem Presensi Berbasis Pengenalan Wajah

## BAB 4

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1 Pengantar

Berbagai penjelasan dalam penelitian ini dirangkum, dimulai dari proses pengambilan wajah, penulisan kode *script*, struktur basis data, hingga analisis yang didapatkan.

#### 4.2 Struktur File

Untuk membangun sebuah sistem presensi berbasis pengenalan wajah, dibutuhkan sejumlah komponen pendukung yang saling terintegrasi, yang terdiri dari *file* kode program yang berfungsi untuk melakukan eksekusi presensi, hingga *folder* untuk menyimpan foto wajah, baik data latih maupun data uji. Seluruh komponen tersebut dimasukkan ke dalam satu *folder* utama, yang dinamakan *Face Attendance Backend*, sebagaimana diuraikan pada tabel 4.1.

Tabel 4.1 Alur *file* project program

• <i>Face Attendance Backend</i>
• <i>uploads</i>
• <i>attendance</i>
• <i>info.txt</i>
• <i>regface</i>
• <i>info.txt</i>
• <i>utils</i>
• <i>helpers.py</i>
• <i>README.md</i>
• <i>app.py</i>

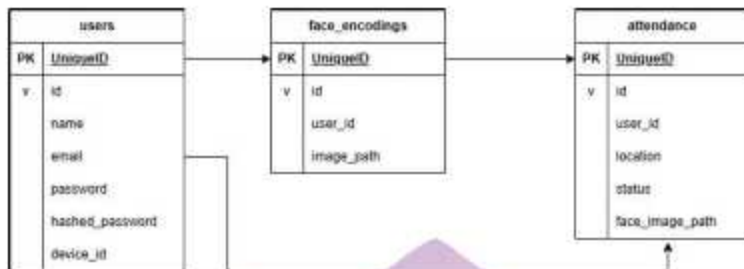
Semua *file* untuk menjalankan sistem presensi berbasis pengenalan wajah disimpan dalam *folder Face Attendance Backend*, yang terdiri dari dua *folder*

utama, yaitu *uploads* dan *utils*, serta dua berkas penting, yaitu *README.md* dan *app.py*. Folder *uploads* digunakan untuk menyimpan gambar wajah, yang terbagi dalam dua *subfolder*, yaitu *regface* untuk menyimpan foto registrasi atau pelatihan wajah, dan *attendance* untuk menyimpan foto yang diambil saat proses presensi; masing-masing *subfolder* dilengkapi dengan berkas *info.txt* yang menjelaskan fungsinya. Folder *utils* berisi kode pendukung sistem yang di dalamnya berupa berkas dengan nama *helpers.py* yang memuat fungsi-fungsi bantu. Sementara itu, *README.md* menyediakan informasi lengkap mengenai instalasi, konfigurasi, struktur basis data, dan cara menjalankan *server*, sedangkan *app.py* merupakan inti dari sistem, mencakup fungsi autentikasi, registrasi wajah, pelaksanaan presensi, hingga pencatatan data kehadiran pegawai.

#### 4.3 Struktur Basis Data

Untuk pemrosesan data akun dan citra wajah, diperlukan sebuah komponen utama yaitu basis data, yang berfungsi untuk menyimpan informasi-informasi yang digunakan dalam sistem presensi berbasis pengenalan wajah. Di dalamnya terdapat tabel untuk menampung data pengguna, data latih wajah, data uji wajah untuk presensi, dan data hasil presensi. Dengan adanya basis data ini, sistem dapat memproses data-data tersebut secara efisien dan terstruktur.

Untuk penelitian ini, struktur basis data yang menjadi acuan untuk sistem presensi berbasis pengenalan wajah dijelaskan pada gambar 4.1.



Gambar 4.1 Struktur Basis Data untuk sistem presensi berbasis pengenalan wajah

Basis data dari sistem presensi berbasis pengenalan wajah yang digunakan meliputi tabel *users* yang berfungsi untuk menampung data pengguna sekaligus pondasi untuk membuat relasi dengan tabel lain, tabel *face\_encodings* yang berfungsi untuk menampung data citra wajah baik data latihan maupun data uji beserta jalur untuk menyimpannya, dan tabel *attendance* yang berfungsi untuk menampung data hasil presensi.

#### 4.3.1 Tabel *users*

Tabel *users* merupakan tabel yang digunakan untuk menampung data pengguna yang mendapatkan hak akses sistem presensi berbasis pengenalan wajah, sebagaimana kolom-kolom yang ditampung diuraikan dalam tabel 4.2.

Tabel 4.2 Daftar kolom yang mengisi tabel *users*

No	Nama Kolom	Type Data	Jumlah Karakter	Keterangan
1	<i>id</i>	<i>serial</i>		Nomor data yang bersifat <i>primary key</i> & <i>auto increment</i>
2	<i>name</i>	<i>varchar</i>	255	Nama pengguna
3	<i>email</i>	<i>varchar</i>	255	Alamat <i>email</i> pengguna yang bersifat <i>unique</i>
4	<i>password</i>	<i>varchar</i>	255	Kata sandi asli
5	<i>hashed_password</i>	<i>varchar</i>	255	Kata sandi yang sudah diberi <i>hash</i>
6	<i>device_id</i>	<i>varchar</i>	255	<i>ID</i> perangkat pengguna

7	<i>created_at</i>	<i>timestamp</i>		Waktu pembuatan akun yang tercatat otomatis
8	<i>updated_at</i>	<i>timestamp</i>		Waktu pembaharuan akun yang tercatat otomatis

Kolom *id* merupakan kolom utama (*primary key*) yang secara umum berfungsi sebagai identitas unik untuk setiap data yang disimpan dalam setiap tabel, kolom tersebut bersifat *auto increment* yang berarti nomor tersebut bertambah secara otomatis setiap data baru dimasukkan, dan kolom ini juga berperan penting dalam relasi antar tabel karena pada umumnya dapat dijadikan sebagai referensi bagi tabel lain yang terhubung. Kolom *id* pada sistem ini juga diterapkan pada tabel *face\_encodings* dan *attendance*.

Kolom lainnya antara lain *name* yang merupakan tempat untuk menampung nama pegawai yang mendapatkan hak akses sistem. Kolom *email* yang berfungsi untuk menyimpan nama *email* setiap pegawai dan bersifat unik yang artinya tidak diperbolehkan adanya dua nama *email* yang sama di setiap penyimpanannya. Kolom *password* dan *hashed\_password* yang digunakan untuk menampung kata sandi pegawai yang masing-masing berisi kata sandi asli dan versi yang sudah diberi *hash* untuk meningkatkan keamanan data. Kolom *device\_id* yang merupakan identitas perangkat yang digunakan oleh pengguna, dimana dalam konteks ini adalah identitas telepon genggam pengguna. Kolom *created\_at* yang berfungsi untuk mencatat waktu pendaftaran secara waktu nyata, dan kolom *updated\_at* untuk mencatat waktu terakhir ketika pengguna mengubah informasi akun.

### 4.3.2 Tabel *face\_encodings*

Tabel *face\_encodings* merupakan tabel yang digunakan untuk menampung data citra wajah setiap pegawai yang terdaftar di sistem, sebagaimana kolom-kolom yang ditampung diuraikan dalam tabel 4.3

Tabel 4.3 Daftar kolom yang mengisi tabel *face\_encodings*

No	Nama Kolom	Tipe Data	Jumlah Karakter	Keterangan
1	<i>id</i>	<i>serial</i>		Nomor data yang bersifat <i>primary key</i> & <i>auto increment</i>
2	<i>user_id</i>	<i>integer</i>		<i>Foreign key</i> dari tabel <i>users</i>
3	<i>image_path</i>	<i>varchar</i>	255	Lokasi penyimpanan data latih citra wajah di <i>server</i>
4	<i>created_at</i>	<i>timestamp</i>		Waktu penyimpanan wajah yang tercatat otomatis

Selain kolom *id* yang sudah dijelaskan di subbab 4.3.1., terdapat kolom *user\_id* yang merupakan tempat penyimpanan nomor *id* dari tabel *users* yang dihubungkan melalui *foreign key*, dalam tabel ini, *user\_id* berfungsi untuk mengidentifikasi pengguna yang mengunggah foto wajah ke *server*. Kolom *image\_path* merupakan kolom untuk menyimpan lokasi penyimpanan data latih citra wajah ke *server*, dan ditulis dalam bentuk nama jalur lengkap dari *folder* utama hingga nama *file* gambar tersebut beserta ekstensinya (seperti *jpg*). Dan kolom *created\_at* untuk mencatat waktu penyimpanan wajah secara otomatis dan waktu nyata.

### 4.3.3 Tabel *attendance*

Tabel *attendance* merupakan tabel yang digunakan untuk menampung data presensi yang berhasil dilakukan, kolom-kolom yang ditampung diuraikan dalam tabel 4.4.

Tabel 4.4 Daftar kolom yang mengisi tabel *attendance*

No	Nama Kolom	Tipe Data	Jumlah Karakter	Keterangan
1	<i>id</i>	<i>serial</i>		Nomor data yang bersifat <i>primary key</i> & <i>auto increment</i>
2	<i>user_id</i>	<i>integer</i>		<i>Foreign key</i> dari tabel <i>users</i>
3	<i>location</i>	<i>varchar</i>	255	Lokasi presensi yang berformat koordinat
4	<i>status</i>	<i>varchar</i>	10	Status keberhasilan presensi
5	<i>recorded_at</i>	<i>timestamp</i>		Waktu pencatatan presensi secara otomatis
6	<i>face_image_path</i>	<i>varchar</i>	255	Lokasi penyimpanan foto presensi wajah di <i>server</i>

Selain kolom *id*, tabel *attendance* juga memiliki kolom *user\_id* yang berfungsi untuk mengidentifikasi pengguna yang melakukan presensi dan mengunggah foto wajahnya ke *server*. Kolom *location* untuk mendeteksi lokasi presensi dengan format titik koordinat. Kolom *status* yang berisi status keberhasilan presensi (masuk & keluar). Kolom *recorded\_at* yang merupakan waktu pencatatan presensi secara otomatis dan waktu nyata (prinsip kerjanya sama seperti *created\_at*), dan kolom *face\_image\_path* yang merupakan lokasi penyimpanan data uji wajah ke *server*, dan ditulis dalam bentuk nama jalur lengkap dari *folder* utama hingga nama *file* gambar tersebut beserta ekstensinya (seperti *.jpg*).

#### 4.3.4 Relasi antar tabel

Basis data yang digunakan juga dilengkapi dengan relasi antar tabel, di mana acuan utama dalam membangun relasi tersebut yaitu tabel *users*, dan hubungan yang diterapkan yaitu menggunakan sistem *one-to-many*, dimana satu pengguna dalam tabel *users* dapat terhubung pada tabel *face\_encodings* dan *attendance*.

