

**PENGEMBANGAN SISTEM PRESENSI KARYAWAN
DENGAN *FACE RECOGNITION* MENGGUNAKAN MODEL
FACENET**



Disusun Oleh:

N a m a : Anjeli Puteri Anggreini
NIM : 20523151

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2024**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**PENGEMBANGAN SISTEM PRESENSI KARYAWAN
DENGAN *FACE RECOGNITION* MENGGUNAKAN MODEL
FACENET**

TUGAS AKHIR JALUR MAGANG



N a m a : Anjeli Puteri Anggreini
NIM : 20523151

الجامعة الإسلامية
الاندونيسية

Yogyakarta, 05 Juli 2024

Pembimbing,

A handwritten signature in blue ink, appearing to read 'Sri Kusumadewi', is written over a horizontal line.

Dr. Sri Kusumadewi, S. Si, M. T.

HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN SISTEM PRESENSI KARYAWAN
DENGAN *FACE RECOGNITION* MENGGUNAKAN MODEL
FACENET**

TUGAS AKHIR JALUR MAGANG

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 4 Juli 2024

Tim Penguji

Dr. Sri Kusumadewi, S. Si, M. T.

Anggota 1

Dr. Syarif Hidayat, S.T., M.I.T.

Anggota 2

Chandra Kusuma Dewa, S.Kom., M.Cs.,
Ph.D.



Mengetahui,

Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia



Thomas Hatta Fudholi, S.T., M.Eng., Ph.D.

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Anjeli Puteri Anggreini

NIM : 20523151

Tugas akhir dengan judul:

**PENGEMBANGAN SISTEM PRESENSI KARYAWAN
DENGAN *FACE RECOGNITION* MENGGUNAKAN MODEL
FACENET**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 4 Juli 2024



Anjeli Puteri Anggreini

HALAMAN PERSEMBAHAN

Puji syukur penulis panjatkan kepada Allah SWT atas segala nikmat yang telah diberikan dalam perjalanan menyelesaikan tugas akhir ini yang berjudul “Pengembangan Sistem Presensi Karyawan dengan *Face Recognition* Menggunakan Model FaceNet”. Penulis juga mengucapkan terima kasih yang sebesar-besarnya kepada orang tua, keluarga, dan teman dekat yang selalu memberikan dukungan, semangat, dan doa dalam perjalanan penulisan laporan tugas akhir ini. Dukungan dan kasih sayang yang diberikan menjadi pendorong utama dalam menghadapi berbagai tantangan selama proses penulisan. Semoga laporan tugas akhir ini dapat memberikan manfaat dan kontribusi yang positif bagi pengembangan teknologi.

HALAMAN MOTO

“Sesungguhnya Allah tidak akan merubah nasib suatu kaum kecuali mereka yang merubah keadaan diri mereka sendiri”

- Q.S. Ar Rad: 11

“This hope is the most obvious reality. If you believe that hope will come true”

- Bill Gates

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh,

Puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah sehingga penulis dapat menyelesaikan tugas akhir ini dengan judul “Pengembangan Sistem Presensi Karyawam dengan *Face Recognition* Menggunakan Model FaceNet”. Tugas akhir ini disusun sebagai salah satu syarat untuk menyelesaikan Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Islam Indonesia.

Penulisan tugas akhir ini tidak lepas dari bantuan, bimbingan, dan dukungan berbagai pihak yang telah memberikan kontribusi besar selama proses penyunan, baik secara langsung maupun tidak langsung. Bantuan mereka sangat berharga dalam menyelesaikan berbagai tantangan yang dihadapi selama penulisan sehingga penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT yang telah memberikan nikmat kesehatan, kekuatan, dan kesempatan sehingga penulis dapat menyelesaikan tugas akhir.
2. Bapak dan Ibu Penulis, Bapak Erson dan Ibu Zurwana yang selalu memberikan doa, dukungan moral, dan semangat yang tiada henti dalam setiap langkah yang penulis tempuh.
3. Saudara – saudara penulis, Erdi Kurniawan, Putri Tiara, Puja Puteri, Angie Putri, dan Salwa Putri yang telah memberikan semangat kepada penulis selama menempuh pendidikan.
4. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Jurusan Informatika Universitas Islam Indonesia.
5. Bapak DThomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Ketua Program Studi Informatika – Program Sarjana.
6. Ibu Dr. Sri Kusumadewi, S.Si., M. T. selaku dosen pembimbing yang telah memberikan bimbingan, arahan, dan motivasi selama proses penyusunan tugas akhir.
7. Seluruh dosen Program Studi Informatika yang telah berbagi ilmu dan pengalaman selama penulis menempuh studi di Universitas Islam Indonesia.
8. Bapak Agus Setiawan selaku CEO PT Indo Xetia Internasional yang telah memberikan ilmu selama pelaksanaan magang.
9. Seluruh karyawan PT Indo Xetia Internasional yang telah memberikan kesempatan dan pengalaman selama pelaksanaan magang.

10. Rekan - rekan *student staff* yang telah memberikan dukungan dan kebersamaan dalam menghadapi berbagai tantangan selama masa studi.
11. Teman – teman seperjuangan Faridah Amaliyah, Hestri Apriliani, Indah Rahma, Irfan Rafid, Sandi Wijaya, dan Syafnides Wulan.
12. Seluruh anggota rifanda kost yang telah menemani dan memberikan dukungan selama penulisan tugas akhir.

Penulisan menyadari bahwa laporan tugas akhir ini masih jauh dari sempurna sehingga penulis mengharapkan saran dan kritik yang membangun untuk perbaikan di masa mendatang. Semoga tugas akhir ini dapat memberikan manfaat bagi pembaca dan menjadi kontribusi positif dalam pengembangan teknologi.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Yogyakarta, 4 Juli 2024



Anjeli Puteri Anggreini

SARI

PT Indo Xetia Internasional adalah perusahaan teknologi yang berfokus pada pengembangan sistem informasi untuk berbagai industri dan menerapkan sistem kerja yang memungkinkan karyawannya untuk bekerja dari rumah (*work from home*). Sistem kerja ini mengharuskan perusahaan untuk memiliki sistem presensi yang dapat memvalidasi kehadiran karyawan secara akurat. Saat ini, karyawan menggunakan aplikasi Discord untuk melakukan presensi dan terdapat kelemahan dalam penggunaan aplikasi Discord yang berpotensi bagi seseorang yang bukan karyawan dapat melakukan presensi atas nama karyawan yang sebenarnya tidak hadir sehingga untuk meningkatkan akurasi dan keamanan pada proses presensi maka dikembangkan sistem presensi karyawan menggunakan teknologi *face recognition*.

Proses sistem presensi karyawan melibatkan beberapa tahapan utama seperti *face detection*, *face registration*, dan *face recognition*. Pada tahap *face detection* maka algoritma SSD bekerja untuk mengidentifikasi wajah pada gambar. Setelah wajah terdeteksi maka dilakukan *pre-processing* untuk normalisasi, transformasi vektor, dan *cropping*. Pada tahap *face registration* karyawan mengambil gambar wajah menggunakan kamera seluler secara *real-time* untuk digunakan sebagai data referensi. Pada tahap *face recognition* maka model FaceNet bekerja untuk menghasilkan vektor fitur wajah yang telah terdaftar menggunakan perhitungan jarak Euclidean untuk menentukan tingkat kemiripan. Jika hasil nilai jarak Euclidean di bawah atau sama dengan *threshold* yang ditentukan maka wajah dianggap dikenali dan pengguna diizinkan melakukan presensi.

Hasil pengujian sistem pengenalan wajah menunjukkan kinerja yang bervariasi tergantung pada kondisi pengujian. Pengujian pada cahaya terang mencapai tingkat keberhasilan 98.7% dan pengujian pada cahaya redup mencapai tingkat keberhasilan 92%. Pada pengujian posisi miring memiliki tingkat keberhasilan 54.7% dan pengujian pada posisi lurus memiliki tingkat keberhasilan 96%. Pada pengujian ekspresi tersenyum menunjukkan tingkat keberhasilan 94.7% dan pengujian pada ekspresi datar menunjukkan tingkat keberhasilan 100%. Pada pengujian jarak 30cm mencapai tingkat keberhasilan 100% dan pengujian jarak 60cm menunjukkan tingkat keberhasilan 98.7%. Pada pengujian menggunakan atribut topi memiliki tingkat keberhasilan 96% dan pengujian menggunakan atribut kacamata mencapai tingkat keberhasilan sebesar 97.3%.

Kata kunci: sistem presensi, *face recognition*, Flutter, Tensorflow Lite, Google ML Kit

GLOSARIUM

Bug	kesalahan dalam kode yang menyebabkan perangkat lunak tidak berfungsi dengan benar.
Debug	proses menemukan dan memperbaiki kesalahan atau cacat dalam kode program agar perangkat lunak dapat berfungsi dengan benar.
Dependency	relasi antara dua komponen atau modul di mana satu komponen membutuhkan yang lain untuk berfungsi dengan benar.
Profiling	proses analisis kinerja suatu program untuk mengidentifikasi area-area yang memerlukan perbaikan atau optimasi.
Noise	gangguan atau informasi yang tidak relevan sehingga dapat mengganggu atau mengaburkan sinyal yang diinginkan dalam suatu sistem.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM	x
DAFTAR ISI	xi
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Ruang Lingkup	3
1.3 Tujuan	3
1.4 Manfaat	4
1.5 Sistematika Penulisan	4
BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA	6
2.1 Sistem Presensi	6
2.2 Deteksi Wajah (<i>Face Detection</i>)	7
2.3 Pengenalan Wajah (<i>Face Recognition</i>)	8
2.4 <i>Framework Flutter</i>	9
2.5 TensorFlow Lite	10
2.6 Google ML Kit	11
2.7 <i>Perfomance Measuremet</i>	11
2.8 FaceNet	12
2.9 <i>Euclidean Distance</i>	13
2.10 <i>Threshold</i>	14
2.11 CNN	15
2.12 SSD	15
2.13 SQLite	16
2.14 Interpretasi Skor Kelayakan	16
2.15 Tinjauan Pustaka	17
BAB III PELAKSANAAN MAGANG	20
3.1 Manajemen Proyek	20
3.2 Pelaksanaan Proyek	21
3.3 Hasil Proyek	32
BAB IV REFLEKSI PELAKSANAAN MAGANG	45
4.1 Relevansi Akademik	45
4.2 Pembelajaran Magang	46
BAB V PENUTUP	49
5.1 Kesimpulan	49
5.2 Saran	50
DAFTAR PUSTAKA	51
LAMPIRAN	54

DAFTAR TABEL

Tabel 2.1 Interpretasi Skor Kelayakan.....	17
Tabel 2.2 Tinjauan Pustaka.....	18
Tabel 3.1 Spesifikasi Sistem Presensi Karyawan.....	22
Tabel 3. 2 Contoh Pengujian.....	40
Tabel 3. 3 Pengujian <i>Face Recognition</i>	43

DAFTAR GAMBAR

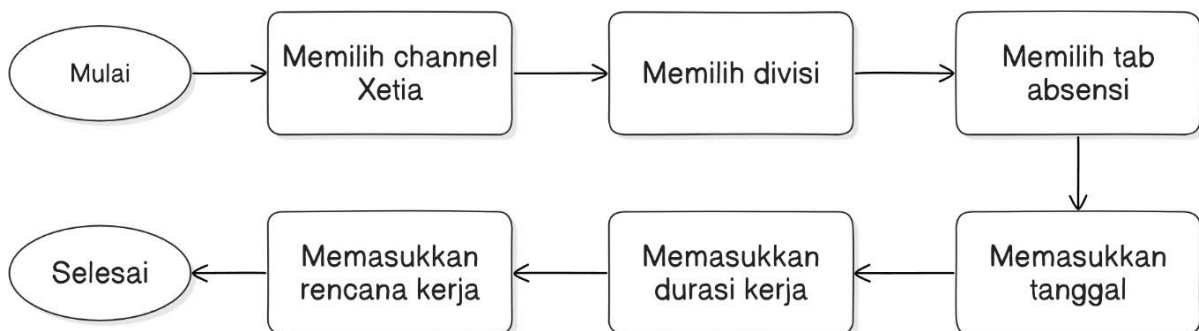
Gambar 1.1 <i>Flowchart</i> Presensi pada Discord.....	1
Gambar 2.1 Hasil Deteksi Wajah.....	7
Gambar 2.2 Tahapan <i>Face Recognition</i>	8
Gambar 2.3 Contoh Vektor Embedding	13
Gambar 3.1 Tahapan Metode <i>Prototyping</i>	20
Gambar 3.2 <i>Flowchart</i> Sistem Presensi Masuk	23
Gambar 3.3 <i>Flowchart</i> Deteksi Wajah.....	24
Gambar 3.4 <i>Flowchart Pre-processing</i>	24
Gambar 3.5 <i>Flowchart</i> Sistem Presensi Keluar	25
Gambar 3.6 Deteksi Wajah.....	26
Gambar 3.7 <i>Flowchart</i> Registrasi Wajah	28
Gambar 3.8 Pendaftaran Wajah	29
Gambar 3.9 <i>Flowchart</i> Pengenalan Wajah	29
Gambar 3.10 Prototipe Sistem Presensi Karyawan (1)	33
Gambar 3.11 Prototipe Sistem Presensi Karyawan (2)	33
Gambar 3.12 Tampilan <i>Face Registration</i>	34
Gambar 3.13 Pengambilan Gambar	34
Gambar 3.14 Tampilan <i>Face Recognition</i>	35
Gambar 3.15 Peringatan Lebih dari Satu.....	35
Gambar 3.16 Peringatan Tidak Terdeteksi	36
Gambar 3.16 Peringatan Wajah Berbeda.....	36
s Gambar 3.18 Tampilan <i>Clock In</i>	37
Gambar 3.19 Tampilan <i>Clock Out</i>	38
Gambar 3.20 Tampilan <i>Daily Standup</i>	39

BAB I PENDAHULUAN

1.1 Latar Belakang

PT Indo Xetia Internasional merupakan perusahaan teknologi yang berfokus pada pengembangan sistem informasi untuk berbagai industri. Perusahaan ini beroperasi di dua lokasi yakni Balikpapan, Kalimantan Timur dan Osaka, Jepang. Pada awalnya PT Indo Xetia Internasional mengembangkan platform makanan halal di Jepang sebagai bentuk dukungan kepada komunitas muslim dan dikenal sebagai Hararu. Namun, seiring waktu PT Indo Xetia Internasional juga berekspansi ke bisnis lain seperti NFT *marketplace* atau dikenal sebagai Harti.

PT Indo Xetia Internasional menerapkan sistem kerja yang memungkinkan karyawannya untuk bekerja dari rumah (*work from home*) sehingga mengharuskan perusahaan untuk memiliki sistem presensi yang dapat memvalidasi kehadiran karyawan secara akurat. Presensi yang akurat juga dapat memastikan bahwa karyawan selalu siap untuk tugas-tugas mendadak yang mungkin terjadi. Saat ini, karyawan menggunakan aplikasi Discord untuk melakukan presensi dengan proses bisnis seperti pada Gambar 1.1.



Gambar 1.1 *Flowchart* Presensi pada Discord

Namun, terdapat kelemahan dalam penggunaan aplikasi Discord yang berpotensi bagi seseorang yang bukan karyawan dapat melakukan presensi atas nama karyawan yang sebenarnya tidak hadir. Sejalan dengan kebutuhan akan sistem presensi yang lebih akurat maka PT Indo Xetia Internasional berupaya untuk mengembangkan sistem presensi baru. Teknologi *face recognition* dapat menjadi solusi karena kemampuannya dalam memverifikasi kehadiran secara akurat berdasarkan data wajah karyawan sehingga hanya pemilik akun yang dapat melakukan presensi dan mengurangi kesalahan identifikasi (Evelyn et al., 2022).

Penggunaan *framework* Flutter dalam pengembangan sistem presensi dipilih karena bersifat *open-source* artinya dapat diakses dan dimodifikasi secara gratis sehingga memungkinkan pertumbuhan ekosistem yang kuat. Keunggulan Flutter dalam *single codebase* memberikan efisiensi dalam menciptakan aplikasi untuk iOS dan Android (Tjandra & Chandra, 2020). Teknologi *face recognition* yang dibantu oleh *library* Tensorflow Lite dan Google ML Kit dalam Flutter meningkatkan akurasi dan responsivitas sistem presensi. Penggunaan SQLite untuk menyimpan data wajah karyawan secara lokal dapat menjaga privasi dikarenakan hanya tersimpan di perangkat karyawan sehingga tidak perlu dipindahkan atau disimpan di server eksternal (Setiyadi & Harihayati, 2015).

Pengembangan sistem presensi karyawan menggunakan model FaceNet yang dikembangkan oleh Google dan terkenal akan tingkat akurasi yang tinggi dalam pengenalan wajah. FaceNet bekerja dengan menghasilkan representasi vektor wajah melalui proses *embedding* sehingga wajah dari gambar diubah menjadi vektor fitur berdimensi tetap. Proses mengukur kesamaan antara vektor-vektor wajah yang dihasilkan menggunakan *euclidean distance* dikarenakan sifatnya yang sederhana dan efektif dalam menghitung jarak antara dua vektor di ruang berdimensi tinggi serta didukung oleh *library* dari *framework* Flutter. FaceNet memungkinkan sistem untuk mengidentifikasi dan membedakan wajah dengan tingkat keakuratan yang tinggi sehingga menjadi salah satu model terbaik dalam teknologi pengenalan wajah (Sugeng & Barus, 2023).

Pada sistem presensi karyawan tidak dilakukan pelatihan model secara mandiri karena menggunakan *library* yang disediakan oleh Flutter untuk menjankan model FaceNet yang telah dilatih sebelumnya sehingga sistem dapat fokus pada proses pencocokan dan identifikasi wajah. Penggunaan FaceNet pada sistem ini didasari oleh kompatibilitas dengan *library* TensorFlow Lite yang digunakan pada *framework* Flutter sehingga memungkinkan implementasi model *machine learning* FaceNet dengan efisiensi tinggi dan sistem dapat dijalankan pada perangkat dengan keterbatasan sumber daya namun tetap mampu menghasilkan performa pengenalan wajah yang akurat. Model FaceNet memerlukan *hardware* berkualitas tinggi untuk beroperasi secara optimal sehingga dapat menjadi hambatan jika terdapat keterbatasan pada *hardware* (Cahyono F, 2020).