Relasi antara *users* dan *face\_encodings* meliputi situasi di mana pengguna mengunggah foto latih wajah lebih dari satu kali, agar pengenalan wajah dapat berjalan optimal, sedangkan relasi antara *users* dan *attendance* meliputi keadaan di mana pengguna melakukan presensi setiap kali pegawai hadir di kantor.

#### 4.4 Arsitektur

Di dalam pengembangan aplikasi, terdapat arsitektur yang menjelaskan seluk beluk proses aplikasi berjalan, yang terdiri *front end*, dan *back end*.

##### 4.4.1 Front End

*Front End* merupakan komponen yang dapat menampilkan antarmuka yang dapat berinteraksi langsung dengan pengguna aplikasi. Bagian ini bertanggung jawab untuk menyajikan informasi, menerima masukan, serta memfasilitasi navigasi pengguna terhadap fitur-fitur yang tersedia. Dalam aplikasi ini terdapat berbagai tampilan *front end* yang mendukung sistem presensi berbasis pengenalan wajah ini.

##### 1. Login

Halaman *login* merupakan halaman autentikasi yang mengharuskan pengguna memasukkan kredensial sebelum memperoleh hak akses terhadap fitur-fitur yang ada di dalamnya. Kredensial merupakan data identitas yang digunakan untuk proses autentikasi, biasanya berupa *email* atau nama pengguna (*username*), dan kata sandi (*password*).

##### 2. Register

Halaman *register* merupakan halaman yang digunakan untuk pembuatan akun baru dengan mengisi data Identitas, umumnya berupa nama, *email*, nama pengguna (*username*), dan kata sandi (*password*).

### 3. Generasi Kata Sandi

Halaman generasi kata sandi merupakan halaman untuk menyajikan kata sandi yang dihasilkan secara otomatis oleh sistem, tujuannya agar mendapatkan kata sandi yang kuat dan aman dari potensi ancaman siber. Untuk implementasi aplikasi ini, halaman generasi kata sandi hanya ditampilkan satu kali, yaitu setelah proses registrasi berhasil dilakukan.

### 4. Daftar Data Latih Wajah

Halaman daftar data latih wajah merupakan halaman yang berfungsi untuk mengunggah data latih wajah yang digunakan sebagai referensi pada proses pengenalan wajah setelah registrasi, dimana fitur ini memperbolehkan pengguna untuk mengunggah foto lebih dari satu, dalam konteks penelitian ini, pengguna diminta untuk mengunggah sepuluh data latih wajah yang sudah di rekam sebelumnya.

Gambaran tampilan antarmuka untuk halaman-halaman autentikasi yang dijelaskan di atas disajikan melalui gambar 4.2.



Gambar 4.2 Rancangan halaman autentikasi sistem presensi berbasis pengenalan wajah (dari kiri ke kanan: *login*, *register*, generasi kata sandi, dan daftar data latih wajah)

## 5. Dasbor

Halaman dasbor merupakan halaman utama yang menampilkan navigasi menuju fitur pengenalan wajah, yang juga dilengkapi dengan informasi tambahan yang ditampilkan seperti salam kedatangan untuk pengguna dan bilah bawah yang menampilkan halaman-halaman lain yang ada di dalam aplikasi tersebut, seperti yang ditampilkan pada gambar 4.3.



Gambar 4.3 Rancangan tampilan halaman dasbor

#### 6. Presensi

Halaman presensi merupakan halaman yang berfungsi untuk melakukan presensi dengan basis pengenalan wajah, di dalamnya terdapat tombol untuk menjalankan kamera *internal* untuk mengambil foto dan tombol untuk mengunggah foto tersebut untuk dilakukan proses pengenalan wajah.

Untuk melakukan proses presensi berbasis pengenalan wajah, pengguna diharuskan untuk mengaktifasi kamera *internal* perangkat. Apabila aplikasi digunakan untuk pertama kali, sistem akan menampilkan notifikasi permintaan izin akses kamera. Setelah itu pengguna melakukan pengambilan swafoto sebagai foto uji untuk proses pengenalan wajah, setelah swafoto selesai, pengguna diarahkan kembali ke halaman pengenalan wajah dengan status tombol *Upload* yang aktif, setelah unggahan dan proses pengenalan wajah selesai, halaman hasil pengenalan wajah kemudian ditampilkan. Di dalamnya terdapat keluaran proses presensi, yang mencakup status keberhasilan pengenalan.

Gambaran tampilan antarmuka halaman-halaman proses presensi berbasis pengenalan wajah ditampilkan pada gambar 4.4.



Gambar 4.4 Rancangan tampilan proses presensi berbasis pengenalan wajah (dari kiri ke kanan; halaman eksekusi presensi, dan hasil presensi)

#### 7. *Resume per User*

Halaman *resume per user* merupakan halaman yang menampilkan dokumentasi hasil presensi setiap pegawai, seperti yang ditampilkan di gambar 4.5.



Gambar 4.5 Rancangan tampilan *resume* presensi per pegawai

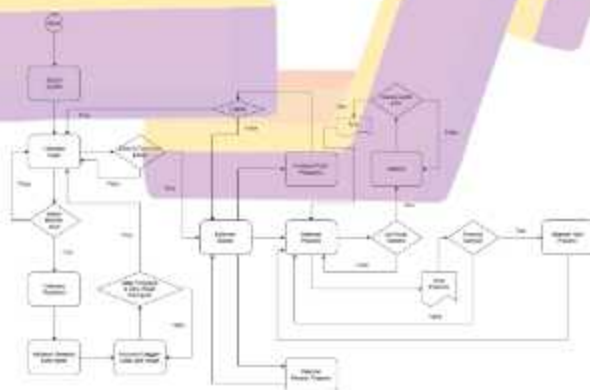
## 8. Halaman profil

Halaman profil merupakan halaman yang menampilkan profil dari pengguna yang sedang *login*, seperti yang ditampilkan di gambar 4.6.



Gambar 4.6 Rancangan tampilan halaman profil

Dalam aplikasi, semua halaman saling terhubung satu sama lain, sehingga menciptakan sebuah pola siklus navigasi yang berulang, seperti di gambarkan pada gambar 4.7.



Gambar 4.7 Interaksi antar *front end*

#### 4.4.2 Back End

*Back End* adalah komponen sistem yang menjalankan proses pengolahan data, pemrosesan logika bisnis, serta penyampaian hasil ke bagian *Front End*. Bagian ini beroperasi di sisi *server* dan bertanggung jawab terhadap integritas, keamanan, dan kecepatan pemrosesan data.

Sistem *back end* meliputi berbagai komponen yang mendukung berjalannya proses operasi sistem, antara lain:

1. Bahasa Pemrograman dan Pustaka

Bahasa pemrograman yang digunakan untuk proses pengenalan wajah menggunakan bahasa *Python* dengan sejumlah pustaka yang mendukung proses pengenalan wajah, antara lain *psycopg2* yang berfungsi untuk melakukan koneksi ke basis data *PostgreSQL*, pustaka *werkzeug.security* untuk melakukan *hashing password*, pustaka *DeepFace* untuk verifikasi wajah, pustaka *logging* untuk pencatatan *error*, dan pustaka *psycopg2.pool.SimpleConnectionPool* untuk menjalankan *connection pool*.

2. Struktur Penyimpanan

Untuk menyimpan data wajah, disediakan satu *folder* utama dengan nama *uploads*, yang di dalamnya terdapat dua *subfolder*, antara lain *subfolder regface* untuk menyimpan foto latih atau registrasi wajah, dan *subfolder attendance* untuk menyimpan foto pada saat melakukan presensi.

3. Manajemen Basis Data

Di dalam kode *Python*, disediakan dua fungsi yang digunakan untuk mengambil atau mengembalikan koneksi ke *connection pool*, yaitu fungsi *get\_db\_connection* dan *release\_db\_connection*. Sedangkan untuk pengolahan data

dan mengirimkannya ke basis data, diperlukan sebuah operasi yang dinamakan *CRUD* (*Create, Read, Update, Delete*) yang terdiri dari fungsi *save\_user\_data* untuk menyimpan data pengguna, fungsi *get\_user\_by\_email* untuk melihat data pengguna yang sedang *login*, fungsi *save\_attendance* untuk menyimpan data presensi, dan fungsi *get\_last\_attendance\_status* untuk melihat status presensi.

#### 4. Keamanan

Untuk keamanan data, diperlukan sejumlah komponen, antara lain fungsi *generate\_password\_hash* dan fungsi *check\_password\_hash* untuk menyimpan kata sandi secara aman (*hashed*), dan mencegah kebocoran data sensitif. Agar koneksi ke basis data lebih efisien, konsep *Connection Pool* diimplementasikan, yang bertujuan agar memungkinkan pengelolaan koneksi secara dinamis dengan batasan satu hingga sepuluh koneksi aktif, sehingga aplikasi dapat menggunakan koneksi tersebut tanpa membuat koneksi baru.

#### 5. Manajemen File Wajah dan Arsitektur FaceNet

Fungsi *helper* yang digunakan untuk menangani penyimpanan dan pengenalan foto wajah antara lain *save\_face\_images* yang bertugas menyimpan foto wajah pengguna di *server* sekaligus mencatat lokasi (*path*) penyimpanan ke basis data. Prosesnya mencakup validasi keberadaan *folder* tujuan dan pemberian nama *file* yang unik, menggunakan kombinasi nilai waktu saat ini (*time.time()*) serta angka acak sebagai *suffix*.

Fungsi *recognize\_face* digunakan untuk membandingkan foto wajah baru dengan foto wajah yang telah terdaftar sebelumnya menggunakan metode:

```
result = DeepFace.verify(face_image, reg_face, model_name="FaceNet")
```

Proses dimulai dengan mendeteksi wajah pada foto *input*, kemudian melakukan perulangan terhadap semua foto wajah yang tersimpan. Selanjutnya dilakukan ekstraksi *embedding* dan perhitungan jarak (*distance*) antar vektor fitur. Hasil perbandingan akan dinyatakan cocok jika jarak tersebut berada di bawah nilai ambang batas (*threshold*), yang ditunjukkan dengan `result['verified'] == True`.

Setelah fungsi `recognize_face` berhasil diimplementasikan, dilakukan tahap pengujian menggunakan *dataset* yang telah disiapkan. *Dataset* terdiri dari sejumlah citra wajah pegawai yang dikelompokkan berdasarkan identitas masing-masing. Setiap *folder* merepresentasikan satu individu dan berisi beberapa citra dengan variasi pose maupun kondisi pencahayaan.

Setelah proses penulisan kode, dilakukan pengujian kode dengan menggunakan *dataset* berupa foto wajah pegawai yang telah direkam. Proses pengujian dilakukan dengan cara membandingkan setiap citra uji terhadap citra wajah yang telah diregistrasikan sebelumnya. Fungsi `recognize_face()` akan menghasilkan keluaran berupa nilai jarak (*distance*) dan status *verified* yang menunjukkan apakah wajah pada citra uji dikenali atau tidak. Evaluasi hasil pengujian dilakukan dengan menghitung jumlah citra uji yang berhasil dikenali dengan benar (*True Positive*), jumlah citra uji yang tidak dikenali (*True Negative*), dan jumlah citra wajah lain yang keliru dikenali (*False Positive*).