Proses pengambilan gambar pada sistem presensi karyawan menggunakan kamera perangkat secara *real-time* sehingga karyawan hanya dapat mengambil gambar dalam kondisi nyata dan tidak dapat menggunakan gambar yang tersimpan di galeri perangkat. Gambar hanya

dapat diambil satu kali pada proses pendaftaran wajah dan presensi sehingga dapat memastikan keaslian dan keakuratan proses pengenalan wajah.

Penelitian ini dilakukan untuk mengetahui akurasi sistem pengenalan wajah dalam berbagai kondisi penggunaan sehari-hari seperti pada cahaya terang, cahaya redup, posisi miring, posisi lurus, ekspresi tersenyum, ekspresi datar, jarak 30cm, jarak 60 cm, penggunaan topi, dan kacamata. Penelitian ini akan mengevaluasi teknologi yang digunakan pada pengembangan *face recognition* dalam mendeteksi dan mengenali wajah. Hasil penelitian diharapkan dapat memberikan gambaran yang jelas tentang efektivitas sistem dalam dunia nyata dan mengidentifikasi area yang memerlukan perbaikan untuk meningkatkan akurasi pada pengenalan wajah.

1.2 Ruang Lingkup

Pengembangan sistem presensi karyawan dilaksanakan selama enam bulan dimulai dari September 2023 hingga Maret 2024. Kontribusi penulis akan mencakup beberapa aspek penting dalam pengembangan sistem sebagai berikut:

- a. Perancangan prototipe sistem presensi karyawan.
- b. Pengembangan fitur *face registration* dengan karyawan hanya dapat mendaftarkan wajah satu kali.
- c. Pengembangan fitur *face recognition* dengan model FaceNet dan dikembangkan dengan bantuan *library* yang disediakan oleh Flutter.
- d. Pengembangan fitur *clock in* agar karyawan dapat memasukkan rencana kerja secara detail.
- e. Pengembangan fitur *clock out* agar karyawan dapat melakukan presensi keluar dengan memasukkan laporan kerja.
- f. Pengembangan fitur *daily standup* agar karyawan dapat memasukkan pekerjaan yang telah dilakukan pada hari sebelumnya, hari ini, rencana besok, dan kendala selama pengerjaan.
- g. Pengujian sistem pada proses pengenalan wajah dalam berbagai kondisi penggunaan sehari-hari seperti pada cahaya terang, cahaya redup, posisi miring, posisi lurus, ekspresi tersenyum, ekspresi datar, jarak 30cm, jarak 60 cm, penggunaan topi, dan kacamata dengan 5 iterasi percobaan.

1.3 Tujuan

Laporan ini ditulis dengan tujuan untuk memberikan pemahaman mendalam tentang proses pengembangan sistem presensi karyawan dengan menggunakan teknologi *face*

recognition dalam mengevaluasi wajah karyawan. Laporan ini akan menjelaskan langkah-langkah yang diambil dalam pengembangan sistem serta manfaat dan implikasi dari implementasi teknologi yang digunakan dalam konteks sistem presensi. Hasilnya diharapkan dapat menjadi acuan bagi perusahaan yang sedang mempertimbangkan penggunaan sistem presensi untuk karyawan yang bekerja secara *work from home* dengan memanfaatkan teknologi *face recognition* yang dikembangkan menggunakan model FaceNet dan didukung *library* yang disediakan oleh *framework* Flutter.

1.4 Manfaat

Manfaat dari pengembangan sistem presensi karyawan dengan *face recognition* menggunakan model FaceNet adalah sebagai berikut:

- a. Meningkatkan keakuratan sistem presensi karyawan karena hanya pemilik akun yang dapat melakukan presensi sehingga mengurangi risiko penyalahgunaan.
- b. Penggunaan *library* Flutter menjadikan pengembangan dan implementasi *face recognition* lebih efisien karena Flutter menyediakan berbagai dukungan yang mempermudah proses integrasi *machine learning*.
- c. Menjaga privasi data wajah karyawan karena hanya disimpan secara lokal dengan SQLite sehingga tidak perlu disimpan atau dipindahkan ke server eksternal.

1.5 Sistematika Penulisan

Sistematika penulisan laporan disusun dengan tujuan mempermudah pembaca dalam memahami isi laporan secara menyeluruh. Berikut adalah susunan sistematika penulisan laporan yang digunakan:

- a. BAB I: Pendahuluan

Bab ini membahas latar belakang, ruang lingkung, tujuan, manfaat, dan sistematika penulisan.

- b. BAB II: Landasan Teori dan Tinjauan Pustaka

Bab ini membahas teori – teori yang menjadi dasar pada pengembangan sistem presensi karyawan dengan *face recognition* menggunakan model FaceNet.

- c. BAB III: Pelaksanaan Magang

Bab ini membahas implementasi dari proses kegiatan magang.

- d. BAB IV: Refleksi Pelaksanaan Magang

Bab ini membahas refleksi dan hasil yang diperoleh selama magang pada proyek pengembangan sistem presensi karyawan dengan *face recognition* menggunakan model FaceNet.

e. BAB V: Penutup

Bab ini membahas kesimpulan dan saran dari materi yang telah dibahas pada bab sebelumnya.

BAB II

LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1 Sistem Presensi

Sistem presensi adalah sistem yang mencatat kehadiran individu dalam suatu aktivitas terutama dalam konteks institusi atau organisasi sehingga melibatkan pengumpulan, pengorganisasian, dan penyimpanan data kehadiran individu yang memudahkan akses bagi pihak yang berkepentingan (Wardoyo et al., 2016). Informasi presensi pada institusi atau organisasi mencakup waktu kehadiran, durasi kehadiran, dan catatan terkait kehadiran apabila diperlukan.

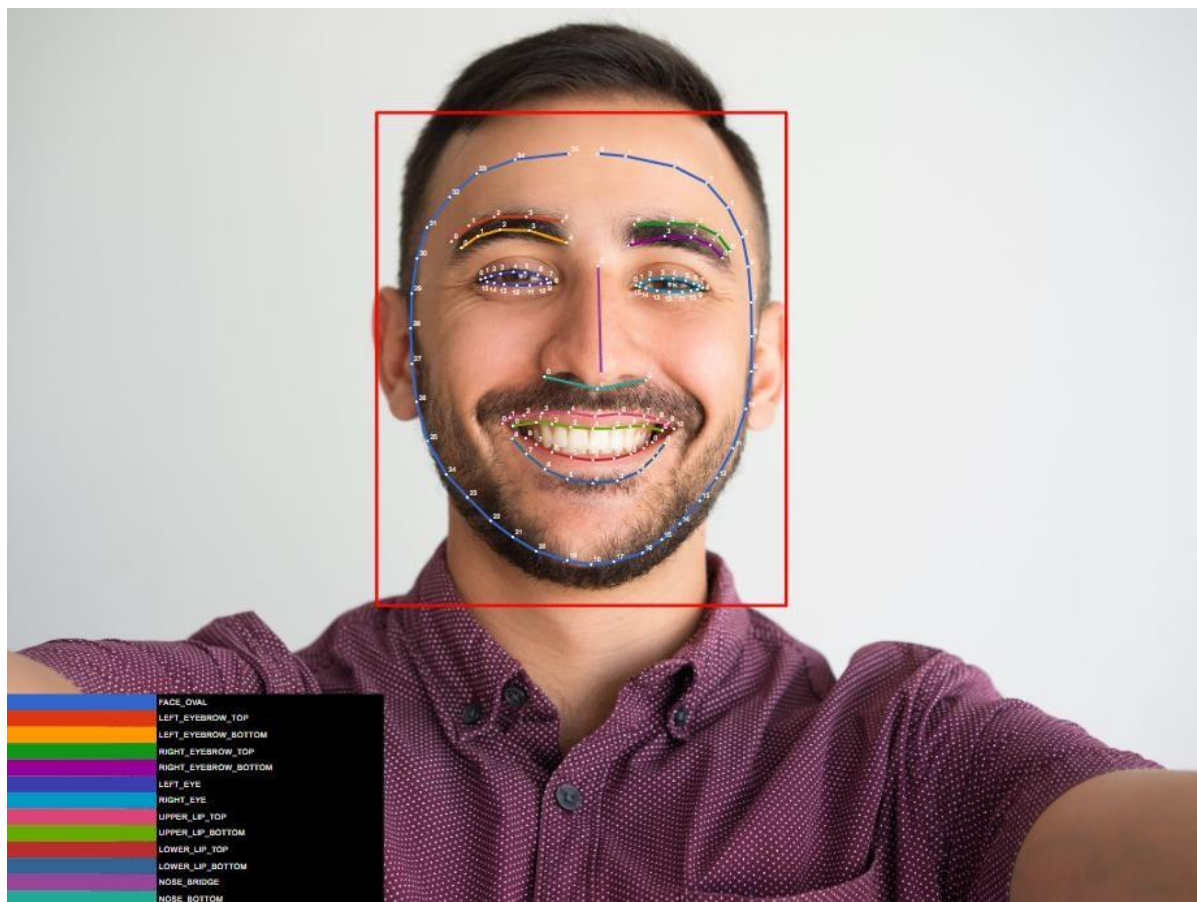
Pada awalnya, presensi dilakukan secara manual dengan kertas namun seiring berjalan waktu menjadi sistem yang lebih canggih dengan penggunaan program komputer, *finger print*, dan *scan* mata. Saat ini, banyak perusahaan beralih ke teknologi *gadget* seperti Android untuk memudahkan proses presensi karyawan sehingga evolusi sistem presensi terus berlanjut untuk memberikan solusi yang lebih efisien dan akurat dalam manajemen kehadiran karyawan (Husain et al., 2017).

Sistem presensi dapat membantu dalam mencatat kehadiran secara akurat dan mengurangi risiko kesalahan dalam proses pencacatan. Informasi yang diperoleh dari sistem presensi dapat digunakan untuk mengoptimalkan proses penggajian dengan menghitung upah berdasarkan jam kerja yang telah dilakukan (Setyadi & Sundari, 2022). Evaluasi kinerja karyawan menjadi lebih objektif dengan data kehadiran yang akurat dari sistem presensi sehingga karyawan mendapatkan umpan balik yang tepat dan konstruktif. Hal ini dapat meningkatkan motivasi karyawan dan produktivitas secara keseluruhan (Samodra & Noermijati, 2016).

Pemantauan kehadiran yang teratur dapat membantu organisasi dalam mengidentifikasi karyawan yang memiliki tingkat kehadiran yang baik dan karyawan yang perlu diberikan penghargaan atau insentif atas kinerja kerja yang konsisten dan dengan mengetahui pola kehadiran maka karyawan berhak mendapatkan pengakuan yang pantas karena telah menunjukkan dedikasi dan komitmen yang tinggi terhadap pekerjaan sehingga dapat menjadi motivasi tambahan bagi karyawan untuk terus memberikan kinerja terbaik dan meningkatkan produktivitas secara keseluruhan di dalam organisasi (Prabu & Wijayanti, 2016).

2.2 Deteksi Wajah (*Face Detection*)

Face detection adalah salah satu tahap awal sebelum dilakukan proses pengenalan wajah. Deteksi wajah didasarkan pada identifikasi dan penentuan lokasi citra wajah manusia dalam gambar terlepas dari ukuran, posisi, dan kondisi wajah tersebut (Miftahuddin et al., 2020). Deteksi wajah dapat dipandang sebagai masalah klasifikasi pola dan inputnya adalah citra masukan serta outputnya berupa label kelas dari citra tersebut. Pada hal ini terdapat dua label yaitu wajah dan non-wajah. Teknik-teknik pengenalan wajah yang umum digunakan untuk mengasumsikan bahwa data wajah yang tersedia memiliki ukuran yang sama dan latar belakang yang seragam. Namun, di dunia nyata asumsi ini tidak berlaku karena wajah dapat muncul dengan berbagai ukuran, posisi, dan latar belakang yang bervariasi. Hasil deteksi wajah dapat dilihat pada Gambar 2.1.

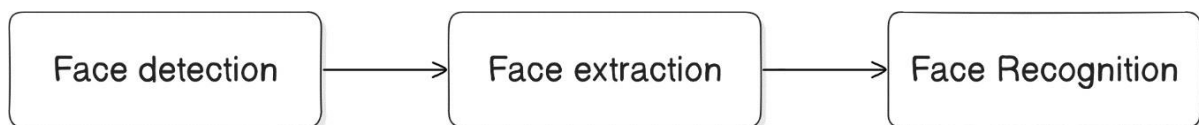


Gambar 2.1 Hasil Deteksi Wajah

(sumber: <https://developers.google.com/ml-kit/vision/face-detection>)

2.3 Pengenalan Wajah (*Face Recognition*)

Face recognition adalah sebuah teknologi biometrik untuk mengidentifikasi atau mengkonfirmasi identitas seseorang berdasarkan ciri-ciri unik pada wajah (Giovanni et al., 2023). Teknologi ini memiliki tingkat kesalahan rendah dan tingkat akurasi tertinggi dibandingkan dengan jenis biometrik lainnya sehingga penggunaan wajah sebagai metode identifikasi biometrik memberikan hasil yang sangat akurat dalam proses otentikasi identitas (Bhattacharyya et al., 2009). *Face recognition* beroperasi dengan menganalisis serta membandingkan pola wajah yang diambil dari gambar atau video dengan data wajah yang telah tersimpan pada basisdata. Proses *face recognition* melibatkan beberapa tahapan seperti deteksi wajah (*face detection*), ekstraksi fitur wajah (*face extraction*), dan pengenalan wajah (*face recognition*) seperti pada Gambar 2.2 (Chihaoui et al., 2016).



Gambar 2.2 Tahapan *Face Recognition*

(sumber: <https://www.mdpi.com/2073-431X/5/4/21>)

Face detection merupakan tahap awal dalam proses pengenalan wajah yang bertujuan untuk mengidentifikasi dan menandai lokasi wajah pada gambar atau video. Tahap ini bertujuan untuk menemukan dan menyusun data wajah dengan berbagai ukuran, posisi, dan latar belakang. Wajah manusia tingkat variasi yang tinggi dalam hal perbedaan antar individu maupun faktor lingkungan seperti pencahayaan sehingga diperlukan penggunaan algoritma yang kompleks untuk mengintegrasikan representasi wajah manusia ke dalam sistem komputer vision (Muhaimin, 2013).

Setelah wajah berhasil terdeteksi maka tahap selanjutnya adalah ekstraksi fitur wajah. Pada tahap ini, fitur-fitur penting dari wajah seperti jarak antara mata, lebar hidung atau bentuk mulut diidentifikasi. Proses ekstraksi fitur penting karena memungkinkan sistem untuk mengidentifikasi wajah secara unik. Tetapi, karena perbedaan individu dalam ciri-ciri wajah sehingga dapat menjadi tantangan (Sejati & Mardhiyyah, 2021).

Face recognition melibatkan perbandingan fitur-fitur yang diekstraksi dari wajah yang diambil dengan data wajah yang telah tersimpan. Jika terdapat kesamaan yang mencakupi antara fitur-fitur tersebut maka sistem akan mengidentifikasi wajah tersebut sebagai milik

individu yang sesuai. Proses ini dapat digunakan untuk autentikasi identitas seseorang atau untuk mengidentifikasi seseorang dalam situasi tertentu (Salamun & Wazir, 2016).

2.4 Framework Flutter

Framework adalah struktur yang dirancang untuk mempermudah pengembangan aplikasi. Penggunaan *framework* dapat membuat aplikasi yang dibangun menjadi lebih terstruktur dan terorganisir. *Framework* dikembangkan untuk membantu pengembang dalam mengelola kompleksitas pengembangan aplikasi dan meningkatkan efisiensi dalam membangun sistem (Haniefardy et al., 2019). *Framework* flutter merupakan sebuah *framework* yang dikembangkan oleh Google untuk mengembangkan aplikasi *mobile* dengan performa yang optimal. Flutter memungkinkan pengembang dapat membuat aplikasi pada platform iOS dan Android menggunakan satu basis kode. Flutter bersifat *open-source* atau tersedia untuk umum sehingga dapat digunakan dan dimodifikasi secara bebas oleh pengembang.

Flutter menggunakan bahasa pemrograman Dart yang dikembangkan oleh Google sebagai pengganti JavaScript. Salah satu fitur Dart adalah penggunaan *static typing* yakni variabel harus dideklarasikan sebelum digunakan. Dart dapat digunakan di berbagai platform seperti *website*, *desktop*, dan *mobile*. Pada pengembangan *website*, Dart menggunakan *dart2js* untuk mengonversi kode Dart menjadi JavaScript agar dapat dijalankan di browser. Sedangkan untuk pengembangan aplikasi *desktop*, Dart menggunakan *dart2aot* untuk mengonversi kode Dart menjadi kode mesin yang dapat dijalankan langsung. Dart juga digunakan bersama dengan Flutter untuk pengembangan aplikasi *mobile*. Syntax Dart relatif mudah dipelajari karena mirip dengan syntax bahasa pemrograman lain seperti JavaScript dan Java.

Flutter memiliki fitur *hot reload* dan *hot restart* saat melakukan debug. *Hot reload* memungkinkan pengembang untuk mengubah kode aplikasi secara langsung tanpa perlu menghentikan atau *re-start* aplikasi sehingga perubahan yang dilakukan dapat segera dilihat pada aplikasi yang sedang berjalan dan mempercepat proses pengembangan. *Hot restart* digunakan untuk *reset state* aplikasi Flutter tanpa perlu menutup dan membuka kembali aplikasi secara manual. *Hot restart* akan mengembalikan aplikasi ke kondisi awal saat pertama kali dijalankan sehingga berguna saat pengembang ingin menguji perubahan yang mempengaruhi *state* aplikasi atau memulai ulang aplikasi dari awal untuk menguji fitur baru (Tjandra & Chandra, 2020).