## 6. Lain-lain

Aplikasi tersebut juga dilengkapi dengan rekap presensi, yang di dalamnya terdapat berbagai *helper*, yaitu fungsi `resume_attendance_per_user` untuk menampilkan daftar presensi setiap pengguna dalam periode tertentu, dan fungsi

*resume\_attendance\_all\_users* digunakan untuk semua pengguna dalam satu bulan (menghitung keseluruhan *IN/OUT*).

Komponen lainnya antara lain *generate\_random\_password* yang digunakan untuk membuat password acak ketika melakukan registrasi, dan *logging\_error* untuk *debugging*.

#### 4.4.3 API (Application Programming Interface)

*API (Application Programming Interface)* merupakan komponen yang berfungsi untuk menghubungkan aplikasi *mobile* dengan sistem pengenalan wajah di *server*. Melalui *API*, perangkat klien dapat mengakses *endpoint* yang telah disediakan untuk mengirimkan data, memproses permintaan, dan menerima hasil pengenalan wajah secara waktu nyata. Dengan demikian, *API* menjadi komponen penting yang memastikan komunikasi antara aplikasi dan *server* berjalan efektif, aman, dan terstruktur.

Tabel 4.5. menampilkan daftar *endpoint* yang digunakan dalam sistem presensi berbasis pengenalan wajah berikut ini.

Tabel 4.5 Daftar *endpoint API*

No	<i>Endpoint</i>	Metode	Fungsi
1	<i>/register</i>	<i>POST</i>	Melakukan registrasi akun
2	<i>/login</i>	<i>POST</i>	Masuk hak akses ke sistem
3	<i>/register_face</i>	<i>POST</i>	Mendaftarkan foto latih wajah
4	<i>/recognize_face</i>	<i>POST</i>	Melakukan pengenalan wajah
5	<i>/users</i>	<i>GET</i>	Melihat pengguna yang sedang aktif
6	<i>/attendance/resume_user</i>	<i>GET</i>	Melihat dokumentasi presensi setiap pengguna
7	<i>/attendance/resume_all</i>	<i>GET</i>	Melihat dokumentasi presensi semua pengguna

Fungsi-fungsi *endpoint* di tabel 4.5. dijelaskan sebagai berikut:

### 1. Register

*Endpoint /register* merupakan *endpoint* yang berfungsi untuk melakukan registrasi akun baru, *field* yang dapat diisi antara lain nama, *email*, dan password yang di generasi secara otomatis oleh sistem. *Endpoint* ini menggunakan metode *POST* karena berupa proses penyimpanan data.

### 2. Register face

Setelah registrasi akun berhasil, pengguna baru dapat mengunggah foto latihan wajah yang dihubungkan dengan *endpoint /register\_face* dengan metode *POST*, dan *field* yang dikirimkan adalah *foto\_latih* yang digunakan sebagai data latihan untuk sistem presensi berbasis pengenalan wajah.

### 3. Login

Untuk mendapatkan hak akses, pengguna harus melakukan *login* terlebih dahulu dengan memasukkan nama dan kata sandi, *endpoint* yang digunakan adalah */login* dengan metode *POST*. Jika autentikasi berhasil, sistem akan mendapatkan *bearer token* (token autentikasi) yang berfungsi sebagai bukti otorisasi untuk mengakses *endpoint* lain, termasuk ketika mengunggah foto uji.

### 4. Recognize face

Untuk melakukan presensi, pengguna dapat mengunggah foto presensi tersebut melalui *endpoint /recognize\_face* dengan metode *POST*, *field* yang diisi antara lain *foto\_wajah*, dan *geolocation* di generasi secara otomatis ketika pegawai melakukan presensi, token autentikasi juga disertakan agar sistem dapat memastikan bahwa presensi dilakukan oleh pengguna yang telah terdaftar dan sedang *login*.

### 5. *Users*

*Endpoint /users* berfungsi untuk menampilkan profil pengguna yang sedang *login*, metode yang digunakan yaitu *GET* yang merupakan metode untuk menampilkan data. Akses ke *endpoint* ini juga menggunakan token autentikasi untuk memastikan bahwa pengguna yang di tampilkan hanya sesuai dengan pengguna yang aktif.

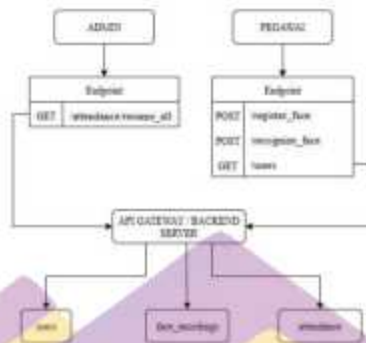
### 6. *Resume User*

*Endpoint /attendance/resume\_user* dengan metode *GET*, berfungsi untuk menampilkan riwayat presensi berdasarkan pengguna yang sedang *login*, akses ke *endpoint* ini juga menggunakan token autentikasi untuk memastikan bahwa riwayat presensi yang di tampilkan hanya berdasarkan pengguna yang sedang *login*.

### 7. *Resume All*

*Endpoint /attendance/resume\_all* dengan metode *GET*, berfungsi untuk menampilkan dokumentasi hasil presensi seluruh pegawai. Akses ke *endpoint* ini biasanya diberikan kepada *admin* untuk keperluan pemantauan dan evaluasi data presensi.

*Endpoint API* tersebut memiliki hak akses yang berbeda, dimana *endpoint* untuk proses pengenalan wajah hanya dilakukan oleh pegawai, sedangkan untuk melihat hasil presensi dilakukan oleh *admin*, seperti yang disusun di gambar 4.8.



Gambar 4.8 Alur *Endpoint API* berdasarkan hak akses

#### 4.5 Evaluasi Model *FaceNet* untuk Pengenalan Wajah

Sebelum di *deploy* ke *server*, script pengenalan wajah yang sudah dibuat kemudian di uji menggunakan dua jenis *dataset*, yaitu publik dan *custom*.

##### 1. Menggunakan *Dataset* Publik

Pengujian awal dilakukan dengan menggunakan *dataset* publik *Labeled Faces in the Wild (LFW)*. *Dataset* ini dipilih karena banyak digunakan sebagai standar pengujian pada penelitian pengenalan wajah.

*Dataset* yang digunakan dalam penelitian ini diperoleh dari laman *Kaggle* dengan kontributor bernama Atul Anand {Jha}<sup>3</sup>, di dalamnya terdapat 5.749 individu dan total 13.244 citra wajah. Untuk keperluan penelitian, pemilihan data dilakukan dengan kriteria setiap individu harus memiliki minimal 10 citra wajah, dan diambil 20 individu yang dapat dijadikan sampel dalam penelitian ini. Selain

<sup>3</sup> <https://www.kaggle.com/datasets/atulanandjha/lfwpeople>

itu dilakukan seleksi apabila ada individu yang memiliki lebih dari 10 citra wajah, seperti yang disajikan dalam gambar 4.9.



Gambar 4.9 Sampel citra wajah dari *dataset* publik

Proses pengujian dilakukan dengan membandingkan citra wajah uji terhadap citra latih yang telah disiapkan, data latih wajah yang diambil sebanyak 20 *set* wajah, masing-masing terdapat 10 foto wajah, dengan keseluruhan 200 foto wajah, dan dilakukan pengujian dengan 10 foto wajah dari 10 nama, kemudian di unggah, dan menghasilkan status verifikasi berupa wajah cocok atau tidak cocok, dan kemudian di kelompokkan ke dalam empat hasil, yaitu *True Positive*, *True Negative*, *False Positive*, dan *False Negative*. Hasil percobaan ditampilkan dalam tabel 4.6.

Tabel 4.6 Hasil pengujian dengan *dataset* publik

No	Status	Jumlah
1	<i>True Positive</i>	7
2	<i>True Negative</i>	0
3	<i>False Positive</i>	0
4	<i>False Negative</i>	3

Dari hasil percobaan diperoleh bahwa sebagian besar wajah berhasil dikenali dengan benar, sementara sebagian kecil mengalami kegagalan pengenalan

akibat kemiripan antar individu atau kualitas citra yang rendah, seperti yang disajikan di gambar 4.10. Secara umum, model *FaceNet* menunjukkan performa yang baik dalam mengenali wajah pada *dataset* publik.



Gambar 4.10 Salah satu perbandingan citra wajah latih dan uji, dimana keluaran yang dihasilkan yaitu tidak cocok

## 2. Menggunakan *Dataset Custom*

Pengujian selanjutnya yaitu menggunakan *dataset custom* yang terdiri dari data citra wajah pegawai Dinas Komunikasi dan Informatika Kota Samarinda yang direkam secara langsung. Prosesnya sama dengan pengujian menggunakan *dataset* publik, yakni dengan membandingkan citra wajah uji terhadap citra latih.

Untuk pengujiannya, dilakukan dengan 20 pegawai terlebih dahulu dengan masing-masing 10 wajah dengan posisi yang berbeda, kemudian dilakukan pengujian dengan melakukan foto terhadap pegawai tersebut, kemudian di unggah, dan menghasilkan status verifikasi berupa wajah cocok atau tidak cocok, dan kemudian di kelompokkan ke dalam empat hasil, yaitu *True Positive*, *True Negative*, *False Positive*, dan *False Negative*. Hasil pengujian dengan *dataset custom* ditampilkan dalam tabel 4.7.

Tabel 4.7 Hasil pengujian dengan *dataset custom*

No	Status	Jumlah
1	<i>True Positive</i>	7
2	<i>True Negative</i>	0
3	<i>False Positive</i>	0
4	<i>False Negative</i>	1

Dari 10 percobaan pertama, diperoleh 8 citra wajah berhasil dieksekusi dengan benar, dimana tujuh di antaranya menghasilkan foto wajah yang cocok dengan data latih yang ada (*True Positive*).

Seluruh evaluasi model *FaceNet* dengan *dataset publik* dan *custom* dilakukan secara *internal* dengan memanfaatkan *localhost* dengan integrasi ke antarmuka *HTML*, sebelum sistem di-*deploy* ke *server*.