2.5 TensorFlow Lite

TensorFlow Lite adalah versi TensorFlow yang dirancang khusus pada perangkat *mobile* dan perangkat tertanam sehingga memungkinkan model *machine learning* berjalan pada perangkat *mobile* walaupun dengan keterbatasan daya pemrosesan yang rendah. TensorFlow Lite dapat melakukan klasifikasi atau tugas lain yang membutuhkan pemrosesan data kompleks tanpa harus terus-menerus terhubung ke server. TensorFlow Lite dapat diterapkan pada berbagai sistem operasi *mobile* termasuk Android dan iOS sehingga pengembang dapat menggunakan model *machine learning* yang dioptimalkan untuk perangkat *mobile* (Roslidar et al., 2022).

TensorFlow Lite mencakup *software* yang memungkinkan eksekusi model yang telah ada sebelumnya dan menyediakan berbagai alat untuk mempersiapkan model agar dapat digunakan pada perangkat seluler tertanam. Namun, TensorFlow Lite tidak dirancang untuk melatih model baru sehingga pengembang dapat melatih model pada mesin dengan daya komputasi yang lebih besar dan mengonversi model tersebut ke format TensorFlow Lite untuk digunakan pada perangkat seluler (Fachriyan & Dharmayanti, n.d.).

TensorFlow Lite mendukung berbagai jenis model *machine learning* termasuk pengenalan gambar, suara, dan model kompleks seperti deteksi objek atau segmentasi gambar sehingga menjadi pilihan dalam mengimplementasikan fungsi *machine learning* pada aplikasi *mobile* tanpa harus memikirkan masalah kinerja atau keterbatasan perangkat (Ferdiawan & Hartono, 2022). TensorFlow Lite memiliki kemampuan dalam mengoptimalkan model agar berjalan lebih efisien pada perangkat *mobile* dengan menggunakan teknik kompresi dan optimasi khusus yang memungkinkan model berjalan dengan kinerja lebih baik tanpa mengorbankan akurasi.

TensorFlow Lite memiliki beberapa kelemahan seperti ukuran model bisa besar tergantung pada kompleksitas model dan jenis data yang digunakan sehingga mempengaruhi kinerja aplikasi dan memerlukan pemikiran matang dalam hal optimasi dan manajemen memori. Pengembang juga perlu mempertimbangkan ketersediaan dan dukungan *hardware* pada perangkat target karena tidak semua perangkat memiliki kemampuan dalam menjalankan model dengan kinerja optimal sehingga perlu melakukan pengujian untuk memastikan model berjalan dengan baik pada perangkat yang dituju.

2.6 Google ML Kit

Google ML Kit adalah sebuah SDK *mobile* yang disediakan oleh Google untuk mengintegrasikan kemampuan *machine learning* ke dalam aplikasi Android dan iOS dengan mudah. Dengan menggunakan Google ML Kit maka pengembang tidak perlu memiliki pengetahuan mendalam tentang *machine learning* dalam mengimplementasikan fitur-fitur *machine learning* ke dalam aplikasi. Google ML Kit melakukan tugas-tugas kompleks seperti pengenalan teks dan deteksi objek dengan mudah (Aditia Muclis & Somantri, 2021).

Google ML Kit memiliki dukungan yang kuat untuk berbagai jenis model *machine learning* termasuk model yang telah dilatih sebelumnya oleh Google serta model yang telah dibuat oleh pengembang sehingga pengembang dapat memilih model yang sesuai dengan kebutuhan aplikasi. Google ML Kit dapat membuat aplikasi yang lebih cerdas dan responsive tanpa perlu menginvestasikan banyak waktu dan sumber daya dalam mengembangkan dan melatih model *machine learning* sehingga dapat menjadi pilihan untuk meningkatkan aplikasi.

Google ML Kit menawarkan beberapa kemampuan seperti berikut:

1. Mengenali teks seperti pengambilan data dari gambar atau penerjemahan teks dalam gambar ke dalam bahasa yang berbeda.
2. Mengenali *landmark* sehingga dapat digunakan pada aplikasi pariwisata atau navigasi untuk memberikan informasi tambahan kepada pengguna tentang lokasi yang akan dikunjungi.
3. Melakukan pengenalan wajah seperti aplikasi yang dapat mengenali dan memverifikasi wajah pengguna untuk pengamanan.
4. Pemindaian kode batang seperti aplikasi pembayaran digital yang dapat memindai dan mengenali kode batang.
5. Memberikan label pada gambar untuk aplikasi yang dapat memberikan deskripsi atau informasi tambahan tentang objek dalam gambar sehingga meningkatkan pengalaman pengguna dengan memberikan konteks tambahan tentang objek yang terlihat pada gambar (Fachriyan & Dharmayanti, n.d.).

2.7 Performance Measuremet

Pengukuran efektivitas dihitung berdasarkan kesuksesan dan kegagalan dalam menyelesaikan tugas yang diberikan. Pada sistem pengenalan wajah dapat dilakukan dengan menilai tingkat keberhasilan dan tingkat kegagalan dalam mengenali wajah. Tingkat keberhasilan adalah metrik yang digunakan untuk menilai keberhasilan sebuah sistem dalam

menyelesaikan tugas yang diberikan (Puspita sari et al., 2023). Pada konteks pengenalan wajah maka tingkat keberhasilan berperan dalam mengukur proporsi dari total upaya yang berhasil dalam mengenali wajah pengguna dibandingkan dengan total upaya keseluruhan. Tingkat keberhasilan dapat ditulis seperti pada persamaan (2.1).

$$\text{Tingkat Keberhasilan} = \frac{\text{Jumlah Berhasil}}{\text{Jumlah Berhasil} + \text{Jumlah Gagal}} \quad (2.1)$$

Metrik ini memberikan gambaran kuantitatif mengenai kemampuan sistem dalam melakukan identifikasi wajah secara akurat. Tingkat keberhasilan yang tinggi menunjukkan bahwa sistem bekerja dengan baik dalam mengenali wajah pengguna sehingga menunjukkan bahwa algoritma dan model yang digunakan telah dioptimalkan dengan baik. Sebaliknya, tingkat keberhasilan yang rendah mengindikasikan bahwa sistem perlu diperbaiki atau dioptimalkan lebih lanjut.

Tingkat kegagalan digunakan untuk menilai kegagalan sebuah sistem dalam menyelesaikan tugas sehingga tingkat kegagalan berperan dalam mengukur proporsi dari total upaya yang gagal dalam mengenali wajah pengguna dibandingkan dengan total upaya keseluruhan (Sulistiya et al., 2016). Tingkat kegagalan dapat ditulis seperti pada persamaan (2.2).

$$\text{Tingkat Kegagalan} = \frac{\text{Jumlah Gagal}}{\text{Jumlah Berhasil} + \text{Jumlah Gagal}} \quad (2.2)$$

2.8 FaceNet

FaceNet adalah model pengenalan wajah yang dikembangkan oleh Google dan terkenal karena tingkat akurasi yang tinggi. Model ini memungkinkan sistem untuk mengidentifikasi dan membedakan wajah dengan tingkat keakuratan yang tinggi sehingga menjadi salah satu model terbaik dalam teknologi pengenalan wajah. Penggunaan FaceNet pada aplikasi dapat secara akurat mengenali wajah dalam gambar atau video.

Keunggulan utama dari FaceNet adalah kemampuannya dalam menghasilkan representasi numerik dari wajah yang dapat diukur dan dibandingkan dengan wajah lainnya dengan tingkat keakuratan yang tinggi. Proses dimulai melalui penggunaan teknik *deep learning* sehingga memungkinkan model untuk belajar dan mengenali pola-pola kompleks dalam wajah manusia. Hasilnya adalah FaceNet dapat menghasilkan representasi yang sangat

diskriminatif untuk setiap wajah sehingga sistem dapat membedakan antara wajah yang mirip secara visual (Sugeng & Barus, 2023).

Proses pengolahan gambar wajah pada model FaceNet melibatkan pengenalan wajah, pemotongan wajah yang terdeteksi, dan penyesuaian ukuran gambar wajah untuk menghasilkan gambar wajah yang lebih kecil. Data yang telah diolah kemudian diekstraksi ke dalam file 128-bit menggunakan model FaceNet dan menghasilkan vektor embedding seperti pada Gambar 2.3.

```
I/flutter (12206): Vector Embedding: [-0.020669879391789436, 0.003616209141910076,
-0.0027272007428109646, -0.025339990854263306, -0.004208545666188002, -0.08892197906970978,
0.015713892877101898, -0.17072
339355945587, -0.1532922089099884, 0.06756497174501419, 0.018221523612737656, 0.006279035471379757,
-0.011196046136319637, 0.026348702609539032, -0.0018011038191616535, -0.20483002066612244,
0.013345984742
045403, 0.011220746673643589, -0.0038757482543587685, -0.01838851347565651, 0.10156667232513428,
0.1609555333852768, 0.03281337767839432, -0.005701384972780943, 0.060259491205215454,
0.04344313219189644, -
0.003606468439102173, -0.06299646943807602, 0.07545378804206848, 0.05427387356758118,
0.005009256303310394, -0.10248428583145142, -0.08256267756223679, -0.007815069518983364,
-0.15152916312217712, 0.048412
7476811409, 0.17954351007938385, -0.04190296679735184, -0.0017287315567955375, -0.1295468509197235,
-0.010677667334675789, -0.0014984308509156108, -0.0062432968989014626, 0.007033342029899359,
0.0007754507823847234, -0.034
```

Gambar 2.3 Contoh Vektor Embedding

Salah satu langkah penting dalam pengenalan wajah dengan FaceNet adalah membandingkan embedding wajah yang dihasilkan menggunakan perhitungan euclidean *distance*. Euclidean *distance* digunakan untuk mengukur seberapa mirip dua embedding wajah yang berbeda sehingga semakin kecil jaraknya maka semakin mirip wajah-wajah tersebut dianggap (Efanntyo & Mitra, 2021).

Penggunaan FaceNet pada sistem ini didasari oleh kompatibilitas dengan *library* TensorFlow Lite yang digunakan pada *framework* Flutter sehingga memungkinkan implementasi model *machine learning* FaceNet dengan efisiensi tinggi dan sistem dapat dijalankan pada perangkat dengan keterbatasan sumber daya namun tetap mampu menghasilkan performa pengenalan wajah yang akurat.

2.9 Euclidean Distance

Euclidean *distance* adalah metode perhitungan jarak antara dua titik dalam ruang euclidean. Ruang euclidean diperkenalkan oleh Euclid, seorang matematikawan Yunani sekitar tahun 300 SM untuk mempelajari hubungan antara sudut dan jarak (Miftahuddin et al., 2020). Pada matematika euclidean *distance* berkaitan erat dengan Terorema Pythagoras untuk

mengukur dua titik dalam satu dimensi. Perhitungan jarak euclidean ditulis seperti pada persamaan (2.3).

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.3)$$

Keterangan :

d = Jarak

x_1 = Koordinat latitude 1

x_2 = Koordinat latitude 2

y_1 = Koordinat longtitude 1

y_2 = Koordinat longtitude 2

Penggunaan euclidean *distance* pada pengenalan wajah didasarkan pada kebutuhan untuk mengukur kesamaan antara dua vektor embedding yang dihasilkan oleh model FaceNet. Vektor embedding merupakan representasi numerik dari wajah yang telah diekstraksi oleh model. Euclidean *distance* digunakan untuk mengukur jarak antara dua vektor embedding sehingga jarak yang lebih kecil menunjukkan kesamaan yang lebih tinggi antara dua wajah.

Euclidean *distance* dipilih karena kesesuaian dengan *library* yang digunakan pada pengembangan aplikasi Flutter. Pada implementasi model FaceNet dengan TensorFlow Lite pada Flutter sehingga perhitungan euclidean *distance* dilakukan untuk membandingkan vektor embedding wajah secara langsung. *Library* TensorFlow Lite digunakan pada Flutter untuk mendukung operasi matematis dasar seperti euclidean *distance* sehingga menjadikan metode yang efisien dan terintegrasi pada *framework* pengembangan aplikasi.

2.10 Threshold

Threshold adalah batas nilai yang digunakan sebagai acuan dalam pengambilan keputusan (Setiawan et al., 2024). Pada konteks pengenalan wajah *threshold* berperan dalam menentukan seberapa besar kemiripan antara wajah yang telah terdaftar. Jika nilai kemiripan wajah sama dengan atau di bawah *threshold* maka wajah tersebut dikenali. Sebaliknya, jika nilai kemiripan melebihi *threshold* maka sistem menilai wajah tersebut tidak dikenali.

Pengaturan *threshold* yang tepat sangat penting untuk memastikan akurasi pada proses pengenalan wajah dan menghindari kesalahan identifikasi. *Threshold* yang terlalu rendah dapat meningkatkan risiko *false positives* yaitu sistem salah mengenali wajah yang sebenarnya tidak ada pada basisdata. Apabila *threshold* terlalu tinggi dapat menyebabkan *false negatives* yaitu wajah yang seharusnya dikenali tidak terdeteksi oleh sistem.

Penelitian yang dilakukan oleh Setiawan et al. (2024) menunjukkan bahwa pengujian pada aplikasi presensi *mobile* menggunakan metode *deep learning* dan menetapkan *threshold* = 1.0 (Setiawan et al., 2024). Peneliti Gao et al. (2022) mengkaji performa sistem pengenalan wajah dengan berbagai *threshold* yang bervariasi dari 1.0 hingga 1.18 (Setiawan et al., 2024). Berdasarkan penelitian tersebut maka sistem menggunakan *threshold* 1.0 karena merupakan nilai awal yang ideal pada banyak kasus.

2.11 CNN

Convolutional Neural Network (CNN) adalah algoritma *deep learning* yang dirancang khusus untuk menganalisis data visual seperti gambar dan dapat mengenali fitur-fitur penting dalam gambar melalui pembelajaran bobot dan bias dari berbagai bagian gambar sehingga memungkinkan deteksi objek yang berbeda. Arsitektur CNN merupakan salah satu jenis *neural network* yang umum digunakan pada pengolahan citra digital.

CNN memiliki struktur unik namun prinsip dasar tidak terlalu berbeda dengan *neural network* secara umum. CNN terdiri dari dua lapisan arsitektur utama yang memiliki peran penting dalam prosesnya yaitu *feature learning layer* dan *classification layer*. *Feature learning layer* bertugas untuk mengekstraksi fitur-fitur penting dari gambar sedangkan *classification layer* digunakan untuk mengklasifikasikan objek berdasarkan fitur-fitur yang telah diekstrak sebelumnya (Azmi et al., 2023).

CNN digunakan dalam berbagai aplikasi termasuk dalam bidang pengenalan wajah seperti deteksi objek dalam gambar, pengenalan tulisan tangan, dan lain-lain. Keunggulan CNN adalah kemampuan untuk mengekstraksi fitur-fitur secara otomatis dari gambar tanpa memerlukan proses ekstraksi fitur secara manual yang memakan waktu (Alwanda et al., 2020).

2.12 SSD

Single Shot Multibox Detector (SSD) adalah metode deteksi objek dengan menggunakan arsitektur CNN. SSD dirancang untuk melakukan deteksi objek dalam gambar dengan menghasilkan *bounding box* dan label kelas objek secara langsung melalui satu kali *feedforward pass* yaitu proses ketika data dimasukkan ke dalam *neural network* dan dijalankan melalui lapisan-lapisan jaringan dari *input* hingga *output* tanpa kembali ke lapisan sebelumnya. Metode ini menggabungkan kecepatan dan akurasi yang tinggi sehingga dapat menjadi pilihan dalam aplikasi *real-time*.

SSD bekerja dengan cara mendeteksi objek pada berbagai skala dalam satu jaringan untuk menangani variasi ukuran dan aspek dari objek yang dideteksi. Struktur arsitektur SSD terdiri dari beberapa lapisan dan setiap lapisan bertugas untuk menghasilkan sejumlah prediksi *bounding box* dengan berbagai ukuran dan aspek rasio (Ravy Pramestya Hayu, 2018).

Proses deteksi objek pada SSD dilakukan secara *real-time* dengan mengambil citra dari kamera dan diolah menggunakan TensorFlow. Proses ini melibatkan tiga langkah utama sebagai berikut:

1. Menentukan kotak pembatas untuk mengukur seberapa dekat kotak terkait jaringan yang diprediksi dengan koordinat objek yang terdeteksi.
2. Mengklasifikasikan objek yang terdeteksi ke dalam kelas yang sesuai.
3. Mendapatkan nilai probabilitas dari objek yang terdeteksi (Fuady et al., 2020).

2.13 SQLite

SQLite adalah sebuah *library* yang menyediakan sistem manajemen basisdata *self-contained* sehingga sistem tidak memerlukan server atau instalasi yang rumit. Penggunaan SQLite tidak memerlukan konfigurasi tambahan dan sangat mudah untuk diimplementasikan dalam berbagai aplikasi. SQLite mendukung transaksi untuk menjalankan serangkaian perintah SQL sebagai satu kesatuan yang dapat di-*rollback* jika terjadi kesalahan.

SQLite memiliki ukuran yang kecil sehingga membantu aplikasi-aplikasi yang membutuhkan basisdata yang ringan dan efisien. Data disimpan dalam file tunggal yang dapat diakses langsung oleh aplikasi tanpa perlu melalui jaringan seperti basisdata server sehingga SQLite termasuk memiliki kinerja yang cepat. SQLite dapat bekerja dengan baik dalam lingkungan dengan sumber daya terbatas seperti perangkat *mobile* atau *embedded systems*. SQLite memiliki dokumentasi lengkap dan komunitas pengguna yang aktif untuk mendapatkan bantuan atau informasi tambahan jika diperlukan. Tetapi, SQLite kurang disarankan untuk aplikasi yang membutuhkan skala besar karena tidak dirancang untuk menangani beban kerja yang tinggi. SQLite tidak mendukung beberapa fitur yang biasanya tersedia dalam basisdata server seperti dukungan untuk prosedur tersimpan atau fungsi-fungsi yang kompleks (Setiyadi & Harihayati, 2015).

2.14 Interpretasi Skor Kelayakan

Kriteria kelayakan pada sistem dapat diidentifikasi dari data yang diperoleh berdasarkan bentuk persentase pada Tabel 2.1.