#### 4.6 Implementasi Model *FaceNet* sebagai Service pada Server

Sebelum diimplementasikan ke *server*, model *FaceNet* diuji terlebih dahulu dalam lingkungan lokal, dengan menggunakan data latih berupa citra wajah serta data uji citra wajah. Pada saat eksekusi, sistem memberikan umpan balik baik pada sisi *front end* maupun *back end*. Pada *front end*, umpan balik ditampilkan dalam bentuk pesan yang telah di definisikan pada kode program, seperti “Unggahan Berhasil” atau “Unggahan Gagal”. Sementara itu, pada *back end*, informasi ditampilkan melalui *console* dalam bentuk *response code*, misalnya 200 (berhasil), 404 (tidak ditemukan), dan kode status lainnya sesuai hasil pemrosesan, seperti pada tangkapan layar pada gambar 4.11. yang diambil dari ketika eksekusi dilakukan di lingkungan lokal.

```

* Running on http://192.168.1.6:9000/ (Press CTRL+C to quit)
192.168.1.6 - - [22/Sep/2025 21:07:41] "GET /form/test HTTP/1.1" 200 -
192.168.1.6 - - [22/Sep/2025 21:07:41] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [22/Sep/2025 21:08:43] "POST /upload/test HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [22/Sep/2025 21:08:43] "POST /upload/test HTTP/1.1" 200 -
127.0.0.1 - - [22/Sep/2025 21:08:43] "GET /favicon.ico HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [22/Sep/2025 21:08:43] "GET /favicon.ico HTTP/1.1" 404 -

```

Gambar 4.11 Tangkapan layar proses eksekusi *endpoint upload/test* pada saat pengujian melalui lingkungan lokal

Dari hasil uji coba, *service* dapat berjalan dengan baik dan mampu merespon permintaan dalam waktu relatif singkat. Hal ini membuktikan bahwa model dapat dioperasikan secara terpusat dan siap diintegrasikan dengan aplikasi *mobile*.

Pengujian pada lingkungan lokal hanya berfokus pada eksekusi proses pengunggahan citra wajah melalui *endpoint*, namun pada gambar 4.11, terdapat umpan balik yang tidak relevan dengan fokus utamanya, seperti "*GET /favicon.ico HTTP/1.1*" 404, yang muncul karena tidak adanya berkas *favicon* pada sistem, dan pada saat itu belum dilakukan penambahan fitur *login* dan registrasi menggunakan *email* dan generasi password secara otomatis.

Setelah tahap pengujian pada lingkungan lokal berhasil dilakukan, sistem kemudian *di-deploy* ke *server* dan diintegrasikan dengan perangkat telepon genggam untuk mendukung proses pengenalan wajah. Hasil yang diharapkan dari implementasi ini adalah sistem mampu memberikan umpan balik yang konsisten dan sesuai dengan hasil yang diperoleh pada tahap uji coba di lingkungan lokal.

Tahap pertama antara lain melakukan registrasi akun dimana pengguna memasukkan nama, dan alamat *email*. Setelah data disimpan, sistem akan menghasilkan kata sandi yang telah digenerasi secara otomatis di tampilan *front*

*end*, dan pada gambar 4.12. terdapat umpan balik yang dihasilkan melalui kode *back end* yang menampilkan keberhasilan registrasi, kata sandi otomatis, serta *device id*.

```
I/flutter (18051): Response Status: 200
I/flutter (18051): Response Body: {
I/flutter (18051):   "status": "success",
I/flutter (18051):   "message": "Pendaftaran berhasil",
I/flutter (18051):   "data": {
I/flutter (18051):     "password": ██████████,
I/flutter (18051):     "device_id": ██████████
I/flutter (18051):   }
I/flutter (18051): }
```

Gambar 4.12 Respon registrasi berhasil

Tahap selanjutnya adalah mengunggah data latih wajah, dimana setelah 10 foto wajah tersebut berhasil diunggah, sistem memberikan umpan balik melalui *back end* berupa respon & *path* penyimpanan data wajah, sebagaimana disajikan pada Gambar 4.13, dimana umpan balik yang ditampilkan setiap satu foto berhasil di unggah, sehingga memiliki keseluruhan 10 umpan balik yang ditampilkan.

```
I/flutter (18051): Upload Response Status: 200
I/flutter (18051): Upload Response Body: {
I/flutter (18051):   "status": "success",
I/flutter (18051):   "message": "Registrasi wajah berhasil",
I/flutter (18051):   "data": {
I/flutter (18051):     "image_image": {
I/flutter (18051):       "upload_image_face": "face_170448026_070.jpg"
I/flutter (18051):     }
I/flutter (18051):   }
I/flutter (18051): }
I/flutter (18051): Upload Response (status: success, message: Registrasi wajah berhasil, data: {image_image: {face_170448026_070_170448026_070.jpg}})
I/flutter (18051): Upload Response Status: 200
I/flutter (18051): Upload Response Body: {
I/flutter (18051):   "status": "success",
I/flutter (18051):   "message": "Registrasi wajah berhasil",
I/flutter (18051):   "data": {
I/flutter (18051):     "image_image": {
I/flutter (18051):       "upload_image_face": "face_170448026_070.jpg"
I/flutter (18051):     }
I/flutter (18051):   }
I/flutter (18051): }
I/flutter (18051): Upload Response (status: success, message: Registrasi wajah berhasil, data: {image_image: {face_170448026_070_170448026_070.jpg}})
```

Gambar 4.13 Respon berhasil unggah data latih wajah (dua dari 10 respon yang ditampilkan)

Setelah pengguna berhasil melakukan *login*, kemudian dilakukan presensi dengan mengunggah foto wajah yang di uji, dan sistem menampilkan notifikasi berupa status keberhasilan proses pengenalan wajah. Umpan balik dari *back end* tersebut dapat dilihat pada Gambar 4.14.

```

21/Julai 10:51:11 - status Response Status Code: 200
21/Julai 10:51:11 - status Response Body: {}
21/Julai 10:51:11 - "header": "success",
21/Julai 10:51:11 - "message": "Wajah sudah teridentifikasi di database.",
21/Julai 10:51:11 - "data": {}
21/Julai 10:51:11 - "status": "200",
21/Julai 10:51:11 - "timestamp": "Sat, 4 Jul 2020, 10:51:11 GMT",
21/Julai 10:51:11 - "file_upload_path": "uploads/2020/07/04/20200704_105111.jpg"
21/Julai 10:51:11 - {}
21/Julai 10:51:11 - {}
  
```

Gambar 4.14 Respon keberhasilan presensi wajah

Berdasarkan hasil implementasi pada *server*, layanan dapat berjalan dengan baik sebagaimana pada tahap pengujian di lingkungan lokal, serta mampu merespon permintaan dalam waktu yang relatif singkat. Temuan ini menunjukkan bahwa proses implementasi berjalan sesuai harapan. Namun masih terdapat beberapa kendala dalam proses eksekusi, antara lain keterlambatan pada saat pengiriman foto, potensi *downtime* apabila pengunggahan dilakukan secara bersamaan oleh beberapa pengguna, dan kemungkinan masalah *server* yang mengakibatkan *endpoint* tidak berfungsi.

#### 4.7 Cara Kerja Sistem Presensi Berbasis Pengenalan Wajah

Sistem presensi berbasis pengenalan wajah memiliki struktur kerja yang mempermudah penggunaan, dimulai dari registrasi akun agar memperoleh hak akses hingga melakukan presensi.

#### 4.7.1 Autentikasi

Setelah pengguna mendapatkan aplikasi presensi berbasis pengenalan wajah, kemudian mereka membuka aplikasi tersebut, dengan menampilkan *splash screen* sebelum di arahkan ke halaman *login* sebagai halaman pembuka aplikasi. Proses membuat kredensial untuk melakukan autentikasi dijelaskan sebagai berikut.

##### 1. Register

Untuk membuka halaman *register*, pengguna menekan tombol 'Tidak memiliki akun? Daftar disini', dimana tombol tersebut mengarahkan pengguna yang belum memiliki akun ke halaman registrasi. Hal tersebut merepresentasikan pengguna baru yang hendak melakukan registrasi agar mendapatkan hak akses sistem.

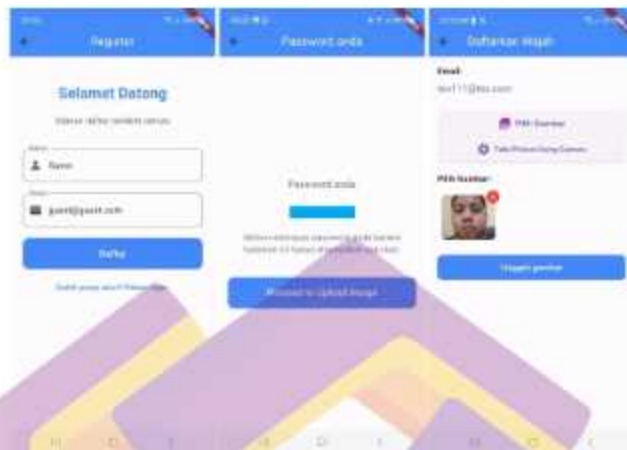
Di dalam halaman tersebut, terdapat formulir untuk mengisi nama dan alamat *email* pengguna yang bersifat unik dan merupakan representasi Identitas tunggal untuk setiap akun yang di simpan di basis data, setelah isian tersebut dilengkapi, pengguna menekan tombol daftar, yang kemudian mengarahkan pengguna ke halaman generasi kata sandi otomatis.

Pada halaman generasi kata sandi, kata sandi tersebut di buat secara otomatis oleh sistem untuk akun baru yang di daftarkan. Data akun dan kata sandi tersebut dikirim melalui *endpoint API/register* untuk selanjutnya di simpan ke basis data melalui tabel *users*, dan sistem memberikan umpan balik dari *back end* yang menunjukkan bahwa proses pendaftaran telah berhasil dilakukan. Selanjutnya pengguna diwajibkan untuk menyimpan kata sandi tersebut dengan cara melakukan tangkap layar (*screenshot*) dan menyimpannya di galeri perangkat, hal ini penting

karena halaman tersebut hanya akan muncul satu kali. Setelah menyimpan kata sandi, pengguna menekan tombol *'Proceed to Upload Image'* untuk selanjutnya di arahkan ke halaman unggah data latih wajah.

Pada halaman unggah data latih wajah, terdapat dua tombol sebagai opsi untuk pemilihan foto latih, yaitu melalui galeri (dengan menekan tombol *'Pilih gambar'*) maupun kamera (dengan menekan tombol *'Ambil gambar melalui kamera'*), dalam konteks ini, pegawai dapat memilih foto dengan memilih tombol *'Pilih gambar'*, di mana sebelum memilih, pegawai diminta untuk mengaktifkan izin akses ke galeri dari telepon genggam masing-masing, kemudian pegawai memilih foto tersebut dengan jumlah maksimal 10 foto, setelah selesai memilih, tekan tombol *'Selesai'*, dan kembali di arahkan ke halaman unggah data latih wajah. Setelah selesai, tekan tombol *'Unggah Gambar'*, kemudian pegawai menunggu hingga proses tersebut selesai. Di sisi *server*, foto yang di unggah di kirim melalui *backend API /register\_face* untuk dilakukan penyimpanan baik di basis data maupun di *server*. Di sisi basis data, foto latih wajah tersebut di simpan melalui tabel *face\_encodings*, di mana satu entri mewakili satu foto, dan di setiap entri di hubungkan dengan tabel *users* melalui atribut *user\_id* yang sesuai dengan akun pengguna yang mendaftar.

Alur dan tampilan proses registrasi akun di tampilan pada gambar 4.15.



Gambar 4.15 Proses registrasi akun

Keluaran yang di dapatkan dalam registrasi antara lain akun yang sudah di daftarkan, *password* yang sudah di generasi otomatis setelah mendaftarkan akun, dan unggahan sampel wajah yang berhasil dimasukkan ke dalam *folder* masing-masing pegawai.

## 2. Login

Setelah proses registrasi selesai, pegawai kemudian di arahkan ke halaman login untuk mendapatkan hak akses ke dalam sistem, pengguna diminta untuk memasukkan alamat *email* dan kata sandi yang sudah di daftarkan sebelumnya, kemudian tekan tombol *Login*, kemudian di arahkan ke halaman dasbor sebagai halaman pembuka setelah pengguna berhasil masuk ke sistem, di sisi *server*, data login dikirim melalui *endpoint API /login*, di mana sistem melakukan verifikasi kredensial pengguna. Jika verifikasi berhasil, sistem akan mengembalikan token

otentikasi yang digunakan sebagai mekanisme keamanan untuk melindungi proses komunikasi data, termasuk pada saat pengiriman data presensi di dalam sistem.

Proses *login* dan halaman dasbor di tampilkan pada gambar 4.16.



Gambar 4.16 Proses login (kiri), dan tampilan halaman dasbor (kanan)

#### 4.7.2 Presensi

Pada halaman dasbor, pengguna memilih menu 'Mulai Pengenalan Wajah' untuk di arahkan ke halaman proses pengenalan wajah, di mana pegawai dapat melakukan presensi dengan melakukan pengenalan wajah, pada halaman tersebut, pengguna menekan tombol 'Capture Image' untuk memulai swafoto, ketika pertama kali dibuka, pegawai akan mendapatkan notifikasi izin penggunaan kamera, dan kemudian pengguna memilih 'Izinkan' untuk mendapatkan akses kamera perangkat, setelah itu pegawai dapat melakukan swafoto, setelah selesai, pegawai dapat memilih antara 'Coba Lagi' atau 'OK' (atau pilihan serupa tergantung perangkat masing-masing), pegawai dapat memilih 'Coba Lagi' jika

dirasa belum maksimal, dan 'OK' jika foto tersebut maksimal. Setelah itu di arahkan kembali ke halaman uji wajah dengan tombol 'Upload Image' sudah aktif, kemudian pilih tombol tersebut untuk menjalankan presensi, dimana dalam proses ini, pengenalan wajah melalui *FaceNet* dilakukan, dan kemudian di eksekusi melalui *endpoint API /recognize\_face* untuk di simpan ke basis data melalui tabel *attendance*, dan menyimpan foto uji wajah tersebut melalui *server*. Setelah berhasil, pegawai di arahkan ke halaman hasil, yang terdapat status Masuk atau Keluar, beserta koordinat yang berasal dari pustaka *Geolocator* dari *framework Flutter*.

Untuk keperluan perhitungan hasil pada penelitian ini, pegawai menunggu hasil pengenalan tersebut hingga 10 menit, jika gagal maka dinyatakan tidak berhasil terunggah. Halaman proses presensi di jelaskan pada gambar 4.17.

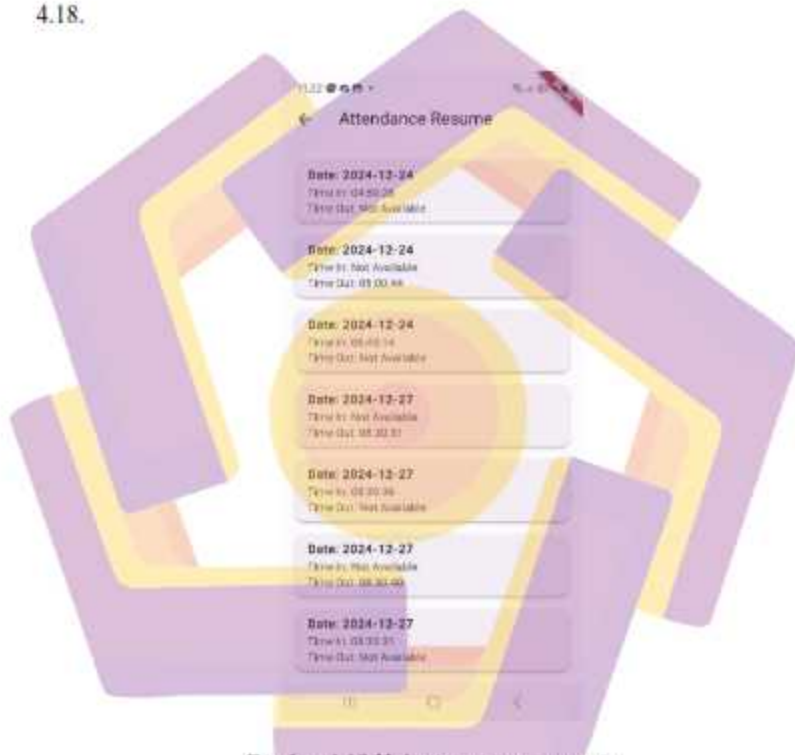


Gambar 4.17 Proses eksekusi pengenalan wajah

#### 4.7.3 Halaman lain

Halaman lain yang turut berperan penting dalam keandalan aplikasi ini antara lain halaman untuk menampilkan riwayat presensi dan profil pengguna yang sedang *login*.

Riwayat presensi pada halaman *resume per user* ditampilkan melalui *endpoint API /attendance/resume\_user* dengan sistem perulangan untuk menampilkan seluruh data presensi dari pegawai tersebut, dan di urutkan dalam format presensi terbaru di susun terlebih dahulu, seperti ditampilkan di gambar 4.18.



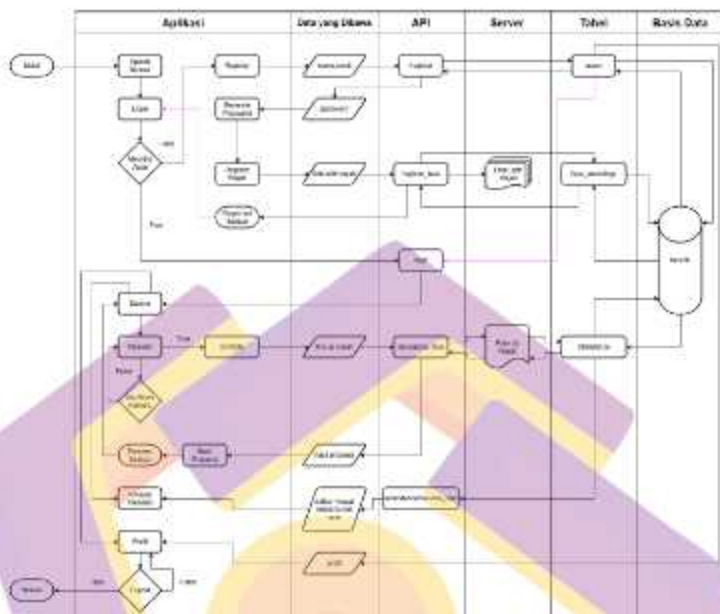
Gambar 4.18 Halaman *resume per user*

Sedangkan halaman profil mengambil dari *endpoint /users* untuk mendapatkan profil pengguna yang sedang *login*, di dalamnya juga terdapat tombol ubah untuk mengubah data diri (termasuk mengubah kata sandi), dan tombol *Logout* untuk keluar dari hak akses, seperti pada gambar 4.19.



Gambar 4.19 Halaman profil

Hubungan antara aplikasi, *server*, *API*, dan basis data di uraikan dalam gambar 4.20.



Gambar 4.20 Hubungan antara aplikasi dengan komponen *back end*

#### 4.8 Evaluasi Sistem Presensi pada Kasus Nyata di Dinas Komunikasi Dan Informatika Kota Samarinda

Pada tahap pengujian, sistem presensi ini diuji pada berbagai kondisi yang mewakili situasi nyata, yaitu dengan memperhatikan variasi pencahayaan dan pose wajah. Pengujian ini dilakukan untuk mengetahui kedua faktor tersebut terhadap kinerja sistem.

##### 1. Pada Variasi Pencahayaan

Evaluasi dilakukan dengan menguji sistem presensi pada kondisi pencahayaan yang berbeda, seperti ruangan terang, ruangan dengan pencahayaan redup, serta kondisi cahaya yang gelap.

## 2. Pada Variasi Pose atau Arah Wajah

Pengujian selanjutnya dilakukan dengan memperhatikan variasi pose wajah, seperti tegak lurus ke kamera, miring ke kanan, dan miring ke kiri.

Penelitian ini telah dilakukan dengan 18 orang yang pose wajahnya telah diuji dengan ponsel masing-masing. Setelah dilakukan penelitian, didapat hasil pengujian dengan rincian secara spesifik sebagai berikut.

### 4.8.1 Hasil Berdasarkan Posisi Wajah

Pengujian dilakukan dengan melakukan swafoto dengan 3 pose berbeda, yaitu posisi tegap, sudut kanan dengan kemiringan sekitar  $90^{\circ}$ , dan sudut kiri dengan kemiringan sekitar  $90^{\circ}$ .



Gambar 4.21 Grafik Hasil Pengenalan Wajah Berdasarkan Posisi Wajah

Pada diagram yang dijelaskan di gambar 4.21., menunjukkan bahwa posisi wajah dapat mempengaruhi hasil pengenalan wajah. Jumlah presensi ketika posisi

wajah tegap menghadap kamera sebanyak 23 presensi masuk, 13 presensi keluar, dan 12 presensi gagal, sedangkan posisi kanan dengan kemiringan sekitar  $90^{\circ}$  menghasilkan 21 presensi masuk, 24 presensi keluar, dan 3 presensi gagal, dan posisi kiri dengan kemiringan sekitar  $90^{\circ}$  menghasilkan 24 presensi masuk, 21 presensi keluar, dan 3 presensi gagal.

Hasil ini menunjukkan bahwa proses presensi di segala posisi wajah berjalan dengan cukup baik, namun masih rentan terhadap kegagalan presensi, terutama ketika hasil pengenalan wajah menunjukkan presensi keluar pada saat posisi wajah menghadap sebelah kanan atau kiri, dan presensi gagal pada saat posisi wajah tegap.

Presensi keluar merupakan kondisi dimana terdapat ketidakcocokan antara wajah yang di unggah untuk presensi dengan data yang telah dilatih sebelumnya, ada beberapa kemungkinan presensi keluar terjadi cukup banyak, antara lain terdapat selisih antar data latih wajah yang secara kasat mata terlihat sama persis, dengan perbedaan kurva pada bagian wajah diantara data tersebut, tangkapan layar ketika hasil presensi keluar dijelaskan pada gambar 4.22.



Gambar 4.22 Tangkapan layar hasil presensi keluar

Presensi gagal merupakan kondisi dimana presensi tidak dapat berjalan dikarenakan faktor teknis, seperti resolusi data wajah, koneksi *internet*, dan latensi *server*, seperti pada tangkapan layar pada gambar 4.23.