Tabel 2.1 Interpretasi Skor Kelayakan

Persentase(%)	Kriteria
0-20%	Sangat Tidak Layak
21-40%	Tidak Layak
41-60%	Cukup Layak
61-80%	Layak
81-100%	Sangat Layak

(sumber: Muhammad, R., Eka Murtinugraha, R., & Sittati Musalamah, 2020)

2.15 Tinjauan Pustaka

Penelitian terkait teknologi *face recognition* menggunakan model FaceNet telah dilakukan oleh peneliti Cahyono (2020) melakukan pengujian performa akurasi model FaceNet dalam mengenali wajah pegawai untuk keperluan presensi. Hasilnya menunjukkan bahwa kombinasi model FaceNet dan Support Vector Machine (SVM) mencapai akurasi sebesar 100% sedangkan model OpenFace hanya mencapai akurasi sebesar 93,3% (Cahyono F, 2020).

Peneliti Sugeng et al. (2023) melakukan pengujian keakuratan sistem dalam mengidentifikasi wajah seseorang meskipun terdapat objek atau aksesoris yang menutupi sebagian wajah pada pengecekan foto paspor. Pengujian ini menggunakan metode Deep Neural Network (DNN) dan FaceNet sebagai teknik pengenalan wajah. Hasilnya menunjukkan bahwa terdapat masalah pada pengenalan wajah dari samping, pengaruh cahaya dari webcam masih tergolong baik dengan mencapai 97,48% terhadap 6.107 dataset (Sugeng & Barus, 2023).

Peneliti Mantara et al. (2022) melakukan pengujian keakuratan sistem dalam mendeteksi keberadaan wajah secara *real-time* dengan variasi kemiringan, jarak, dan intensitas cahaya. Pengujian ini menggunakan metode Haar Cascade dan FaceNet. Hasilnya menunjukkan bahwa sistem terbukti efektif dan akurat dengan tingkat keberhasilan 80% dalam pengenalan wajah saat diuji menggunakan 6 dataset (Mantara et al., 2022).

Peneliti Fiqihilmi et al. (2023) melakukan pengujian ketepatan sistem pengenalan wajah menggunakan FaceNet dalam aplikasi presensi mahasiswa berbasis Android. Pengujian dilakukan dalam berbagai kondisi seperti wajah dalam posisi tegak menghadap kamera, menghadap kanan, dan menghadap kiri. Hasil pengujian menunjukkan bahwa sistem memiliki tingkat akurasi sebesar 94,44% (Lalu Fiqihilmi et al., 2023).

Peneliti Chrisyanto et al. (2022) melakukan pengujian keakuratan pengenalan wajah pada aplikasi Android untuk presensi kehadiran *real-time* menggunakan model FaceNet. Hasil pengujian menunjukkan bahwa sistem memiliki tingkat akurasi sebesar 100% dalam 15

percobaan (Christyanto et al., 2022). Berikut ringkasan beberapa penelitian terdahulu yang disajikan pada Tabel 2.2.

Tabel 2.2 Tinjauan Pustaka

No.	Peneliti	Judul	Tahun	Tujuan	Metode	Hasil
1.	Ferry Cahyono	Pengenalan Wajah Menggunakan Model FaceNet untuk Presensi Pegawai	2020	Mengevaluasi performa akurasi dalam mengenali wajah pegawai	Pengujian kombinasi model FaceNet dan SVM serta OpenFace	Model FaceNet dan Support Vector Machine (SVM) dalam pengenalan wajah mencapai akurasi sebesar 100% sedangkan model Openface sebesar 93,3%
2.	Winarno Sugeng dan Daniel Barus	Pengecekan Foto Paspor Menggunakan Metode DNN dan <i>Facenet</i> Sebagai Pengenalan Wajah	2023	Menguji keakuratan sistem dalam mengidentifikasi wajah seseorang meskipun terdapat objek atau aksesoris yang menutupi sebagian wajah	<i>Deep Neural Network</i> dan <i>Facenet</i>	Terdapat masalah pada pengenalan wajah dari samping, pengaruh cahaya dari webcam tetapi masih tergolong akurasi baik sebesar 97,48% terhadap 6107 dataset
3.	Dewangga Mantara, Wahyu Sudoro Murti, Ayu Kurniasari, dan Jaml Rosid	<i>Face recognition</i> dengan metode Haar Cascade dan <i>Facenet</i>	2022	Menguji keakuratan sistem dalam mendeteksi keberadaan wajah secara <i>real-time</i> dengan variasi kemiringan,	Haar Cascade dan <i>Facenet</i>	Sistem terbukti efektif dan akurat dengan hasil 80% dalam pengenalan dengan 6 dataset

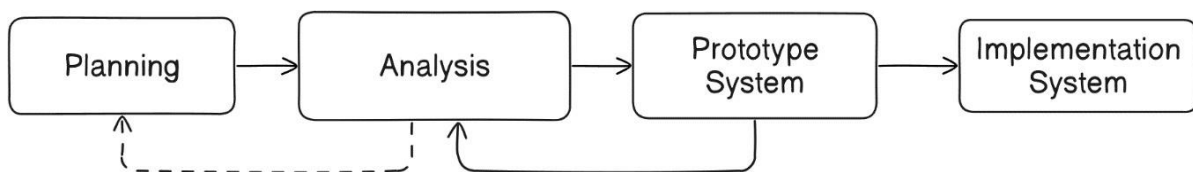
				jarak, dan intensitas cahaya		
4.	Lalu Fiqihilmi, Lalu A.Syamsul Irfan Akbar, Dwi Ratnasari, dan Cipta Ramadhani	Implementasi Pengenalan Wajah Pada Aplikasi Presensi Perkuliahan Menggunakan FaceNet Berbasis Android	2023	Mengevaluasi tingkat ketepatan sistem pengenalan wajah menggunakan FaceNet dalam aplikasi presensi mahasiswa berbasis Android	FaceNet	Sistem menunjukkan keakuratan sebesar 94,44%
5.	Natanniel Eka Chrisyanto, Eriq Muhammad Adams Jonemaro, dan Novanto Yudistira	Pengembangan Aplikasi Android Presensi Kehadiran <i>Realtime</i> menggunakan Pengenalan Wajah dengan Model Facenet	2022	Menguji keakuratan pengenalan wajah	FaceNet	Sistem memiliki nilai hasil pengujian keakuratan pengenalan wajah sebesar 100% dengan 15 percobaan

BAB III

PELAKSANAAN MAGANG

3.1 Manajemen Proyek

PT Indo Xetia Internasional menggunakan metode pengembangan aplikasi yang dikenal sebagai *prototyping*. Metode *prototyping* adalah metode pada pengembangan perangkat lunak yang melibatkan pendekatan dalam merancang perangkat lunak secara cepat dan bertahap agar calon pengguna atau klien dapat mengevaluasi seperti pada Gambar 3.1. Pengembang dan calon pengguna dapat berkomunikasi satu sama lain selama proses pembuatan prototipe. Terkadang calon pengguna hanya memberikan definisi umum tentang kebutuhan tanpa merinci proses *input* dan *output* dari sistem yang akan dikembangkan sehingga diperlukan kerjasama yang baik agar pengembang dapat memahami dengan jelas kebutuhan calon pengguna (Hary Prasetyo, 2017).



Gambar 3.1 Tahapan Metode *Prototyping*

(sumber: Mardhiah Fadhli & Annisa Marion, 2022)

Berikut merupakan penjelasan dari Gambar 3.1 Tahapan Metode *Prototyping*:

1. Tahap *planning* adalah proses awal dalam metode *prototyping* meliputi penyusunan rencana kerja proyek secara menyeluruh, menetapkan tujuan proyek, ruang lingkup, anggaran sumber daya, dan jadwal pelaksanaan proyek.
2. Tahap *analysis* adalah proses mengidentifikasi dan memahami kebutuhan sistem agar sistem dapat berintegrasi dengan proses bisnis yang sedang berjalan.
3. Tahap *prototype system* adalah proses identifikasi dan pemahaman kebutuhan sistem dengan membangun prototipe sederhana. Pembuatan prototipe dapat dilakukan secara berulang-ulang untuk perbaikan dan penyempurnaan fitur serta desain sistem. Setiap iterasi prototipe melibatkan analisis ulang kebutuhan, perancangan ulang, dan pengujian oleh pengguna.
4. Tahap *implementation system* adalah melibatkan konversi desain sistem menjadi bentuk yang dapat dieksekusi oleh komputer seperti menulis kode program.

Penulis menjalani kegiatan magang sebagai *mobile developer* di PT Indo Xetia Internasional selama periode 6 bulan. PT Indo Xetia Internasional merupakan perusahaan yang menerapkan sistem kerja *work form home* yakni karyawan diberi kesempatan untuk bekerja dari rumah sehingga memberikan kenyamanan dan fleksibilitas bagi karyawan dalam menjalankan tugas. Selama menjalani magang, penulis terlibat dalam berbagai proyek pengembangan aplikasi *mobile* di bawah bimbingan mentor. Penulis aktif berpartisipasi dalam rapat tim yang dilakukan secara daring melalui berbagai platform komunikasi. Aturan kerja yang berlaku di PT Indo Xetia Internasional adalah setiap karyawan wajib bekerja selama 5 hari dalam seminggu dengan durasi kerja 8 jam per hari. Waktu kerja disesuaikan dengan jadwal yang telah diterapkan oleh perusahaan untuk menjaga produktivitas dalam pelaksanaan tugas sehari-hari.

Kegiatan magang dimulai dengan peninjauan terhadap pekerjaan yang telah dikerjakan pada hari sebelumnya. Jika pekerjaan tersebut telah sesuai maka penulis dapat melanjutkan tugasnya. Namun, apabila terdapat kekurangan atau perlu perbaikan maka revisi akan dilakukan untuk memastikan kualitas sistem yang dikembangkan. Selama menjalankan magang, penulis terkadang menghadapi kendala atau hambatan dalam mengerjakan tugas yang telah diberikan sehingga untuk mengatasi hal tersebut dapat menghubungi mentor untuk meminta bimbingan dan bantuan dalam menyelesaikan masalah yang dihadapi. Diskusi dengan mentor juga dapat membantu dalam perbaikan dan pengembangan sistem. PT Indo Xetia Internasional mengadakan rapat bulanan khusus untuk peserta magang yang bertujuan untuk mengevaluasi kemajuan pekerjaan yang telah dilakukan selama sebulan. Melalui rapat bulanan maka semua peserta magang memiliki kesempatan untuk menerima umpan balik, saran, dan arahan agar meningkatkan kualitas dan perkembangan sistem yang sedang dikembangkan.

3.2 Pelaksanaan Proyek

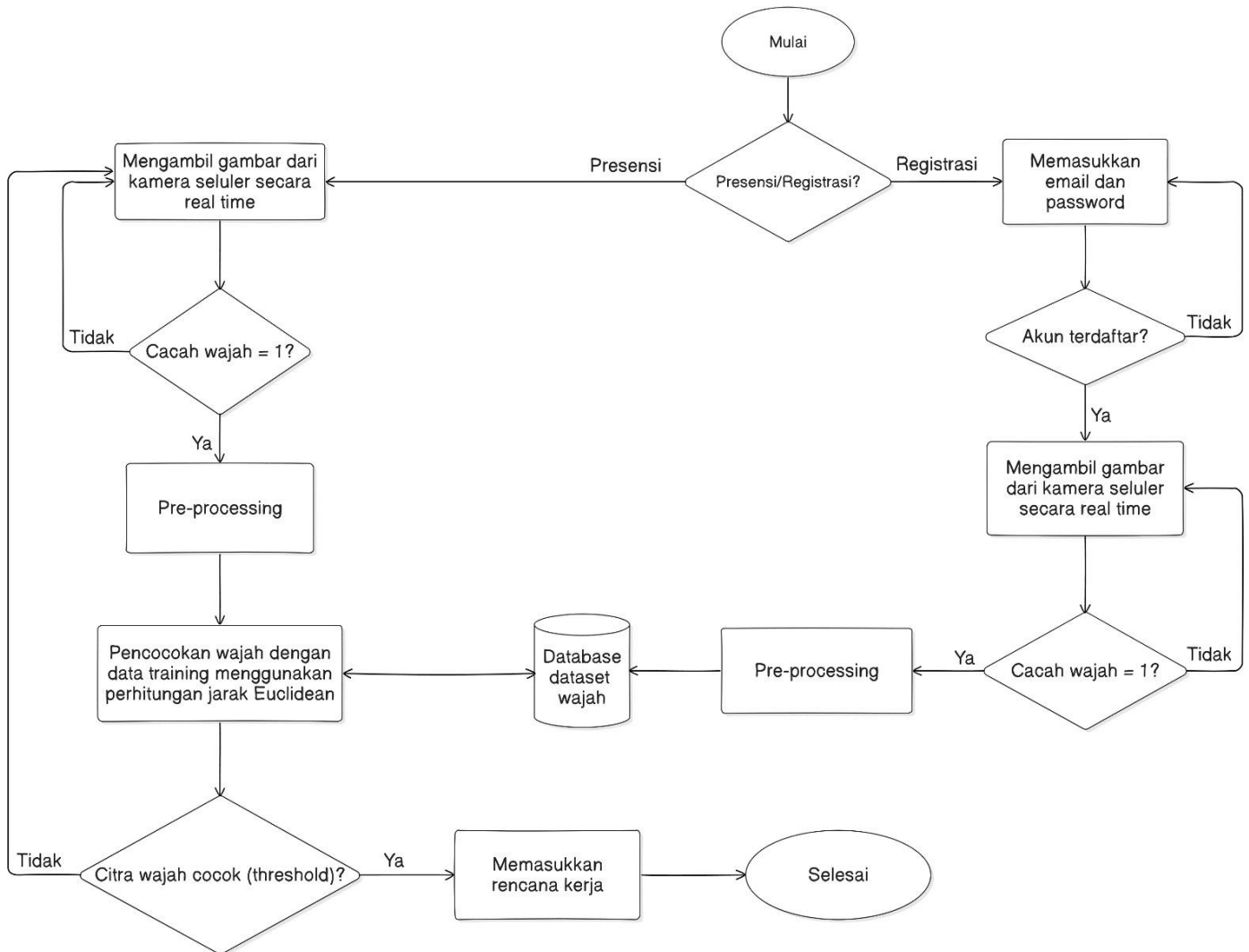
Pada pengerjaan proyek sistem presensi karyawan dengan *face recognition* menggunakan *framework* Flutter terdapat beberapa tahapan yaitu perancangan prototipe, pengembangan fitur *face registration*, *face recognition*, *clock in*, *clock out*, *daily standup*, pengujian *face recognition* di beberapa situasi. Spesifikasi untuk mengembangkan sistem presensi karyawan seperti pada Tabel 3.1.

Tabel 3.1 Spesifikasi Sistem Presensi Karyawan

No.	Software	Versi
1.	Flutter	3.19.5
2.	Dart	3.3.3
3.	Emulator Android	11.0
4.	Android Studio Hedgehog	2023.1.1

3.2.1 Perancangan Prototipe

Perancangan prototipe merupakan tahap awal dalam pengembangan sistem presensi karyawan. Langkah pertama menganalisis kebutuhan melalui observasi langsung untuk memahami proses presensi secara menyeluruh. Interaksi dengan *stakeholder* terkait seperti sekretaris dan karyawan untuk mendapatkan masukan langsung dan harapan terhadap sistem presensi yang akan dikembangkan. Setelah kebutuhan teridentifikasi maka dapat merancang desain sistem dengan arsitektur yang melibatkan beberapa tahapan untuk menghasilkan alur kerja sistem presensi masuk pada Gambar 3.2 dan sistem presensi keluar pada Gambar 3.5. Tim pengembang kemudian dapat membangun prototipe dengan mencakup fitur utama yang kemudian didiskusikan kepada *stakeholder* untuk mendapatkan masukan. Berdasarkan masukan tersebut maka tim dapat melakukan penyesuaian pada prototipe. Proses pembuatan prototipe dapat berulang kali dilakukan untuk meningkatkan fitur dan desain sistem. Setiap iterasi prototipe melibatkan analisis kebutuhan, perancangan ulang, dan pengujian oleh pengguna untuk memastikan kesesuaian sistem berdasarkan kebutuhan. Sistem presensi karyawan mengalami dua iterasi yaitu perubahan warna pada tampilan sistem dan fitur aktivasi akun.