Gambar 4.23 Tangkapan layar ketika presensi gagal

#### 4.8.2 Hasil Berdasarkan Intensitas Cahaya

Sebelum pengujian wajah dilakukan, terlebih dahulu dilakukan pengecekan intensitas cahaya untuk mengelompokkan hasil pengenalan wajah berdasarkan intensitas cahaya. Pengukuran dilakukan menggunakan aplikasi pengukur cahaya, yaitu 'Lux', sedangkan untuk rentan cahaya, dibagi menjadi 3 kategori, antara lain rentan gelap dengan intensitas kurang dari 50 *lumen*, rentan remang dengan intensitas antara 50 sampai 99 *lumen*, dan rentan terang dengan intensitas hingga lebih dari 100 *lumen*, seperti yang disusun pada gambar 4.24.



Gambar 4.24 Wajah yang difoto dalam tiga kategori cahaya yang berbeda, dari kiri ke kanan antara lain cahaya gelap, remang, dan terang.



Gambar 4.25 Grafik Hasil Pengenalan Wajah Berdasarkan Intensitas Cahaya

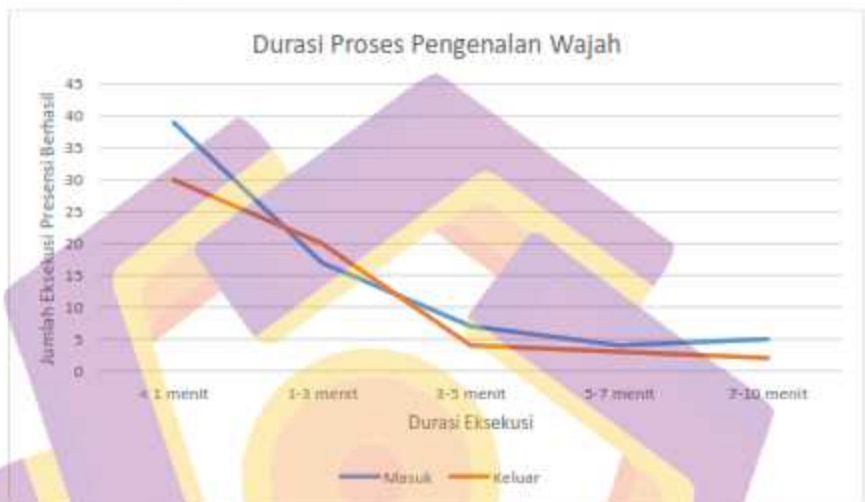
Grafik di gambar 4.25. menunjukkan bahwa cahaya di bawah 50 *lumen* menghasilkan 21 presensi masuk, 22 presensi keluar, dan 5 presensi gagal, sedangkan untuk pencahayaan antara 50 sampai 99 *lumen* juga menghasilkan 21 presensi masuk, 22 presensi keluar, dan 5 presensi gagal, dan kondisi pencahayaan di atas 100 *lumen* menghasilkan 26 presensi masuk, 14 presensi keluar, dan 8 presensi gagal.

Hasil ini menunjukkan bahwa presensi dapat berjalan baik di berbagai kondisi cahaya, terutama dalam posisi cahaya yang lebih terang, namun rentan terhadap presensi keluar ketika kondisi cahaya cenderung gelap atau terlalu terang, yang menyebabkan *FaceNet* tidak mendeteksi secara optimal.

#### 4.8.3 Hasil Berdasarkan Durasi Pengunggahan

Untuk mengetahui seberapa lama proses pengunggahan foto, diperlukan penghitung waktu untuk mengukur lama prosesnya, batas masing-masing foto

antara lain 10 menit, ini termasuk ketika terjadi *error*, dan perlu mengunggah ulang wajah tersebut. Waktu akan berhenti jika proses berhasil, dengan menampilkan notifikasi status pengenalan wajahnya.



Gambar 4.26 Grafik Hasil Pengenalan Wajah Berdasarkan Uji Latensi (lama proses pengenalan wajah)

Pada grafik yang dijelaskan melalui gambar 4.26., proses eksekusi pengenalan wajah lebih banyak diproses kurang dari satu menit dengan total 69 presensi, dimana 39 diantaranya masuk, dan 30 diantaranya keluar. Sedangkan 37 presensi lainnya diproses antara satu sampai tiga menit, dimana 17 diantaranya masuk, dan 20 diantaranya keluar. Adapun 11 presensi lainnya diproses antara tiga sampai lima menit, dimana 7 diantaranya masuk dan 4 diantaranya keluar. Namun terdapat 14 presensi yang membutuhkan waktu lebih dari lima menit untuk mendapatkan hasil pengenalannya, dengan total 9 presensi masuk dan 5 presensi keluar, satu diantaranya terproses lebih dari 10 menit.

Hasil tersebut di atas menunjukkan proses pengenalan wajah dapat diproses hanya dalam kurang dari satu menit, namun tidak sedikit di antara presensi tersebut yang membutuhkan waktu yang lebih lama, faktor-faktor yang membuat proses presensi lebih lama antara lain resolusi wajah yang cukup besar dan memiliki kualitas yang berbeda setiap ponsel masing-masing pegawai, proses pengenalan wajah yang membutuhkan waktu untuk mencocokkan wajah yang di uji dengan yang dilatih, koneksi *internet*, dan latensi dari *server*.

Dari uraian di atas, eksekusi pengenalan wajah cukup baik, dimana foto wajah yang diunggah melalui perangkat pegawai tersebut dapat membaca wajah yang telah di latih. Namun masih memiliki beberapa kekurangan, antara lain deteksi wajah yang masih belum optimal yang disebabkan oleh beberapa faktor seperti objek yang bukan wajah dianggap sebagai wajah, dan lamanya proses komputasi yang disebabkan oleh tingginya resolusi foto wajah dan pengenalan wajah yang dilakukan secara bersamaan, sehingga memungkinkan *server* mengalami kegagalan. Disamping itu, terdapat faktor eksternal seperti pada ponsel tertentu tidak dapat melakukan presensi karena terjadi galat 404 (*endpoint API* tidak ditemukan).

## BAB 5 PENUTUP

### 5.1 Kesimpulan

Penelitian ini bertujuan untuk membangun dan mengembangkan sistem presensi berbasis pengenalan wajah di lingkungan Dinas Komunikasi dan Informatika Kota Samarinda.

Sistem presensi ini dikembangkan menggunakan algoritma *FaceNet*, yang diimplementasikan melalui bahasa pemrograman *Python*, dan diintegrasikan dengan aplikasi telepon genggam melalui *API*. Aplikasi telepon genggam untuk sistem ini dikembangkan menggunakan bahasa pemrograman *Dart* melalui *framework Flutter*, dan penggunaannya dimulai dari autentikasi, pengunggahan data latih wajah, hingga pengunggahan data uji wajah dengan melakukan swafoto, serta pencatatan riwayat presensi.

*FaceNet* berperan untuk mengekstraksi fitur wajah dan membandingkannya dengan data wajah yang telah terdaftar, sehingga dapat menentukan apakah seseorang teridentifikasi sebagai pengguna yang sah atau tidak pada saat proses presensi dilakukan.

Berdasarkan hasil pengujian, sistem presensi berbasis pengenalan wajah dengan menggunakan *FaceNet* menunjukkan performa yang cukup baik dalam hal kinerja sistem pengenalan wajah dari pose dan kondisi pencahayaan yang berbeda. Salah satu faktornya adalah data latih yang digunakan berada di bawah kualitas cahaya yang memadai (tidak terlalu gelap, dan tidak terlalu terang), dan variasi pose

yang beragam. Ini memungkinkan *FaceNet* untuk mempelajari representasi wajah pegawai dengan lebih komperhensif, sehingga eksekusi pencocokan wajah dapat berjalan dengan akurat, baik di saat kondisi cahaya gelap maupun terang, selain itu *FaceNet* juga dapat beradaptasi dengan wajah yang tidak berada pada posisi tegap dengan baik, karena setiap pegawai memiliki representasi vektor dan fitur wajah yang berbeda, sehingga *FaceNet* tetap dapat melakukan pemetaan dan klasifikasi identitas dengan akurat.

Dengan demikian, sistem presensi berbasis pengenalan wajah tersebut layak diimplementasikan dalam lingkungan kerja, dan dapat digunakan sebagai dasar pengembangan sistem presensi otomatis di lingkungan instansi pemerintah.

## 5.2 Saran

Namun masih terdapat kekurangan seperti adanya *noise* maupun *overexposure* ketika eksekusi pengenalan wajah dalam kondisi gelap dan terang, sehingga kegagalan rentan terjadi, terutama ada pose tegap, selain itu terdapat keterbatasan dalam optimasi ponsel, proses pengunggahan wajah yang memakan waktu, maupun kegagalan *server* dalam menerima unggahan ketika foto wajah diunggah secara bersamaan oleh banyak pegawai.

Agar sistem ini dapat berkembang lebih lanjut, disarankan untuk melakukan berbagai pendekatan lain seperti pengujian dengan pencahayaan yang konsisten, penggunaan algoritma yang lebih baik dan efektif, penerapan ke ponsel yang menggunakan sistem operasi lain, pembatasan resolusi wajah, hingga peningkatan latensi pada *server*.

## DAFTAR PUSTAKA

- [1] Cui, Y. (2023). Python How-To: 63 techniques to improve your Python code. Manning Publications. [https://www.google.co.id/books/edition/Python\\_How\\_To/W\\_DDEAAAQBAJ?hl=id&gbpv=1&dq=python&printsec=frontcover](https://www.google.co.id/books/edition/Python_How_To/W_DDEAAAQBAJ?hl=id&gbpv=1&dq=python&printsec=frontcover)
- [2] Panjuta, D., & Nwokike, L. (2024). Tiny Android Projects Using Kotlin. CRC Press.
- [3] Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep Face Recognition.
- [4] Srivastava, D. (2024). Ultimate Python Programming: Learn Python with 650+ programs, 900+ practice questions, and 5 projects (English Edition) (First). BPB Publications.
- [5] Topsakal, Y. (2024). ARTIFICIAL INTELLIGENT TOOLS. [https://www.google.co.id/books/edition/ARTIFICIAL\\_INTELLIGENT\\_TOOLS/0X0yEQAAQBAJ?hl=id&gbpv=1&dq=FACENET&pg=PA61&printsec=frontcover](https://www.google.co.id/books/edition/ARTIFICIAL_INTELLIGENT_TOOLS/0X0yEQAAQBAJ?hl=id&gbpv=1&dq=FACENET&pg=PA61&printsec=frontcover)
- [6] Verma, P., Sankar, Er. D. P., Bhardwaj, A., Chaudhari, V., Pandey, A., & Dumka, A. (2025). Handbook of Deep Learning Models: Vol. One: Fundamentals. CRC Press.
- [7] Alburaiqi, M. S. M., Johar, G. M., Helmi, R. A. A., & Alkawaz, M. H. (2021). Mobile Based Attendance System: Face Recognition and Location Detection using Machine Learning. 2021 IEEE 12th Control and System Graduate Research Colloquium, ICSGRC 2021 - Proceedings, 177–182. <https://doi.org/10.1109/ICSGRC53186.2021.9515221>
- [8] Alhanaee, K., Alhammadi, M., Almenhali, N., & Shatnawi, M. (2021). Face recognition smart attendance system using deep transfer learning. *Procedia Computer Science*, 192, 4093–4102. <https://doi.org/10.1016/j.procs.2021.09.184>
- [9] Anggraini, R., Susita, Prof. Dr. D., & Wiradendi, Dr. C. (2023). THE EFFECT OF WORK DISCIPLINE AND KNOWLEDGE SHARING ON EMPLOYEE PERFORMANCE THROUGH WORK MOTIVATION AS AN INTERVENING VARIABLE AT PETROLEUM AND GAS COMPANIES IN INDONESIA. *Journal of Entrepreneurship*, 30–42. <https://doi.org/10.56943/joe.v2i2.323>
- [10] Budiman, A., Fabian, Yaputera, R. A., Achmad, S., & Kurniawan, A. (2022). Student attendance with face recognition (LBPH or CNN): Systematic literature review. *Procedia Computer Science*, 216, 31–38. <https://doi.org/10.1016/j.procs.2022.12.108>
- [11] Dang, T. V., & Tran, H. L. (2023). A Secured, Multilevel Face Recognition based on Head Pose Estimation, MTCNN and FaceNet. *Journal of Robotics and Control (JRC)*, 4(4), 431–437. <https://doi.org/10.18196/jrc.v4i4.18780>
- [12] Geetha, M., Latha, R. S., Nivetha, S. K., Hariprasath, S., Gowtham, S., &