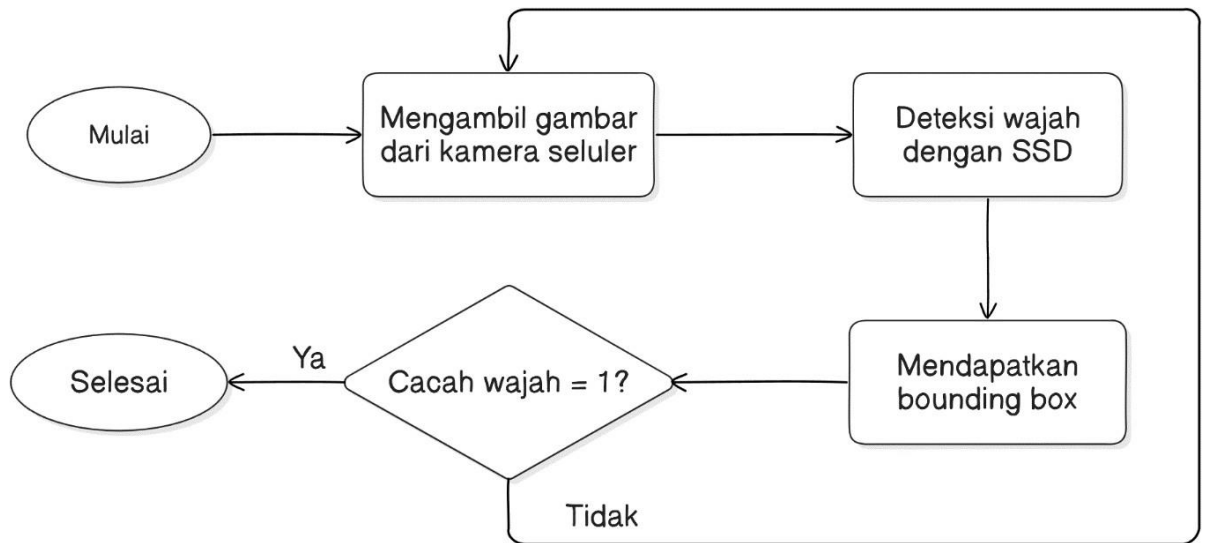


Gambar 3.2 *Flowchart* Sistem Presensi Masuk

Berikut adalah penjelasan mengenai *flowchart* sistem presensi masuk pada Gambar 3.2:

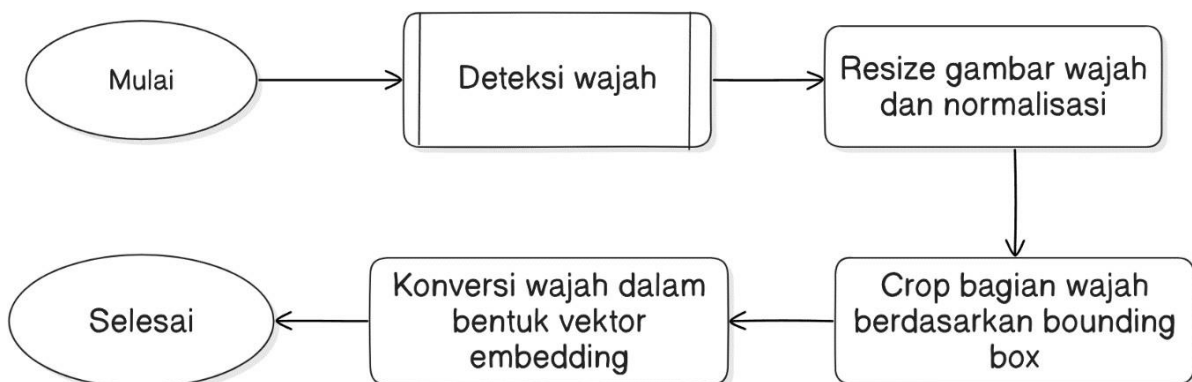
1. Memasukkan email dan password yang telah terdaftar dan apabila berhasil akan dialihkan ke halaman selanjutnya. Namun, sistem akan menampilkan peringatan apabila akun tidak terdaftar.
2. Mengambil gambar menggunakan kamera perangkat secara *real-time* sehingga karyawan hanya dapat mengambil gambar di kondisi nyata dan tidak dapat menggunakan gambar yang tersimpan di galeri perangkat. Gambar hanya diambil satu kali untuk memastikan keaslian dan keakuratan pada kondisi nyata.
3. Deteksi wajah adalah proses pencarian keberadaan wajah dalam sebuah gambar. Algoritma SSD bekerja untuk mendeteksi dan mengidentifikasi wajah dalam gambar sehingga mempersiapkan data yang diperlukan untuk proses pengenalan wajah. Algoritma tersebut memastikan bahwa semua wajah dalam gambar teridentifikasi dengan tepat sehingga data

yang diperoleh dapat digunakan untuk langkah pengenalan. Detail alur deteksi wajah dapat dilihat pada *flowchart* yang terlampir pada Gambar 3.3.



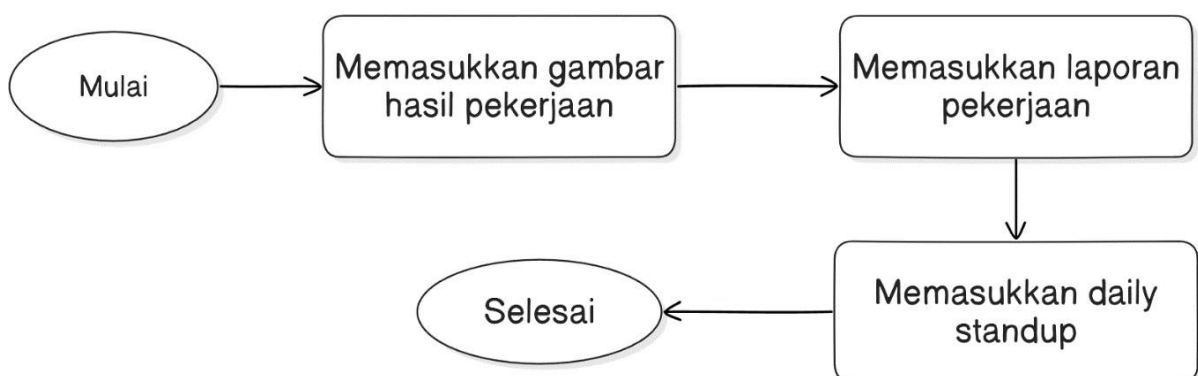
Gambar 3.3 *Flowchart* Deteksi Wajah

4. Tahap *pre-processing* melibatkan pengolahan lanjutan terhadap citra wajah yang telah terdeteksi. Setelah wajah terdeteksi maka dilakukan transformasi vektor untuk memastikan citra wajah berada dalam format yang sesuai untuk analisis lanjutan. Proses ini mencakup normalisasi ukuran, rotasi, dan transformasi lainnya untuk menghasilkan representasi vektor yang lebih standar. Selanjutnya, dilakukan *cropping* berdasarkan *bounding box* untuk menghapus bagian citra yang tidak relevan dan mempertahankan hanya bagian wajah yang diperlukan sehingga menghilangkan *noise* dan fokus pada fitur wajah yang penting. Terakhir adalah wajah akan dikonversi dalam bentuk vektor embedding. Detail alur *pre-processing* dapat dilihat pada *flowchart* yang terlampir pada Gambar 3.4.



Gambar 3.4 *Flowchart* Pre-processing

5. Proses pencocokan wajah dilakukan dengan menggunakan perhitungan jarak Euclidean yang melibatkan model FaceNet. Jarak Euclidean adalah metrik yang digunakan untuk mengukur jarak antara dua titik dalam ruang Euclidean. Dalam konteks pengenalan wajah, jarak ini digunakan untuk membandingkan representasi vektor dari dua citra wajah. Model FaceNet menghasilkan vektor fitur dari citra wajah dan jarak Euclidean antara dua vektor dihitung untuk menentukan seberapa mirip kedua wajah tersebut. Semakin kecil jaraknya maka semakin besar kemiripan antara wajah-wajah tersebut sehingga memungkinkan pencocokan yang akurat. Pengambilan gambar pada proses pencocokan wajah dilakukan menggunakan kamera perangkat secara *real-time* sehingga karyawan hanya dapat mengambil gambar dalam kondisi nyata dan tidak dapat menggunakan gambar yang tersimpan di galeri perangkat sehingga memastikan keaslian dan keakuratan proses pengenalan wajah. Gambar hanya diambil satu kali untuk memastikan keaslian dan keakuratan proses pengenalan wajah. Sistem ini tidak melakukan pelatihan model secara mandiri karena menggunakan *library* yang disediakan oleh Flutter untuk menjalankan model FaceNet yang telah dilatih sebelumnya sehingga sistem dapat fokus pada proses pencocokan dan identifikasi wajah.
7. Memasukkan rencana kerja dengan detail dan jelas sehingga memastikan bahwa setiap tugas dan tanggung jawab tercatat dengan baik. Apabila berhasil maka proses presensi masuk telah berjalan dengan baik sehingga menandakan bahwa semua data yang diperlukan telah diinput dengan benar.



Gambar 3.5 *Flowchart* Sistem Presensi Keluar

Berikut adalah penjelasan mengenai *flowchart* sistem presensi keluar pada Gambar 3.5:

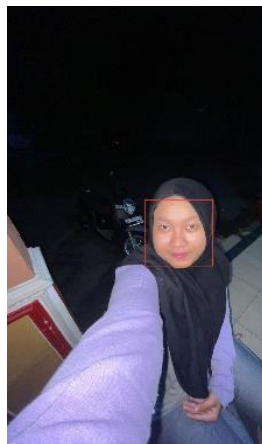
1. Memasukkan gambar hasil pekerjaan yang telah dikerjakan dan sistem akan mengarahkan ke galeri untuk memilih gambar yang akan digunakan.

2. Memasukkan laporan pekerjaan secara detail sehingga memastikan setiap tugas yang telah dikerjakan tercatat dengan baik.
3. Memasukkan *daily standup* berdasarkan perkembangan kerja yang telah dilakukan pada hari sebelumnya, pencapaian pada hari ini, rencana untuk besok, dan masalah atau kendala yang dihadapi selama mengerjakan proyek.

3.2.2 Pengembangan Fitur *Face Registration*

Sistem presensi karyawan memanfaatkan *library* Google ML Kit dan Tensorflow Lite yang tersedia pada Flutter. *Library* Google ML Kit seperti 'google_mlkit_commons' menyediakan komponen yang digunakan oleh berbagai modul dalam Google ML Kit mencakup beragam fungsi dasar pada pengembangan aplikasi berbasis *machine learning* termasuk pengolahan gambar, pemrosesan data, dan fungsi utilitas lainnya. Pada penerapan fitur deteksi wajah digunakan 'google_ml_kit_face_detection' untuk autentikasi pengguna, analisis ekspresi wajah, atau penandaan wajah pada gambar. *Library* Tensorflow Lite seperti 'tflite_flutter' mengintegrasikan model Tensorflow Lite ke dalam aplikasi Flutter.

Sistem memanfaatkan SSD dalam mengenali objek pada gambar termasuk wajah manusia. SSD menggunakan arsitektur CNN dalam mendeteksi objek, menghasilkan kotak pembatas atau *bounding box*, dan skor probabilitas untuk setiap objek yang terdeteksi. SSD tidak terbatas pada penggunaan FaceNet tetapi dapat menggunakan berbagai arsitektur CNN lainnya tergantung pada konfigurasi. Tensorflow Lite menyediakan SSD sebagai salah satu model deteksi objek yang dapat diterapkan pada aplikasi *mobile* untuk mengenali objek termasuk wajah dengan skor probabilitas tertentu. Implementasi deteksi wajah pada sistem dapat dilihat pada Gambar 3.6.



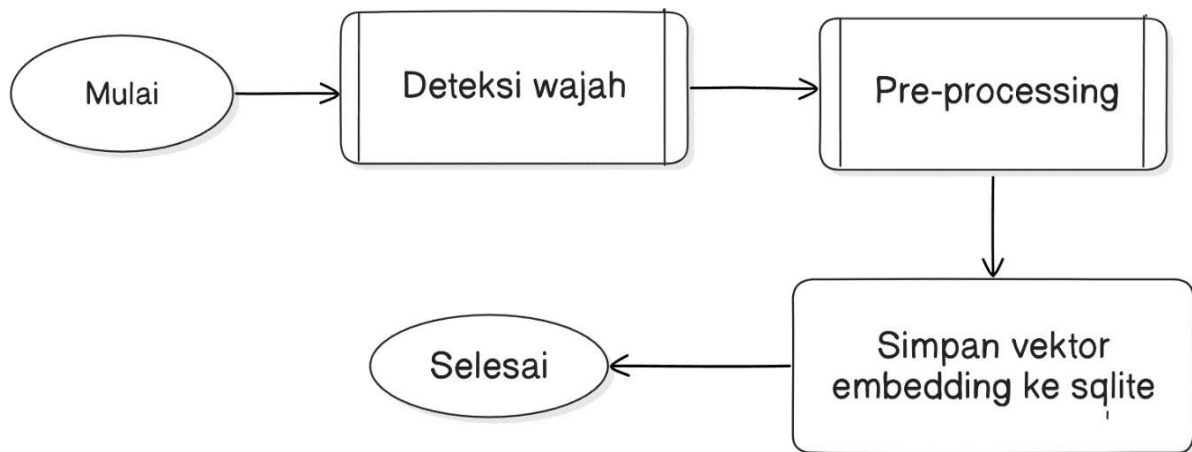
Gambar 3.6 Deteksi Wajah

Pada tahap pendaftaran wajah penggunaan SQLite sangat penting untuk menyimpan data wajah menggunakan bantuan *library* 'sqflite' dari Flutter. SQLite memungkinkan penyimpanan data secara lokal pada perangkat pengguna sehingga menjamin keamanan data dari akses pihak luar. Kelas 'DatabaseHelper' dibuat untuk mempermudah interaksi dengan SQLite agar operasi basisdata dan manajemen data dapat dioptimalkan. Kelas ini bertujuan untuk memastikan integritas data serta menjaga kinerja sistem tetap optimal sehingga proses penyimpanan dan pengelolaan data wajah dapat dilakukan secara terstruktur.

Pada direktori ML terdapat dua kelas yakni 'Recognition' dan 'Recognizer'. Kelas 'Recognition' berperan sebagai representasi hasil pengenalan wajah sehingga setiap hasil dari kelas ini dilengkapi dengan informasi seperti nama, lokasi, gambar wajah, data embedding, dan jarak. Fungsi utamanya adalah untuk menyimpan semua informasi terkait hasil pengenalan wajah termasuk hasil pemrosesan dari proses inferensi model yang digunakan pada sistem pengenalan wajah.

Peran kelas 'Recognizer' sangat penting dalam mengelola keseluruhan proses pengenalan wajah. Kelas ini bertanggung jawab atas langkah-langkah seperti registrasi wajah, proses *machine learning*, dan melakukan inferensi pada data wajah yang diberikan. Pada ruang lingkup fungsionalitas 'Recognizer' memfasilitasi proses registrasi dan pengenalan wajah menggunakan TensorFlow Lite. Kelas ini juga terhubung dengan SQLite untuk penyimpanan dan pengambilan data wajah. 'Recognizer' menyediakan fungsi-fungsi seperti inisiasi dan membersihkan sumber daya yang digunakan pada proses pengenalan wajah.

Pada pembuatan kelas 'FaceRegistration' memungkinkan pengguna untuk mendaftarkan wajah pada sistem menggunakan teknologi deteksi dan pengenalan wajah. Pada tahap inisiasi, beberapa objek dideklarasikan seperti 'ImagePicker' untuk memilih gambar secara langsung dari kamera perangkat. 'File_image' sebagai wadah penyimpanan yang menampung gambar secara langsung melalui kamera. 'FaceDetector' menjadi komponen dalam menangani deteksi wajah dengan memanfaatkan Google ML Kit Face Detection. Peran kelas 'Recognizer' terlihat pada pengembangan ini karena melakukan proses pengenalan wajah. Detail alur registrasi wajah dapat dilihat pada *flowchart* yang terlampir pada Gambar 3.7.

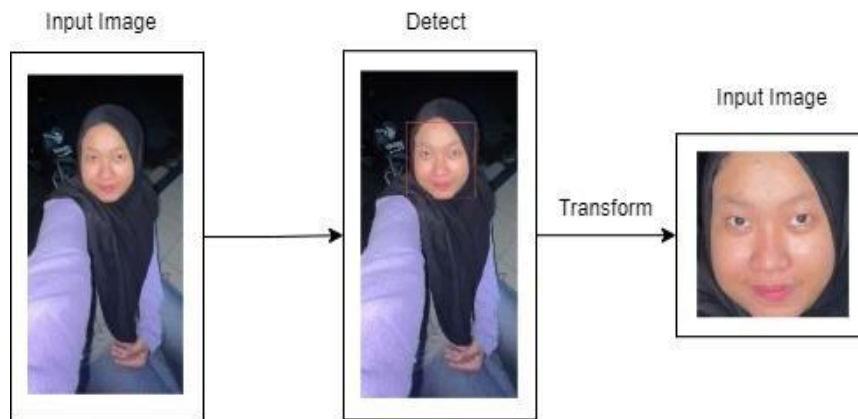


Gambar 3.7 *Flowchart* Registrasi Wajah

Pada pengembangan fitur *face registration* terdapat fungsi-fungsi seperti ‘imgFromCamera()’ yang berfungsi untuk membuka kamera perangkat sehingga pengguna dapat mengambil gambar wajah dan memulai proses deteksi wajah. Fungsi ‘navigateToHomeScreen()’ digunakan untuk mengarahkan pengguna kembali ke layar utama sistem setelah berhasil menyelesaikan proses registrasi. Fungsi penting lainnya adalah ‘doFaceDetection()’ untuk menangani deteksi wajah pada gambar yang diambil termasuk situasi terdapat lebih dari satu wajah atau tidak ada wajah yang terdeteksi. Sedangkan, fungsi ‘removeRotation(FileinputImage)’ berperan dalam menghapus rotasi sebelum memulai proses deteksi wajah untuk memastikan akurasi deteksi. Terakhir adalah fungsi ‘drawRectangleAroundFaces()’ digunakan untuk menandai wajah yang terdeteksi dengan persegi pada gambar menggunakan *bounding box*.

Citra wajah diubah menjadi *array byte* yang dapat dijalankan pada proses pengenalan wajah. Pada langkah ini, gambar wajah akan mengalami beberapa tahap seperti pada Gambar 3.8 sebagai berikut:

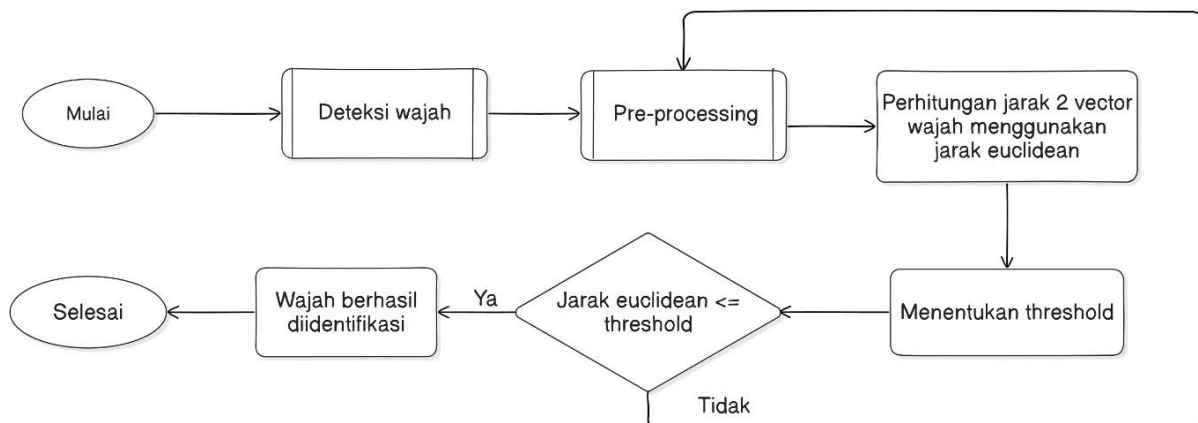
1. Gambar wajah dipotong berdasarkan *bounding box* yang menunjukkan area wajah yang terdeteksi
2. Gambar wajah diubah ukurannya dan dinormalisasi untuk proses selanjutnya sehingga memastikan konsistensi dalam representasi gambar wajah.
3. Setiap gambar wajah yang sudah dinormalisasi akan diubah menjadi vektor embedding. Vektor ini kemudian disimpan dalam SQLite sehingga data wajah yang telah terdaftar dapat diakses untuk proses pengenalan wajah berikutnya.



Gambar 3.8 Pendaftaran Wajah

3.2.3 Pengembangan Fitur *Face Recognition*

Fitur pengenalan wajah menggunakan algoritma deteksi wajah dari Google ML Kit Face Detection serta kelas ‘Recognizer’ untuk melakukan proses pengenalan wajah. Pengguna dapat mengambil gambar wajah melalui kamera perangkat kemudian sistem akan memprosesnya untuk mengenali wajah yang terdeteksi dan memverifikasi keberadaannya dalam basisdata yang terkait. Fitur ini juga memungkinkan perbandingan antara wajah yang terdeteksi dengan wajah yang telah terdaftar selama proses pendaftaran wajah. Detail alur pengenalan wajah dapat dilihat pada *flowchart* yang terlampir pada Gambar 3.9.

Gambar 3.9 *Flowchart* Pengenalan Wajah

Pada pengembangan fitur *face recognition* terdapat beberapa fungsi yang menjadi landasan seperti fungsi ‘imgFromCamera()’ membuka kamera perangkat agar pengguna dapat mengambil gambar wajah dan memulai proses deteksi. Selanjutnya, fungsi ‘navigateToClockIn()’ mengarahkan pengguna ke layar ‘ClockInScreen’ setelah proses pengenalan wajah berhasil diselesaikan. Untuk memeriksa wajah yang terdeteksi dengan yang

terdaftar dalam basisdata digunakan fungsi ‘isFaceRegistered(Recognition recognition)’. Fungsi ‘processImage()’ digunakan untuk deteksi wajah pada gambar yang diambil termasuk penanganan jika terdeteksi lebih dari satu wajah atau tidak ada wajah yang terdeteksi. Ketika wajah tidak terdaftar maka fungsi ‘showUnregisteredFaceDialog()’ menampilkan dialog. Sedangkan fungsi ‘drawRectangleAroundFaces()’ berfungsi untuk menampilkan persegi panjang di sekitar wajah yang terdeteksi. Terakhir ‘build()’ digunakan untuk membuat tampilan *widget* pada layar pengenalan wajah termasuk teks pengantar, gambar, dan tombol untuk mengambil gambar dari kamera atau melanjutkan ke layar presensi.

Proses ekstraksi data wajah akan dilakukan oleh FaceNet CNN. Deskriptor wajah yang telah diekstraksi akan dibandingkan dengan dataset deskriptor wajah yang sudah tersimpan dalam bentuk *array* numerik di basisdata. Sistem akan menghitung jarak Euclidean antara kedua deskriptor wajah dan mengevaluasi kedua wajah tersebut serupa berdasarkan pada nilai *threshold*. Semakin kecil nilai jarak Euclidean antara dua vektor maka semakin mirip atau serupa kedua vektor dianggap (Setiawan et al., 2024).

Pada sistem ini tidak ada proses pelatihan model dikarenakan sistem menggunakan *library* yang disediakan oleh Flutter untuk menjalankan model FaceNet yang sudah dilatih sebelumnya. Sistem hanya berfokus pada proses pencocokan dan identifikasi wajah berdasarkan deskriptor yang dihasilkan oleh model yang telah dilatih tersebut. Proses pengambilan gambar dilakukan menggunakan kamera perangkat secara *real-time* sehingga karyawan hanya dapat mengambil gambar dalam kondisi nyata dan tidak dapat menggunakan gambar yang tersimpan di galeri perangkat. Gambar hanya diambil satu kali baik untuk memastikan keaslian dan keakuratan proses pengenalan wajah.

3.2.4 Pengembangan Fitur *Clock In*

Pada pengembangan *clock in* terdapat beberapa variabel yang dideklarasikan seperti ‘formattedDate’ digunakan untuk menyimpan tanggal saat ini dalam format tentu, ‘image’ merupakan gambar yang diterima melalui konstruktor, ‘submitTime’ menyimpan waktu saat ini dalam format tertentu, dan ‘textEditingController’ adalah controller yang digunakan untuk mengontrol input teks di TextField.

Terdapat beberapa fungsi seperti ‘createClockIn()’ digunakan untuk menyimpan data *clock in* ke Firestore. Fungsi ini mengambil email pengguna, mendapatkan referensi ke koleksi ‘clock-in’ di Firestore, dan menambahkan dokumen baru ke koleksi tersebut yang berisi email

pengguna, *timestamp*, dan teks rencana yang diinput oleh pengguna. Fungsi ini juga menangani kesalahan jika terjadi masalah saat menyimpan data.

Pada fungsi 'build' *user interface* layar dibangun menggunakan 'Scaffold' dengan latar belakang berwarna dan konten diatur dalam 'SingleChildScrollView'. Sistem menampilkan tanggal saat ini di bagian atas dan diikuti dengan 'TextField' untuk memasukkan rencana harian pengguna. Tombol submit diperiksa untuk memastikan 'TextField' tidak kosong sebelum data disimpan. Apabila 'TextField' tidak terisi maka sistem akan menampilkan dialog peringatan. Namun, apabila terisi maka data akan disimpan menggunakan fungsi 'createClockIn' dan pengguna diarahkan ke 'HomeScreen' dengan membawa data input yang telah dimasukkan.

3.2.5 Pengembangan Fitur *Clock Out*

Pada pengembangan *clock out* terdapat beberapa variabel yang dideklarasikan seperti 'formattedDate' untuk menyimpan tanggal saat ini dalam format tentu, 'img' adalah gambar yang diterima melalui konstruktor, 'submitTime' menyimpan waktu saat ini dalam format tertentu, dan 'clockOutController' adalah TextEditingController untuk mengontrol input teks pada TextField.

Terdapat beberapa fungsi seperti 'checkRole' untuk memeriksa peran pengguna berdasarkan UID yang diambil dari Firebase Authentication. Fungsi 'createClockOut' digunakan untuk menyimpan data *clock out* ke Firestore meliputi email pengguna, waktu saat ini (*timestamp*), laporan yang diinput oleh pengguna, dan URL gambar dari konstruktor. Apabila terjadi kesalahan saat menyimpan data maka sistem akan menampilkan pesan error.

Fungsi 'build' membangun *user interface* layar menggunakan 'Scaffold' dengan latar belakang berwarna dan konten diatur dalam 'SingleChildScrollView'. *User interface* ini terdiri dari beberapa elemen seperti tanggal saat ini ditampilkan pada bagian atas, 'TextField' untuk input laporan harian pengguna, dan tombol submit. Tombol submit akan memeriksa 'TextField' terisi atau tidak. Apabila tidak terisi maka sistem akan menampilkan dialog peringatan dan apabila terisi maka data akan disimpan menggunakan fungsi 'createClockOut'.

3.2.6 Pengembangan Fitur *Daily Standup*

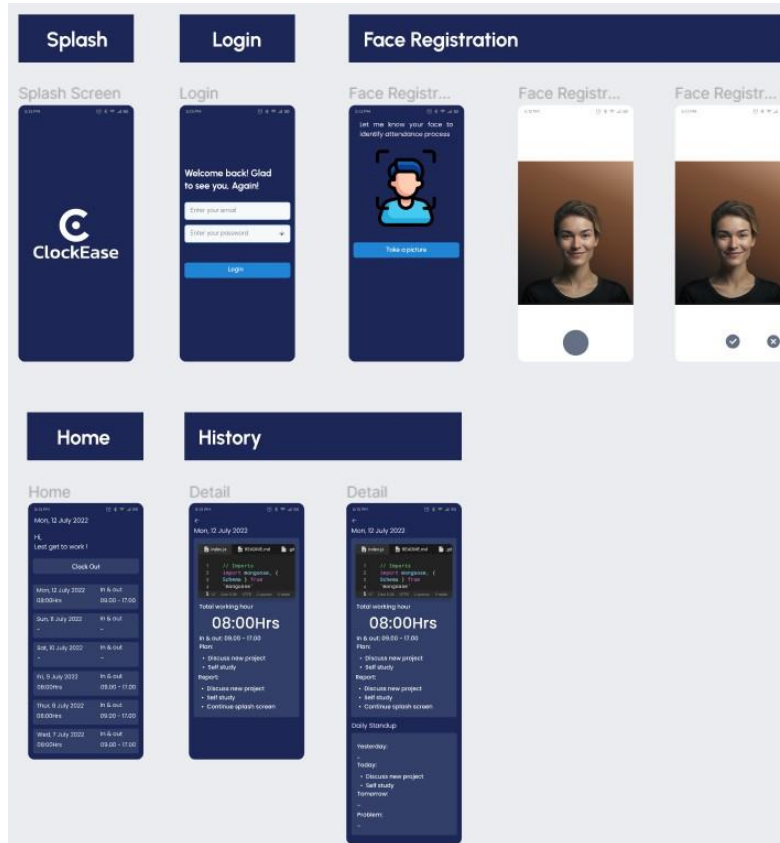
Pada pengembangan *daily standup* terdapat beberapa 'TextEditingController' untuk mengontrol input teks pada TextField seperti 'yesterdayController', 'todayController', 'tomorrowController', dan 'problemController'. Terdapat beberapa fungsi seperti

'createDailyStandup' untuk menyimpan data *daily standup* ke Firestore meliputi email pengguna, waktu saat ini (*timestamp*), tugas yang telah dilakukan pada hari sebelumnya, hari ini, besok, dan masalah yang dihadapi selama mengerjakan proyek. Fungsi 'build' untuk membangun *user interface* layar menggunakan 'Scaffold' dengan latar belakang berwarna dan konten diatur dalam 'SingleChildScrollView', menampilkan tanggal saat ini di bagian atas layar, menampilkan teks "Daily Standup", menampilkan empat 'TextField' untuk tugas kemarin, hari ini, besok, dan masalah serta memiliki kontrol *input* melalui 'TextEditingController' yang sesuai. Tombol submit ditekan dan memanggil fungsi 'createDailyStandup()' untuk menyimpan data ke Firestore dan dialihkan ke halaman selanjutnya.

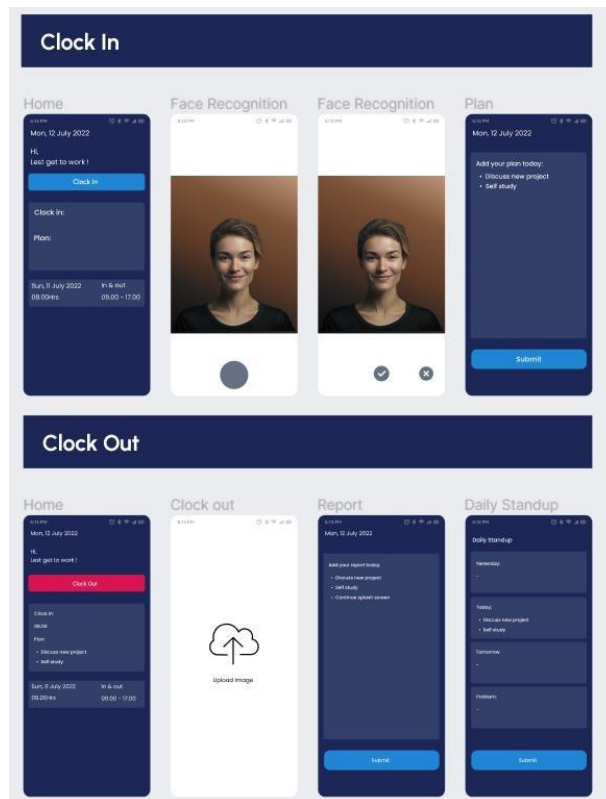
3.3 Hasil Proyek

3.3.1 Prototipe Sistem Presensi Karyawan

Prototipe sistem presensi karyawan pada Gambar 3.10 dan Gambar 3.11 merupakan representasi awal dari desain dan fungsi sistem yang dikembangkan. Prototipe memberikan gambaran tentang sistem akan beroperasi dan membantu dalam mengidentifikasi serta memperbaiki kekurangan sebelum implementasi penuh. Prototipe ini mencakup berbagai fitur utama seperti perekaman waktu masuk dan keluar karyawan, verifikasi identitas melalui teknologi pengenalan wajah serta integrasi dengan basisdata untuk penyimpanan dan pengelolaan data presensi.



Gambar 3.10 Prototipe Sistem Presensi Karyawan (1)

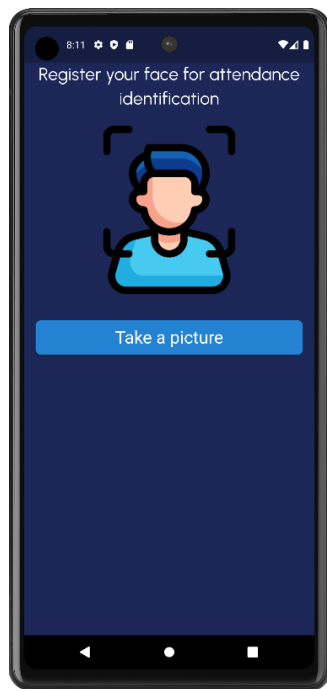


Gambar 3.11 Prototipe Sistem Presensi Karyawan (2)

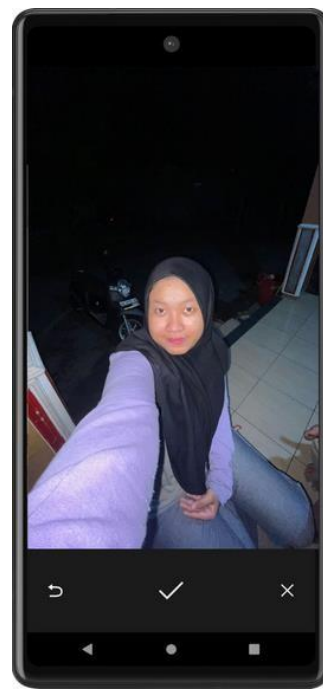
Pembuatan prototipe dilakukan melalui beberapa tahap yakni dimulai dari perancangan antarmuka pengguna yang intuitif dan mudah digunakan. Antarmuka tersebut dirancang untuk meminimalisir kesalahan pengguna dan mempercepat proses presensi. Prototipe juga diuji coba dengan melibatkan beberapa pengguna untuk mendapatkan umpan balik mengenai performa dan kegunaannya. Umpan balik sangat bermanfaat dalam menyempurnakan desain dan fungsionalitas sistem sebelum diterapkan.

3.3.2 Tampilan *Face Registration*

Antarmuka pengguna untuk fitur *face registration* yang telah dikembangkan seperti pada Gambar 3.12 menawarkan opsi kepada pengguna untuk mengakses fungsionalitas registrasi wajah melalui tombol ‘Take a picture’. Tombol tersebut membuat perangkat secara otomatis akan mengaktifkan kamera untuk mengambil gambar wajah pengguna seperti pada Gambar 3.13. Apabila tidak ada wajah yang terdeteksi atau terdeteksi lebih dari satu maka sistem akan menampilkan peringatan kepada pengguna.



Gambar 3.12 Tampilan *Face Registration*

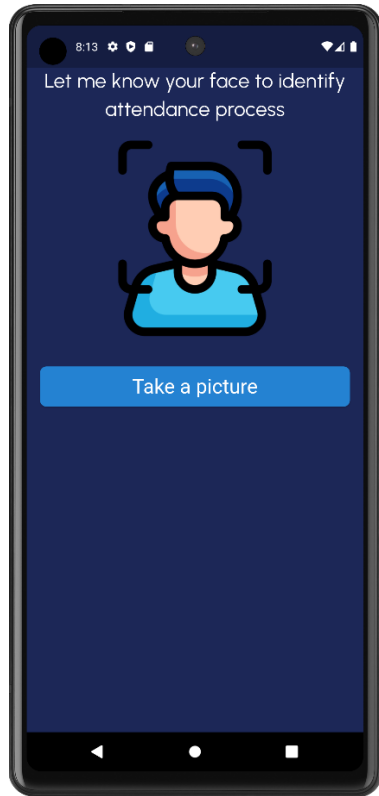


Gambar 3.13 Pengambilan Gambar

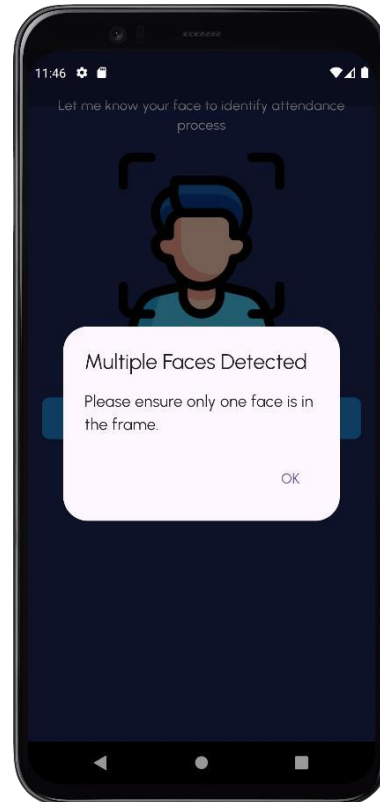
3.3.3 Tampilan *Face Recognition*

Tampilan *face recognition* yang telah dikembangkan seperti pada Gambar 3.14 terdapat tombol ‘Take a picture’ sehingga secara otomatis akan mengaktifkan kamera untuk mengambil gambar wajah pengguna seperti pada Gambar 3.13. Proses *face recognition* adalah pengenalan

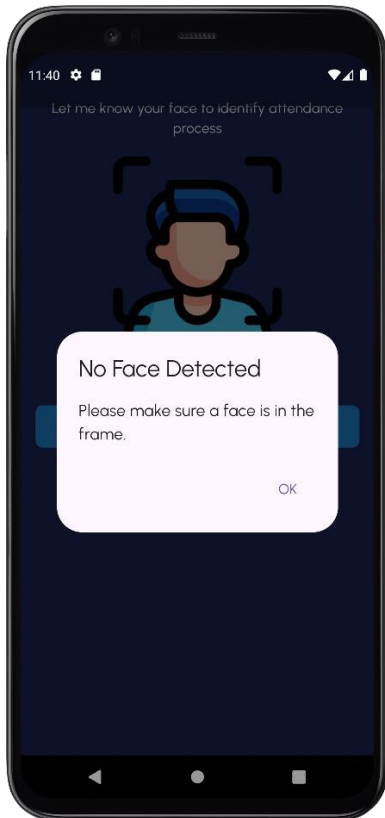
wajah yang terdeteksi dengan data wajah yang telah terdaftar pada basisdata untuk memverifikasi wajah yang terdeteksi. Peringatan akan ditampilkan pada sistem apabila lebih dari satu wajah terdeteksi seperti pada Gambar 3.15 atau tidak ada wajah yang terdeteksi seperti pada Gambar 3.16. Apabila wajah yang terdeteksi tidak sesuai dengan basisdata maka sistem juga akan menampilkan peringatan seperti pada Gambar 3.17.