- Deepak, C. S. (2021). Design of face detection and recognition system to monitor students during online examinations using Machine Learning algorithms. 2021 International Conference on Computer Communication and Informatics.
- [13] 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>
- [14] Liu, Y., & Qu, Y. (2024). Construction of a smart face recognition model for university libraries based on FaceNet-MMAR algorithm. *PLoS ONE*, 19(1 January). <https://doi.org/10.1371/journal.pone.0296656>
- [15] Mensah, J. A., Appati, J. K., Boateng, E. K. A., Ocran, E., & Asiedu, L. (2024). FaceNet recognition algorithm subject to multiple constraints: Assessment of the performance. *Scientific African*, 23. <https://doi.org/10.1016/j.sciaf.2023.e02007>
- [16] Oloyede, M. O., Hancke, G. P., & Myburgh, H. C. (2020). A review on face recognition systems: recent approaches and challenges. *Multimedia Tools and Applications*, 79(37–38), 27891–27922. <https://doi.org/10.1007/s11042-020-09261-2>
- [17] Rollon, L. A., Lusing, C. A., Aquino, J. B., Centeno, C., & Sison, A. A. R. (2023). Attendance system based on facial recognition using multi-task convolutional neural network. *World Journal of Advanced Research and Reviews*, 18(3), 001–006. <https://doi.org/10.30574/wjarr.2023.18.3.0971>
- [18] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A Unified Embedding for Face Recognition and Clustering. <https://doi.org/10.1109/CVPR.2015.7298682>
- [19] Sunaryono, D., Siswantoro, J., & Anggoro, R. (2021). An android based course attendance system using face recognition. *Journal of King Saud University - Computer and Information Sciences*, 33(3), 304–312. <https://doi.org/10.1016/j.jksuci.2019.01.006>
- [20] Wu, C., & Zhang, Y. (2021). MTCNN and FACENET Based Access Control System for Face Detection and Recognition. *Automatic Control and Computer Sciences*, 55(January 2021), 102–112.
- [21] Yang, H., & Han, X. (2020). Face recognition attendance system based on real-time video processing. *IEEE Access*, 8, 159143–159150. <https://doi.org/10.1109/ACCESS.2020.3007205>
- [22] Zysman, J., & Costinot, A. (2022). Medical Research, Nursing, Health and Midwife Participation THE INFLUENCE OF WORK DISCIPLINE AND WORKLOAD ON EMPLOYEE PERFORMANCE (Study on Community Empowerment for Helath Service Employees At California District). <https://medalionjournal.com/>

## LAMPIRAN

### Lampiran 1 Kode Program (utils/helpers.py)

```
import psycopg2
from psycopg2 import pool
from werkzeug.security import generate_password_hash,
check_password_hash
import os
import random
import string
import time
from deepface import DeepFace
from werkzeug.utils import secure_filename
import logging
from datetime import datetime

# Konfigurasi connection pool
DB_POOL = psycopg2.pool.SimpleConnectionPool(
    1, 10, host="localhost", port="5432", dbname="faceatt_db",
    user="faceatt_user", password="faceatt@123"
)

# Folder untuk menyimpan file upload
UPLOAD_FOLDER = 'uploads'
os.makedirs(UPLOAD_FOLDER, exist_ok=True)
# Folder untuk menyimpan file upload lebih spesifik
REGFACE_FOLDER = os.path.join(UPLOAD_FOLDER, 'regface')
ATTENDANCE_FOLDER = os.path.join(UPLOAD_FOLDER,
'attendance')
os.makedirs(REGFACE_FOLDER, exist_ok=True)
os.makedirs(ATTENDANCE_FOLDER, exist_ok=True)

# Fungsi untuk mengambil koneksi dari pool
def get_db_connection():
    return DB_POOL.getconn()

# Fungsi untuk melepaskan koneksi kembali ke pool
def release_db_connection(conn):
    DB_POOL.putconn(conn)

# Fungsi untuk menyimpan data pengguna ke database
def save_user_data(email, name, password, device_ID):
    try:
        conn = get_db_connection()
        cursor = conn.cursor()

        hashed_password = generate_password_hash(password)
        cursor.execute("""
```

```

INSERT INTO users (name, email, password, hashed_password,
device_ID)
VALUES (%s, %s, %s, %s, %s)
""" , (name, email, password, hashed_password, device_ID))

conn.commit()
except Exception as e:
    logging.error(f'Error saving user data: {e}')
finally:
    release_db_connection(conn)

# Fungsi untuk mendapatkan pengguna berdasarkan email
def get_user_by_email(email):
    conn = get_db_connection()
    cursor = conn.cursor()

    cursor.execute("SELECT ID, email, password, hashed_password,
device_ID FROM users WHERE email = %s", (email,))
    user = cursor.fetchone()

    release_db_connection(conn)

    if user:
        return {"ID": user[0], "email": user[1], "password": user[2],
"hashed_password": user[3], "device_ID": user[4]}
    return None

# Fungsi untuk memverifikasi password
def verify_password(stored_password, input_password):
    return check_password_hash(stored_password, input_password)

# Fungsi untuk menyimpan foto wajah
def save_face_images(email, face_images):
    user = get_user_by_email(email)
    if not user:
        return None # User tidak ditemukan

    user_ID = user['ID'] # Ambil ID pengguna dari database
    saved_paths = []
    uploads_dir = 'uploads/regface'

    if not os.path.exists(uploads_dir):
        os.makedirs(uploads_dir) # Membuat direktori jika belum ada

    for IDx, face_image in enumerate(face_images):
        try:
            # Tentukan path untuk menyimpan gambar dengan nama unik

```

```

        file_name =
f"{user_ID}_face_{int(time.time())}_{random.randint(1000, 9999)}.jpg"
        file_path = os.path.join(uploads_dir, file_name)

        # Simpan file gambar
        face_image.save(file_path)

        # Simpan path ke database
        conn = get_db_connection()
        cursor = conn.cursor()
        cursor.execute("INSERT INTO face_encodings (user_ID,
image_path) VALUES (%s, %s)", (user_ID, file_path))
        conn.commit()
        release_db_connection(conn)

        saved_paths.append(file_path)
    except Exception as e:
        logging.error(f"Error saving face image: {e}")

    return saved_paths if saved_paths else None

# Fungsi untuk mengenali wajah menggunakan DeepFace
def recognize_face(email, face_image):
    registered_face_folder = REGFACE_FOLDER
    attendance_face_folder = ATTENDANCE_FOLDER

    # Pastikan folder presensi ada
    os.makedirs(attendance_face_folder, exist_ok=True)

    print("Type of face_image:", type(face_image))
    print("Value of face_image:", face_image)

    if not os.path.exists(registered_face_folder):
        return False, "User not found"

    # Ambil gambar wajah terdaftar
    registered_faces = [os.path.join(registered_face_folder, f) for f in
os.listdir(registered_face_folder) if f.endswith('.jpg')]
    if not registered_faces:
        return False, "No face registered for this user"

    try:
        # Bandingkan dengan gambar yang terdaftar
        for reg_face in registered_faces:
            try:
                # result = DeepFace.verify(face_image, reg_face,
model_name="VGG-Face") # akurat masih di atas 1 detik

```

```

        result = DeepFace.verify(face_image, reg_face,
model_name="FaceNet") # akurat masih di atas 1 detik
        # result = DeepFace.verify(face_image, reg_face,
model_name="OpenFace") # tidak akurat
        print(result)
        if result['verified']:
            # Hapus file sementara setelah verifikasi berhasil
            # os.remove(temp_image_path)
            return True, "Face matched"
        except Exception as e:
            logging.error(f'Error during face verification: {e}')
            continue

    except Exception as e:
        logging.error(f'Error get face images: {e}')
        return False, "Failed to get face images"

    return False, "Face not recognized"

# Fungsi untuk menyimpan data presensi
def save_attendance(user_ID, location, status, face_image_path):
    conn = get_db_connection()
    cursor = conn.cursor()

    try:
        cursor.execute("""
            INSERT INTO attendance (user_ID, location, status,
face_image_path)
            VALUES (%s, %s, %s, %s)
            """, (user_ID, location, status, face_image_path))
        conn.commit()
    finally:
        release_db_connection(conn)

# Fungsi untuk mengambil data presensi terakhir
def get_last_attendance_status(user_ID):
    conn = get_db_connection()
    cursor = conn.cursor()

    cursor.execute("""
        SELECT status FROM attendance
        WHERE user_ID = %s
        ORDER BY recorded_at DESC LIMIT 1
        """, (user_ID,))
    last_record = cursor.fetchone()
    release_db_connection(conn)

    return last_record[0] if last_record else None # 'IN', 'OUT', atau None

```

```

# Fungsi untuk mengambil seluruh data pengguna
def load_user_data():
    conn = get_db_connection()
    cursor = conn.cursor()

    cursor.execute("SELECT name, email, device_ID FROM users")
    users = cursor.fetchall()

    release_db_connection(conn)

    user_list = []
    for user in users:
        user_list.append({
            "name": user[0],
            "email": user[1],
            "device_ID": user[2]
        })

    return user_list

# Fungsi untuk membuat password acak
def generate_random_password(length=8):
    """Generate a random password with a default length of 8
    characters."""
    characters = string.ascii_letters + string.digits + "!@#$%^&*()"
    password = "".join(random.choice(characters) for _ in range(length))
    return password

# Fungsi untuk resume kehadiran per user per bulan
def resume_attendance_per_user(user_ID, month, year):
    conn = get_db_connection()
    cursor = conn.cursor()

    cursor.execute("""
        SELECT recorded_at, status FROM attendance
        WHERE user_ID = %s AND EXTRACT(MONTH FROM
        recorded_at) = %s AND EXTRACT(YEAR FROM recorded_at) = %s
        ORDER BY recorded_at
        """, (user_ID, month, year))
    records = cursor.fetchall()

    release_db_connection(conn)

    attendance_resume = []
    for record in records:
        date = record[0].strftime("%Y-%m-%d")