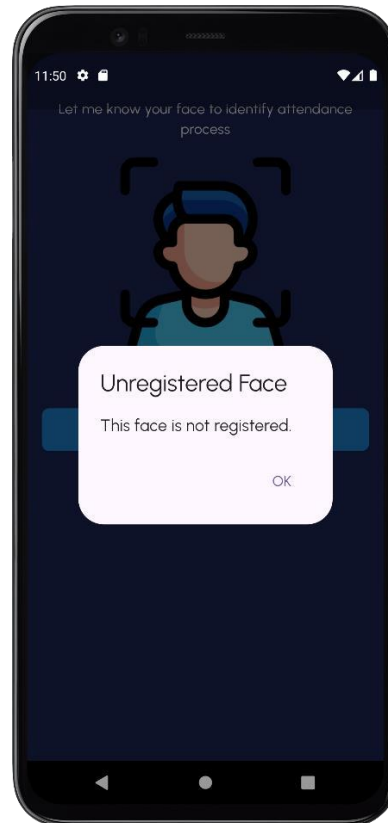
Gambar 3.14 Tampilan *Face Recognition*



Gambar 3.15 Peringatan Lebih dari Satu



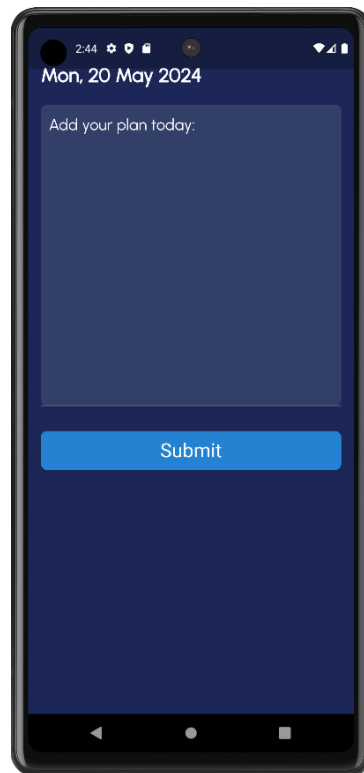
Gambar 3.16 Peringatan Tidak Terdeteksi



Gambar 3. 17 Peringatan Wajah Berbeda

3.3.4 Tampilan *Clock In*

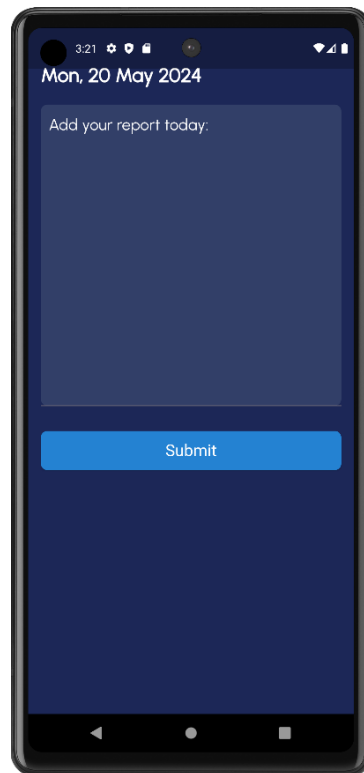
Antarmuka pengguna untuk fitur *clock in* yang menampilkan tanggal hari ini dan TextField seperti pada Gambar 3.18. Pengguna dapat memasukkan rencana kerja hari ini melalui TextField. Setelah pengguna mengisi rencana kegiatan maka dapat menekan tombol 'Submit' agar data kehadiran dapat tersimpan ke basisdata Firestore. Tombol ini akan memicu untuk menyimpan data kehadiran pengguna ke dalam basisdata Firestore. Jika pengguna mencoba menekan tombol 'Submit' tanpa mengisi rencana kerja maka akan muncul peringatan untuk mengisi rencana kerja sebelum melakukan *submit*. Setelah data berhasil disimpan maka pengguna akan dialihkan ke *home screen* untuk melanjutkan pekerjaan.



Gambar 3.18 Tampilan *Clock In*

3.3.5 Tampilan *Clock Out*

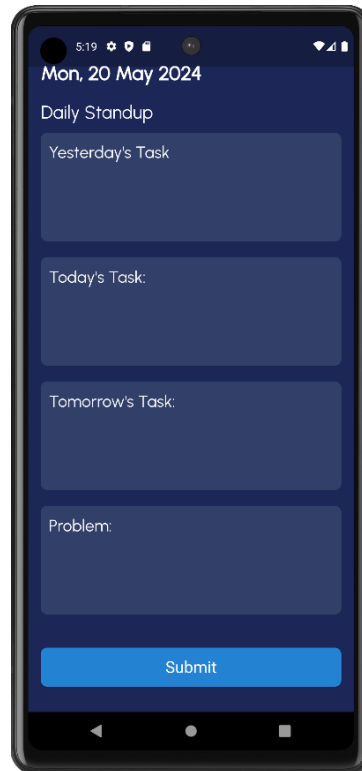
Fitur *clock out* digunakan pengguna saat melakukan presensi keluar. Pengguna dapat memasukkan gambar hasil pekerjaan dan sistem akan mengarahkan ke *clock out screen* seperti pada Gambar 3.19 yang menampilkan tanggal hari ini dalam format “E, dd MMM yyyy” dan pengguna diminta untuk memasukkan laporan mengenai pekerjaan yang telah dilakukan sebelum melakukan proses *submit* untuk menyimpan data.



Gambar 3.19 Tampilan *Clock Out*

3.3.6 Tampilan *Daily Standup*

Setelah pengguna memasukkan laporan kerja di *clock out* maka sistem akan mengarahkan ke halaman *daily standup* seperti pada Gambar 3.20. Pada halaman *daly standup* terdapat empat bagian berisi input teks seperti “Yesterday’s Task” untuk memasukkan pekerjaan yang telah dikerjakan kemarin, “Today’s Task” untuk memasukkan pekerjaan yang telah dikerjakan hari ini, “Tomorrow’s Task” untuk memasukkan pekerjaan yang akan dilakukan besok, dan “Problem” untuk memasukkan masalah atau kendala yang dialami selama bekerja hari ini.



Gambar 3.20 Tampilan *Daily Standup*




3.3.7 Hasil Pengujian




Penelitian pada sistem pengenalan wajah telah dilakukan oleh peneliti Ramdhon et al. (2021) melakukan uji coba pengenalan dan deteksi wajah pada sistem presensi dalam berbagai kondisi seperti cahaya terang, cahaya redup, posisi miring, posisi lurus, ekspresi tersenyum, ekspresi datar, jarak 100-250cm, jarak 300-500cm, atribut penutup kepala, dan kacamata sebanyak 5 iterasi untuk setiap jenis pengujian (Ramdhon & Febriya, 2021).


Peneliti Setiawan et al. (2024) menguji pengenalan dan deteksi wajah pada aplikasi presensi *mobile* menggunakan metode *deep learning* pada kondisi seperti cahaya terangm cahaya redup, jarak 30cm, jarak 60cm (Setiawan et al., 2024). Berdasarkan penelitian-penelitian tersebut maka pengujian sistem presensi pada penelitian ini dilakukan dalam berbagai kodisi seperti pada cahaya terang, cahaya redup, posisi miring, posisi lurus, ekspresi tersenyum, ekspresi datar, jarak 30cm, jarak 60 cm, penggunaan topi, dan kacamata sebanyak 5 iterasi untuk setiap jenis pengujian. Contoh pengujian pengenalan wajah dengan beberapa kondisi dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Contoh Pengujian

No.	Gambar	Jenis Pengujian
1.		Cahaya terang
2.		Cahaya redup
3.		Posisi miring

No.	Gambar	Jenis Pengujian
4.		Posisi lurus
5.		Ekspresi tersenyum
6.		Ekspresi datar

No.	Gambar	Jenis Pengujian
7.		Jarak 30cm
8.		Jarak 60cm
9.		Topi

No.	Gambar	Jenis Pengujian
10.		Kacamata

Pengujian dilakukan terhadap 15 calon pengguna dan setiap jenis pengujian melewati 5 iterasi percobaan. Proses pengujian bertujuan untuk menilai performa sistem pada penggunaan sehari-hari. Detail hasil pengujian sistem dapat ditemukan pada Tabel 3.3.

Tabel 3. 3 Pengujian *Face Recognition*

Jenis Pengujian	Dikenali	Tidak Dikenali	Tingkat Keberhasilan	Tingkat Gagal
Cahaya				
- Terang	74	1	98.7%	1.3%
- Redup	69	6	92%	8%
Posisi				
- Miring	41	34	54.7%	45.3%
- Lurus	72	3	96%	4%
Ekspresi				
- Tersenyum	71	4	94.7%	5.3%
- Datar	75	0	100%	0%
Jarak				
- 30cm	75	0	100%	0%
- 60cm	74	1	98.7%	1.3%
Atribut				
- Topi	72	3	96%	4%
- Kacamata	73	2	97.3%	2.7%

Hasil pengujian pada Tabel 3.3 menunjukkan bahwa pengujian pada cahaya terang mencapai tingkat keberhasilan 98.7% sehingga berdasarkan Tabel 2.1 maka masuk dalam kategori “Sangat Layak”. Pengujian pada cahaya redup mencapai tingkat keberhasilan 92% sehingga masuk kategori “Sangat Layak”. Pada pengujian posisi miring memiliki tingkat keberhasilan 54.7% dan masuk kategori “Cukup Layak”. Pengujian pada posisi lurus memiliki tingkat keberhasilan 96% sehingga masuk kategori “Sangat Layak”. Pada pengujian ekspresi tersenyum menunjukkan tingkat keberhasilan 94.7% termasuk kategori “Sangat Layak”. Pengujian pada ekspresi datar menunjukkan tingkat keberhasilan 100% termasuk kategori “Sangat Layak”. Pada pengujian jarak 30cm mencapai tingkat keberhasilan 100% termasuk kategori “Sangat Layak”. Pengujian jarak 60cm menunjukkan tingkat keberhasilan 98.7% sehingga masuk kategori “Sangat Layak”. Pada pengujian menggunakan atribut topi memiliki tingkat keberhasilan 96% sehingga masuk dalam kategori “Sangat Layak”. Pengujian menggunakan atribut kacamata mencapai tingkat keberhasilan sebesar 97.3% dan masuk kategori “Sangat Layak”.

Berdasarkan hasil pengujian yang telah dilakukan dapat ditemukan bahwa sistem ini memiliki kelemahan dalam mengenali wajah yang berada pada posisi miring. Pada kondisi tersebut tingkat keberhasilan sistem hanya mencapai 54.7% yang menunjukkan bahwa sistem masih belum optimal dalam menangani variasi sudut wajah. Keterbatasan ini dapat mempengaruhi akurasi keseluruhan sistem terutama pada situasi pengguna tidak berada dalam posisi yang ideal atau simetris sehingga perlu dilakukan pengembangan lebih lanjut untuk meningkatkan kemampuan sistem dalam mengenali wajah dari berbagai sudut pandang untuk mencapai performa yang lebih konsisten dan akurat.

Menurut Aminin (2024) sebagai salah satu *stakeholder* yang terlibat bahwa sistem presensi karyawan ini dapat membantu dalam pemantauan kehadiran karyawan secara akurat serta mendukung manajemen waktu. Sistem ini diharapkan mampu mendukung efisiensi perusahaan dalam pengaturan jadwal kerja karyawan dan meningkatkan produktivitas secara menyeluruh. Kontribusi positif pada sistem ini diharapkan dapat memberikan dampak bagi perusahaan.

BAB IV

REFLEKSI PELAKSANAAN MAGANG

4.1 Relevansi Akademik

Selama magang terdapat hubungan antara materi perkuliahan dan pengalaman praktik di lapangan. Hal ini memungkinkan penulis untuk menerapkan pengetahuan teoretis yang telah dipelajari ke dunia kerja agar memperkuat pemahaman dan meningkatkan keterampilan praktik yang diperlukan di industri.

Penulis mendapatkan kesempatan untuk menerapkan pengetahuan pada proyek seperti terlibat dalam proses pengembangan sistem dari tahap perancangan hingga implementasi. Selama magang penulis menghadapi dan menyelesaikan tantangan teknis yang tidak ditemukan pada lingkungan akademis. Pengalaman ini tidak hanya memperdalam pemahaman tentang materi yang telah dipelajari tetapi dapat membekali dengan keterampilan praktik yang sangat dibutuhkan pada dunia kerja.

Magang berfungsi sebagai jembatan dalam menghubungkan teori dengan praktik. Penulis tidak hanya memperoleh pengetahuan teoretis yang kuat tetapi pengalaman praktik yang relevan sehingga menjadi lebih siap dan kompetitif di dunia kerja. Praktik ini mencakup penerapan metodologi *prototyping*, implementasi *library* Google ML Kit dan Tensorflow Lite, dan penggunaan SQLite sebagai basisdata untuk penyimpanan data wajah.

4.1.1 Metodologi *Prototyping*

Metodologi *prototyping* merupakan model pengembangan *System Development Life Cycle* (SDLC) yang berfokus pada pembuatan model awal atau prototipe dari sistem yang akan dikembangkan. Prototipe sebagai alat visual dan interaktif yang digunakan pengembang dan pengguna untuk memahami dan mengevaluasi kebutuhan sistem sebelum pengembangan penuh dimulai. *Prototyping* memungkinkan iterasi dan umpan balik yang cepat sehingga membantu dalam mengidentifikasi dan mengatasi masalah lebih awal pada siklus pengembangan.

Pada mata kuliah Rekayasa Perangkat Lunak diajarkan tentang model pengembangan SDLC. Pengetahuan ini penting dikarenakan setiap metode memiliki pendekatan dan manfaat tersendiri yang dapat disesuaikan dengan jenis proyek yang dikerjakan. Metodologi *prototyping* mempelajari cara mengembangkan prototipe yang efektif dengan mengumpulkan dan memanfaatkan umpan balik pengguna untuk menyempurnakan sistem.

4.1.2 *Library* Google ML Kit dan Tensorflow Lite

Library Google ML Kit dan Tensorflow Lite digunakan pada pengembangan sistem presensi karyawan dalam memanfaatkan kemampuan *machine learning*. Google ML Kit digunakan untuk mengimplementasikan fitur-fitur berbasis kecerdasan buatan pada aplikasi *mobile* sedangkan Tensorflow Lite adalah versi ringan dari Tensorflow yang dirancang untuk perangkat *mobile* dan *embedded*.

Pada mata kuliah Pengembangan Aplikasi Bergerak diajarkan berbagai aspek tentang implementasi *library* dalam pengembangan aplikasi *mobile*. Meskipun implementasi *library* seperti Google ML Kit dan Tensorflow Lite tidak secara langsung diajarkan tetapi dapat memberikan landasan bagi mahasiswa dalam memahami dan mengimplementasikan *library*. Implementasi *library* Google ML Kit dan Tensorflow Lite mengajarkan tentang berbagai konsep *machine learning*, pengolahan data, dan integrasi dengan aplikasi *mobile* sehingga menjadi pengetahuan penting dalam mengembangkan aplikasi *mobile*.

4.1.3 SQLite sebagai basisdata

SQLite adalah sistem manajemen basisdata relasional yang ringan dan bersifat *self-contained* sehingga tidak memerlukan server untuk beroperasi. SQLite menjadi pilihan yang sesuai untuk aplikasi *mobile* dikarenakan efisiensi penggunaan sumber daya. Pada sistem presensi karyawan SQLite digunakan untuk menyimpan data wajah karyawan secara lokal di perangkat pengguna.

Pada mata kuliah Pengembangan Aplikasi Berbasis Web diajarkan tentang penggunaan sintaks SQL yang juga dapat diterapkan pada SQLite. Meskipun SQLite tidak diajarkan secara spesifik pada mata kuliah ini tetapi pemahaman tentang SQL sangat penting karena SQLite menggunakan sintaks yang sama dengan SQL dalam melakukan operasi basisdata. Implementasi SQLite mengajarkan tentang memahami cara kerja relasional pada pengembangan aplikasi *mobile*.

4.2 Pembelajaran Magang

Magang merupakan langkah penting bagi mahasiswa untuk meningkatkan keterampilan yang diperlukan dalam dunia kerja. Pengalaman magang tidak hanya berperan sebagai pembelajaran tetapi dapat menjadi persiapan dalam menghadapi berbagai tantangan di masa depan. Selama pelaksanaan magang, penulis mendapatkan manfaat dalam pengembangan keterampilan teknis dan non-teknis.

4.2.1 Keterampilan Teknis

Penulis menghadapi beberapa tantangan teknis selama magang terutama dalam penggunaan teknologi. Penulis belum berpengalaman menggunakan *framework* Flutter sehingga harus belajar dari awal tentang konsep dasar Flutter dan memahami berbagai teknik yang diperlukan dalam pengembangan aplikasi. Mempelajari *widget* pada Flutter sebagai elemen dasar dalam membangun tampilan aplikasi seperti teks, tombol, dan gambar serta cara mengatur tata letak menggunakan *widget* seperti baris (*row*) dan kolom (*coloumn*). Navigasi dan perpindahan antar halaman dalam aplikasi merupakan bagian penting dari pembelajaran untuk mengelola perpindahan antar halaman.

Mengembangkan keterampilan pemrograman umum seperti pemrograman berorientasi objek, memahami struktur data, dan algoritma serta prinsip-prinsip desain perangkat lunak yang baik. Mempelajari teknik-teknik dalam meningkatkan performa aplikasi Flutter seperti *lazy loading* untuk menunda pemuatan data hingga diperlukan, penggunaan *builder widgets* untuk menghindari *rebuild* yang tidak perlu dan cara meminimalisir penggunaan memori. Belajar cara mendeteksi dan memperbaiki masalah kinerja pada aplikasi dengan menggunakan *debugging* dan *profiling* yang tersedia di Flutter. Magang mencakup penggunaan berbagai teknologi yang mendukung pengembangan aplikasi seperti Firebase untuk *backend-as-a-service* dan Git untuk *version control* sehingga penulis mempelajari integrasi teknologi tersebut ke sistem.

Penulis menghadapi masalah dalam penggunaan *dependency* yang tidak kompatibel satu sama lain sehingga penulis harus memastikan bahwa semua *dependency* memiliki versi yang kompatibel dan dapat bekerja dengan baik. Hal ini memerlukan pemahaman tentang manajemen *dependency* dan setiap komponen dalam aplikasi saling berinteraksi sehingga penulis harus melakukan pembaruan dan pengujian secara menyeluruh dalam memastikan bahwa perubahan pada satu *dependency* tidak menyebabkan kerusakan atau *bug* pada bagian lain dari aplikasi. Pengembangan keterampilan teknis membuat penulis dapat menghadapi tuntutan dunia kerja yang terus berkembang dan semakin kompetitif sehingga magang memberikan landasan kuat untuk memulai karier di bidang teknologi dan pengembangan perangkat lunak.

4.2.2 Keterampilan Non Teknis

Magang tidak hanya berperan dalam meningkatkan keterampilan teknis tetapi juga keterampilan non-teknis seperti kemampuan bekerja secara tim melalui partisipasi dalam berbagai proyek sehingga mempelajari kerja sama, mendengarkan pendapat orang lain, dan berkontribusi dalam mencapai tujuan bersama. Kemampuan berkolaborasi dalam tim menjadi sangat penting pada lingkungan dunia kerja seperti yang dialami penulis dalam pengembangan sistem presensi karyawan dengan bekerja sama pengembang lainnya. Kerja tim ini melibatkan berbagai tahapan dari perencanaan hingga implementasi. Selama perencanaan, penulis bersama *stakeholder* mendiskusikan rancangan fitur pada sistem. Penulis bekerja sama dengan pengembang lainnya untuk mengimplementasikan dan mengintegrasikan rancangan tersebut ke dalam kode aplikasi.

Penulis menghadapi kesulitan dalam mengerjakan tugas karena perusahaan menerapkan sistem kerja *work from home* yang mengharuskan karyawan berkomunikasi secara *online*. Penulis merasa komunikasi secara *online* kurang nyaman dibandingkan dengan komunikasi secara langsung karena interaksi yang lebih terbatas sehingga memerlukan penyesuaian dalam menyampaikan informasi dengan jelas dan memastikan pemahaman yang sama.

Lingkungan kerja yang dinamis membantu dalam mengatur waktu dengan efisien sehingga penulis dapat mengidentifikasi dan mengutamakan tugas-tugas yang penting serta menyelesaikan pekerjaan tepat waktu. Magang dapat mengasah kemampuan dalam menghadapi berbagai tantangan. Pengalaman ini memerlukan pemecahan masalah agar tetap tenang dan memerlukan solusi yang efektif. Manfaat lain dari magang adalah pengalaman kerja langsung di lapangan, membangun jaringan profesional, peningkatan kepercayaan diri serta pemahaman yang lebih mendalam tentang industri sehingga magang memiliki banyak manfaat yang dapat membantu dalam mengembangkan diri.

BAB V PENUTUP

5.1 Kesimpulan

Hasil pengujian sistem pengenalan wajah menunjukkan kinerja yang bervariasi tergantung pada kondisi pengujian. Pengujian pada cahaya terang mencapai tingkat keberhasilan 98.7% dan pengujian pada cahaya redup mencapai tingkat keberhasilan 92%. Pada pengujian posisi miring memiliki tingkat keberhasilan 54.7% dan pengujian pada posisi lurus memiliki tingkat keberhasilan 96%. Pada pengujian ekspresi tersenyum menunjukkan tingkat keberhasilan 94.7% dan pengujian pada ekspresi datar menunjukkan tingkat keberhasilan 100%. Pada pengujian jarak 30cm mencapai tingkat keberhasilan 100% dan pengujian jarak 60cm menunjukkan tingkat keberhasilan 98.7%. Pada pengujian menggunakan atribut topi memiliki tingkat keberhasilan 96% dan pengujian menggunakan atribut kacamata mencapai tingkat keberhasilan sebesar 97.3%.

Berdasarkan hasil pengujian yang telah dilakukan dapat ditemukan bahwa sistem ini memiliki kelemahan dalam mengenali wajah yang berada pada posisi miring sehingga sistem masih belum optimal dalam menangani variasi sudut wajah. Keterbatasan ini dapat mempengaruhi akurasi keseluruhan sistem terutama pada situasi pengguna tidak berada dalam posisi yang ideal atau simetris sehingga perlu dilakukan pengembangan lebih lanjut untuk meningkatkan kemampuan sistem dalam mengenali wajah dari berbagai sudut pandang untuk mencapai performa yang lebih konsisten dan akurat.

Hasil penelitian ini diharapkan dapat memberikan landasan bagi perusahaan yang sedang mempertimbangkan pengembangan sistem presensi karyawan yang menerapkan sistem kerja jarak jauh. *Framework* Flutter memungkinkan pengembangan aplikasi yang responsif dan mudah digunakan sedangkan Tensorflow Lite dan Google ML Kit memastikan model *machine learning* dapat berjalan optimal pada perangkat *mobile*.

Perusahaan dapat memastikan bahwa proses presensi dapat berjalan dengan baik meskipun karyawan bekerja dari jarak jauh sehingga mengurangi kemungkinan kecurangan. Teknologi *face recognition* memberikan perlindungan tambahan terhadap keamanan data dan integritas sistem presensi dikarenakan hanya karyawan terdaftar yang dapat melakukan presensi dan memberikan perlindungan tambahan bagi perusahaan dalam menjaga integritas data kehadiran.

5.2 Saran

Perusahaan disarankan untuk melakukan pembaruan teknologi secara teratur. Pembaruan teknologi perlu dilakukan untuk memastikan bahwa semua sistem yang telah dikembangkan tetap aman, efisien, dan fungsional. Perusahaan dapat mengakses fitur-fitur terbaru yang dapat meningkatkan efisiensi operasional dan memberikan pengalaman yang lebih baik bagi pengguna.

Konsep *clean code* merupakan prinsip penting dalam pengembangan sistem. *Clean code* mengacu pada penulisan kode yang memprioritaskan keterbacaan, pemahaman, dan pemeliharaan kode. Penerapan *clean code* dapat mengurangi kerentanan terhadap *bug* dan mempermudah pemeliharaan sistem di masa depan. *Clean code* juga memungkinkan pengembang untuk lebih mudah memahami dan mengembangkan fitur-fitur baru sehingga mempercepat proses pengembangan.

Penerapan *clean code* dapat dilakukan dengan beberapa langkah seperti memberikan nama variabel, fungsi, dan kelas yang deskriptif serta membatasi fungsi agar tidak terlalu kompleks dan menghindari duplikasi kode. Perusahaan dapat mengoptimalkan pengembangan sistem dan meningkatkan kualitas kode yang dihasilkan dengan menerapkan prinsip *clean code*.

DAFTAR PUSTAKA

- Aditia Muclis, P., & Somantri. (2021). Implementasi Text Recogniter yang Diterjemahkan ke Bahasa Lain dengan Firebase ML-Kit Berbasis Android. *Jurnal Informatika Universitas Pamulang*, 6(2), 2622–4615. <http://openjournal.unpam.ac.id/index.php/informatika414>
- Alwanda, M. R., Ramadhan, R. P. K., & Alamsyah, D. (2020). Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle. *Jurnal Algoritme*, 1(1), 45–56. <https://doi.org/10.35957/algoritme.v1i1.434>
- Azmi, K., Defit, S., & Sumijan, S. (2023). Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat. *Jurnal Unitek*, 16(1), 28–40. <https://doi.org/10.52072/unitek.v16i1.504>
- Bhattacharyya, D., Ranjan, R., a, F. A., & Choi, M. (2009). Biometric Authentication : A Review. *International Journal of Service, Science and Technology*, 2(3), 13–28.
- Cahyono F. (2020). *Pengenalan Wajah Menggunakan Model FaceNet untuk Presensi Pegawai*.
- Chihaoui, M., Elkefi, A., Bellil, W., & Amar, C. Ben. (2016). A survey of 2D face recognition techniques. *Computers*, 5(4), 1–28. <https://doi.org/10.3390/computers5040021>
- Christyanto, N. E., Muhammad, E., Jonemaro, A., & Yudistira, N. (2022). *Pengembangan Aplikasi Android Presensi Kehadiran Realtime menggunakan Pengenalan Wajah dengan Model Facenet*. 6(10), 4839–4847.
- Efanntyo, & Mitra, A. R. (2021). *Perancangan Aplikasi Sistem Pengenalan Wajah Dengan Metode Convolutional Neural Network (CNN) Untuk Pencatatan Kehadiran Karyawan*. 3(1), 1–11.
- Evelyn, E., Adipranata, R., & Gunadi, K. (2022). *Sistem Presensi Mahasiswa Menggunakan Face Recognition Dengan Metode Facenet Pada Android*. 2–8.
- Fachriyan, M., & Dharmayanti, D. (n.d.). *PEMBANGUNAN APLIKASI PENGENALAN OBJEK TERDEKAT UNTUK PENYANDANG TUNANETRA MENGGUNAKAN MLKIT DAN TEXT TO VOICE BERBASIS ANDROID*.
- Ferdiawan, F., & Hartono, B. (2022). *DETEKSI SUARA CHORD PIANO MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK*. 5(1), 62–68.
- Fuady, S., Nehru, N., & Anggraeni, G. (2020). Deteksi Objek Menggunakan Metode Single Shot Multibox Detector Pada Alat Bantu Tongkat Tunanetra Berbasis Kamera. *Journal of Electrical Power Control and Automation (JEPCA)*, 3(2), 39.

<https://doi.org/10.33087/jepca.v3i2.38>

- Giovanni, A., Indrasari, W., & Firmasyah, H. (2023). *Pendeteksi Wajah Sebagai Sebuah Sistem Keamanan Ruangan*. *XI*, 33–38. <https://doi.org/10.21009/03.1101.fa06>
- Haniefardy, A., Bayu, M., Fadhillah, A., & Rochimah, S. (2019). *Tinjauan Literatur Sistematis : Pengaruh Penggunaan Framework Khusus dalam Proses Pengembangan dan Pembuatan Web*. *9*(2).
- Hary Prasetyo. (2017). *PROTOTYPING UNTUK ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM MONITORING MENTORING (STUDI KASUS: PROGRAM ASISTENSI AGAMA ISLAM FAKULTAS TEKNOLOGI INDUSTRI UNIVERSITAS ISLAM INDONESIA)*. *4*(1), 9–15. <https://dspace.uui.ac.id/handle/123456789/10072>
- Husain, A., Prastian, A. H. A., & Ramadhan, A. (2017). *Perancangan Sistem Absensi Online Menggunakan Android Guna Mempercepat Proses Kehadiran Karyawan Pada PT. Sintech Berkah Abadi*. *2*(1), 105–116.
- Lalu Fiqihilmi, Lalu A.Syamsul Irfan Akbar, Dwi Ratnasari, & Cipta Ramadhani. (2023). Implementasi Pengenalan Wajah Pada Aplikasi Presensi Perkuliahan Menggunakan FaceNet Berbasis Android . *JEITECH – Journal of Electrical Engineering and Information Technology* , *1*(1), 14–18.
- Mantara, D., Sudoro, W., Kurniasari, A., & Rosid, J. (2022). *Face recognition dengan metode Haar Cascade dan Facenet*. *3*(1), 30–34.
- Miftahuddin, Y., Umaroh, S., & Karim, F. R. (2020). Perbandingan Metode Perhitungan Jarak Euclidean, Haversine, Dan Manhattan Dalam Penentuan Posisi Karyawan. *Jurnal Tekno Insentif*, *14*(2), 69–77. <https://doi.org/10.36787/jti.v14i2.270>
- Muhaimin, M. S. (2013). Rancang Bangun Aplikasi Multi Face Detector Menggunakan Metode Viola Jones Pada Face Recognition. *Tugas Akhir*, 131.
- Prabu, A. S., & Wijayanti, D. T. (2016). Pengaruh Penghargaan dan Motivasi Terhadap Kinerja Karyawan (Studi Pada Divisi Penjualan PT. United Motors Center Suzuki Ahmad Yani, Surabaya). *Jurnal Ekonomi Bisnis Dan Kewirausahaan*, *5*(2), 104. <https://doi.org/10.26418/jebik.v5i2.17144>
- Puspita sari, P. aggraeny, Fajar, M., & Arianti. (2023). Perancangan Ui/Ux Pada Website Medisol Dengan Metode User Centered Design. *KHARISMA Tech*, *18*(1), 40–54. <https://doi.org/10.55645/kharismatech.v18i1.284>
- Ramdhon, A. N., & Febriya, F. (2021). *Penerapan Face Recognition Pada Sistem Presensi*. *2*(1), 12–17.

- Ravy Pramestya Hayu. (2018). *Deteksi Kerusakan Jalan Menggunakan Convolutional Neural Network Berbasis SSD EfficientNet* Artikel Ilmiah. 672016033, 1–91.
- Roslidar, R., Syahputra, M. R., Muharar, R., & Arnia, F. (2022). Adaptasi Model CNN Terlatih pada Aplikasi Bergerak untuk Klasifikasi Citra Termal Payudara. *Jurnal Rekayasa Elektrika*, 18(3), 185–192. <https://doi.org/10.17529/jre.v18i3.8754>
- Salamun, & Wazir, F. (2016). *RANCANG BANGUN SISTEM PENGENALAN WAJAH DENGAN METODE PRINCIPAL COMPONENT ANALYSIS*. 1(2), 59–75.
- Samodra, M. R., & Noermijati. (2016). Pengaruh penilaian kinerja karyawan terhadap motivasi kerja dan kinerja karyawan di PT. Telkomsel branch malang. *Jurnal Aplikasi Bisnis Dan Manajemen*, 2(3), 248–257.
- Sejati, R. H. P., & Mardhiyyah, R. (2021). *DETEKSI WAJAH BERBASIS FACIAL LANDMARK MENGGUNAKAN OPENCV DAN DLIB*. 5(2), 144–148.
- Setiawan, R., Nurcahya, P. T. P., & Yuana, R. A. (2024). *Rekayasa Perangkat Lunak Aplikasi Presensi Mobile Menggunakan Metode Deep Learning*. 17(1), 36–48.
- Setiyadi, A., & Harihayati, T. (2015). penerapan SQLITE. *Penerapan Sqlite Pada Aplikasi Pengaturan Waktu Ujian Dan Presentasi*, 13(2), 221–226.
- Setyadi, H. A., & Sundari, S. (2022). Sistem Informasi Manajemen Kehadiran Dan Jam Kerja Karyawan Untuk Kelengkapan Perhitungan Gaji Karyawan. *Indonesian Journal Computer Science*, 1(1), 28–33. <https://doi.org/10.31294/ijcs.v1i1.1114>
- Sugeng, W., & Barus, D. (2023). *Pengecekan Foto Paspor Menggunakan Metode DNN dan Facenet Sebagai Pengenalan Wajah*. 169–180.
- Sulistiya, M., Mu, Z., Natasia, S. R., & Yusuf, M. (2016). *Penerapan Metode Think Aloud untuk Evaluasi Usability pada Website Dinas Pendidikan dan Kebudayaan Kota MNO*. 16(1), 25–32.
- Tjandra, S., & Chandra, G. S. (2020). Pemanfaatan Flutter dan Electron Framework pada Aplikasi Inventori dan Pengaturan Pengiriman Barang. *Journal of Information System, Graphics, Hospitality and Technology*, 2(02), 76–81. <https://doi.org/10.37823/insight.v2i02.109>
- Wardoyo, S., Wiryadinata, R., & Sagita, R. (2016). Sistem Presensi Berbasis Algoritma Eigenface Dengan Metode Principal Component Analysis. *Setrum : Sistem Kendali-Tenaga-Elektronika-Telekomunikasi-Komputer*, 3(1), 61. <https://doi.org/10.36055/setrum.v3i1.498>

LAMPIRAN

LAMPIRAN A

Berikut merupakan lampiran A yang berisi lembar evaluasi pengujian sistem presensi karyawan

LEMBAR EVALUASI PENGUJIAN SISTEM PRESENSI KARYAWAN

Lampiran : 1
Judul Tugas Akhir : Pengembangan Sistem Presensi Karyawan dengan *Face Recognition* Menggunakan *Framework Flutter*
Tanggal : 5 Januari 2024

LAMPIRAN HASIL EVALUASI

Identitas	Hasil
Nama: Aminin Posisi: Sekretaris PT Indo Xetia Internasional	Sistem presensi karyawan ini dapat membantu dalam pemantauan kehadiran karyawan secara akurat dan mendukung manajemen waktu. Sistem ini diharapkan mampu mendukung efisiensi perusahaan dalam pengaturan jadwal kerja karyawan sehingga mengurangi beban administrasi manual dan meningkatkan produktivitas secara menyeluruh. Kontribusi positif pada sistem ini diharapkan dapat memberikan dampak bagi perusahaan.

LAMPIRAN B

Berikut merupakan lampiran B yang berisi video dan foto pengujian yang dapat diakses pada tautan berikut <https://bit.ly/DokumentasiPengujian>