```

```

        time_in = record[0].strftime("%H:%M:%S") if record[1] == 'IN' else
None
        time_out = record[0].strftime("%H:%M:%S") if record[1] == 'OUT'
else None
        attendance_resume.append({
            "date": date,
            "time_in": time_in,
            "time_out": time_out
        })

    return attendance_resume

# Fungsi untuk resume kehadiran semua user dalam 1 bulan
def resume_attendance_all_users(month, year):
    conn = get_db_connection()
    cursor = conn.cursor()

    cursor.execute("""
        SELECT u.name, u.email, COUNT(CASE WHEN a.status = 'IN'
THEN 1 END) as total_in,
        COUNT(CASE WHEN a.status = 'OUT' THEN 1 END) as
total_out
        FROM users u
        LEFT JOIN attendance a ON u.ID = a.user_ID
        WHERE EXTRACT(MONTH FROM a.recorded_at) = %s AND
EXTRACT(YEAR FROM a.recorded_at) = %s
        GROUP BY u.ID
    """, (month, year))

    records = cursor.fetchall()
    release_db_connection(conn)

    attendance_summary = []
    for record in records:
        attendance_summary.append({
            "name": record[0],
            "email": record[1],
            "total_in": record[2],
            "total_out": record[3]
        })

    return attendance_summary

```

#### Lampiran 2 Kode Program (app.py)

```

from flask import Flask, request, jsonify
from flask_restx import Api, Resource, fields
from flask_httpauth import HTTPBasicAuth

```

```

import os
import time
from utils.helpers import (
    generate_random_password, save_face_images, save_user_data,
    recognize_face, verify_password, get_user_by_email,
    load_user_data, get_last_attendance_status, save_attendance,
    resume_attendance_per_user, resume_attendance_all_users
)

# Inisialisasi Flask, API, dan otentikasi
app = Flask(__name__)
api = Api(app, version="1.0", title="Face Attendance API",
description="API untuk presensi berbasis wajah menggunakan
Flask")
auth = HTTPBasicAuth()

# Variabel untuk otentikasi (misal: user dan password)
USERS = {
    "developer": "pwd@mik0m" # Ganti dengan password yang lebih kuat
}

# Fungsi untuk otentikasi
@auth.verify_password
def verify_password api(username, password):
    if username in USERS and USERS[username] == password:
        return username
    return None

# Folder untuk menyimpan file upload
UPLOAD_FOLDER = 'uploads'
os.makedirs(UPLOAD_FOLDER, exist_ok=True)
# Folder untuk menyimpan file upload lebih spesifik
REGFACE_FOLDER = os.path.join(UPLOAD_FOLDER, 'regface')
ATTENDANCE_FOLDER = os.path.join(UPLOAD_FOLDER,
'attendance')
os.makedirs(REGFACE_FOLDER, exist_ok=True)
os.makedirs(ATTENDANCE_FOLDER, exist_ok=True)

### DEFINISI MODELS UNTUK VALIDASI INPUT ###
register_model = api.model('RegisterUser', {
    'name': fields.String(required=True, description='Nama lengkap
pengguna'),
    'email': fields.String(required=True, description='Email pengguna'),
    'device ID': fields.String(required=True, description='Device ID
pengguna'),
})

```

```

login_model = api.model('LoginUser', {
    'email': fields.String(required=True, description='Email pengguna'),
    'password': fields.String(required=True, description='Password
pengguna'),
})

attendance_model = api.model('RecognizeFace', {
    'email': fields.String(required=True, description='Email pengguna'),
    'location': fields.String(required=True, description='Lokasi saat
presensi'),
})

resume_user_model = api.model('ResumeUser', {
    'user_ID': fields.String(required=True, description='ID pengguna untuk
resume kehadiran'),
    'month': fields.Integer(required=True, description='Bulan untuk resume
kehadiran'),
    'year': fields.Integer(required=True, description='Tahun untuk resume
kehadiran')
})

resume_all_model = api.model('ResumeAllUsers', {
    'month': fields.Integer(required=True, description='Bulan untuk resume
kehadiran semua pengguna'),
    'year': fields.Integer(required=True, description='Tahun untuk resume
kehadiran semua pengguna')
})

### ENDPOINT REGISTER USER ###
@api.route('/register')
class RegisterUser(Resource):
    @api.expect(register_model)
    def post(self):
        """Mendaftarkan pengguna baru"""
        data = request.get_json()
        name = data.get('name')
        email = data.get('email')
        device_ID = data.get('device_ID')

        if not name or not email or not device_ID:
            return {"status": "fail", "message": "Kolom yang diperlukan tidak
lengkap", "data": {}}, 400

        existing_user = get_user_by_email(email)
        if existing_user:
            return {"status": "fail", "message": "Pengguna dengan email ini
sudah terdaftar", "data": {}}, 400

```

```

password = generate_random_password()
save_user_data(email, name, password, device_ID)

return {
    "status": "success",
    "message": "Pendaftaran berhasil",
    "data": {"password": password, "device_ID": device_ID}
}, 200

### ENDPOINT LOGIN ###
@api.route('/login')
class LoginUser(Resource):
    @api.expect(login_model)
    def post(self):
        """Login pengguna"""
        data = request.get_json()
        email = data.get('email')
        password = data.get('password')

        user = get_user_by_email(email)
        if user and verify_password(user['hashed_password'], password):
            return {
                "status": "success",
                "message": "Login berhasil",
                "data": {"email": email, "device_ID": user['device_ID']}
            }, 200
        return {
            "status": "fail",
            "message": "Tidak berwenang",
            "data": {}
        }, 401

### ENDPOINT REGISTER FACE ###
@api.route('/register_face')
class RegisterFace(Resource):
    @api.param('email', 'Email pengguna', required=True)
    @api.doc(params={'faces': 'Gambar wajah pengguna untuk pendaftaran'}, security='Bearer')
    @auth.login_required
    def post(self):
        """Mendaftarkan wajah pengguna"""
        data = request.form
        email = data.get('email')

        if not email:
            return {"status": "fail", "message": "Email diperlukan", "data":
            {}}, 400

```

```

if 'faces' not in request.files:
    return {"status": "fail", "message": "Gambar wajah tidak disertakan", "data": {}}, 400

face_images = request.files.getlist('faces')
if len(face_images) < 1:
    return {"status": "fail", "message": "Minimal 1 gambar wajah diperlukan", "data": {}}, 400

saved_paths = save_face_images(email, face_images)
if not saved_paths:
    return {"status": "fail", "message": "Registrasi wajah gagal. Coba lagi.", "data": {}}, 400

return {
    "status": "success",
    "message": "Registrasi wajah berhasil",
    "data": {"saved_images": saved_paths}
}, 200

### ENDPOINT ATTENDANCE ###
@api.route('/recognize_face')
class RecognizeFace(Resource):
    @api.param('email', 'Email pengguna', required=True)
    @api.param('location', 'Lokasi saat presensi', required=True)
    @api.doc(params={'face': 'Gambar wajah pengguna untuk pencocokan'}, security='Bearer')
    @auth.login_required
    def post(self):
        """Presensi masuk/keluar pengguna berdasarkan wajah"""
        data = request.form
        email = data.get('email')
        location = data.get('location')

        if not email or not location or 'face' not in request.files:
            return {"status": "fail", "message": "Email, lokasi, dan gambar wajah diperlukan", "data": {}}, 400

        face_image = request.files['face']
        user = get_user_by_email(email)
        if not user:
            return {"status": "fail", "message": "Pengguna tidak ditemukan", "data": {}}, 404

        user_ID = user['ID']
        face_image_path = os.path.join('uploads/attendance',
f'attendance_{user_ID}_{int(time.time())}.jpg')
        face_image.save(face_image_path)

```

```

match, message = recognize_face(email, face_image_path)
if match:
    last_status = get_last_attendance_status(user_ID)
    new_status = 'OUT' if last_status == 'IN' else 'IN'
    save_attendance(user_ID, location, new_status, face_image_path)

    return {
        "status": "success",
        "message": f"Wajah cocok, kehadiran {new_status} dicatat.",
        "data": {
            "status": new_status,
            "location": location,
            "face_image_path": face_image_path
        }
    }, 200
else:
    if os.path.exists(face_image_path):
        os.remove(face_image_path)
    return {
        "status": "fail",
        "message": message,
        "data": {}
    }, 400

### ENDPOINT USERS ###
@api.route('/users')
class Users(Resource):
    @auth.login_required
    def get(self):
        """Melihat daftar pengguna yang sudah terdaftar"""
        users = load_user_data()
        if not users:
            return {
                "status": "fail",
                "message": "Tidak ada pengguna yang ditemukan",
                "data": {}
            }, 404
        return {
            "status": "success",
            "message": "Data pengguna ditemukan",
            "data": {"users": users}
        }, 200

### ENDPOINT RESUME KEHADIRAN PER USER ###
@api.route('/attendance/resume_user')
class ResumeUser(Resource):
    @api.expect(resume_user_model)

```

```

@auth.login_required
def get(self):
    """Resume kehadiran per pengguna berdasarkan bulan dan tahun"""
    data = request.args
    user_ID = data.get('user_ID')
    month = int(data.get('month'))
    year = int(data.get('year'))

    if not user_ID or not month or not year:
        return {
            "status": "fail",
            "message": "Parameter yang diperlukan tidak lengkap",
            "data": {}
        }, 400

    resume = resume_attendance_per_user(user_ID, month, year)
    return {
        "status": "success",
        "message": "Resume kehadiran berhasil diambil",
        "data": {"attendance_resume": resume}
    }, 200

### ENDPOINT RESUME KEHADIRAN SEMUA USER ###
@api.route('/attendance/resume_all')
class Resume.AllUsers(Resource):
    @api.expect(resume_all_model)
    @auth.login_required
    def get(self):
        """Resume kehadiran semua pengguna dalam bulan dan tahun
        tertentu"""
        data = request.args
        month = int(data.get('month'))
        year = int(data.get('year'))

        if not month or not year:
            return {
                "status": "fail",
                "message": "Parameter yang diperlukan tidak lengkap",
                "data": {}
            }, 400

        summary = resume_attendance_all_users(month, year)
        return {
            "status": "success",
            "message": "Resume kehadiran semua pengguna berhasil
            diambil",
            "data": {"attendance_summary": summary}
        }, 200

```

```
if __name__ == '__main__':  
    app.run(host="0.0.0.0", port=5000, debug=True)
```

