

**PERANCANGAN DAN IMPLEMENTASI
SISTEM ABSENSI BERBASIS *FACE*
RECOGNITION PADA PERUSAHAAN JASA
TEKNOLOGI INFORMASI**

PROYEK AKHIR

Disusun Oleh:

Dina Ramadani

3312301014

Disusun untuk memenuhi salah satu syarat memperoleh gelar Ahli Madya
Teknik Informatika



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
2026**

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan tugas akhir yang berjudul “Perancangan dan Implementasi Sistem Absensi Berbasis *Face Recognition* pada Perusahaan Jasa Teknologi Informasi” dengan baik dan tepat waktu.

Laporan ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan pada Program Studi Teknik Informatika, Jurusan Teknik Informatika, Politeknik Negeri Batam. Penulis menyadari bahwa selesainya laporan ini tidak terlepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Ir. Bambang Hendrawan, ST., MSM., CIPMP., CISCP. selaku Direktur Politeknik Negeri Batam.
2. Bapak Ahmad Hamim Thohari, S.S.T., M.T. selaku Ketua Jurusan Teknik Informatika.
3. Ibu Yeni Rokhayati, S.Si., M.Sc. selaku Ketua Program Studi Teknik Informatika.
4. Ibu Mir’atul Khusna Mufida, Ph.D. selaku Dosen Pembimbing yang senantiasa memberikan bimbingan, dukungan, serta arahan kepada penulis selama proses pengerjaan dan penyusunan tugas akhir ini.
5. Dosen Penguji yang telah memberikan masukan, evaluasi, serta saran yang membangun selama proses sidang dan pengujian, sehingga laporan Tugas Akhir ini dapat disusun dengan lebih baik.
6. Pimpinan dan seluruh staf Perusahaan Jasa Teknologi Informasi yang telah memberikan kesempatan, fasilitas, dan ilmu praktis selama penelitian berlangsung.
7. Kedua orang tua penulis yang senantiasa memberikan dukungan.
8. Dalam proses penyusunan Tugas Akhir ini, penulis juga mendapatkan semangat dan inspirasi dari karya musik beberapa musisi seperti Charli

XCX, Lorde, The Cure, Julian Casablancas, Slowdive, The Smiths, dan Ethel Cain, yang membantu penulis dalam memahami emosi, kreativitas, serta menjaga motivasi selama proses pengerjaan.

9. Terakhir saya ucapkan kepada diri saya sendiri, karena telah bertahan sejauh ini. Tidak selalu mudah, tapi akhirnya bisa juga sampai di titik menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa laporan ini masih memiliki banyak kekurangan dan belum sepenuhnya sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun sebagai bahan evaluasi dan perbaikan di masa yang akan datang. Semoga laporan Tugas Akhir ini dapat memberikan manfaat serta menambah wawasan, khususnya dalam bidang Teknologi Informasi.

DAFTAR ISI

KATA PENGANTAR	2
DAFTAR ISI	4
DAFTAR TABEL	6
DAFTAR GAMBAR	8
DAFTAR LAMPIRAN	10
ABSTRAK	11
BAB I	12
PENDAHULUAN	12
1.1. Latar Belakang	12
1.2. Rumusan Masalah	14
1.3. Tujuan	14
1.4. Batasan Masalah	14
1.5. Manfaat	15
BAB II	16
TINJAUAN PUSTAKA	16
2.1. Penelitian Terkait	16
2.2. Landasan Teori	19
2.2.1 Pencatatan Kehadiran Karyawan	19
2.2.2 Presensi Online	19
2.2.3 <i>Face Detection</i>	19
2.2.4 <i>Face Recognition</i>	19
2.2.5 <i>Computer Vision</i>	20
2.2.6 OpenCV	20
2.2.7 <i>Haar Cascade Classifier</i>	21
2.2.8 Local Binary Pattern Histogram (LBPH)	22
2.2.9 YOLO	24
2.2.10 NestJs	25
2.2.11 VueJs	25
2.2.12 <i>Global Positioning System</i>	25
2.3. Metode Pengembangan Produk	25
BAB III	29
ANALISIS DAN PERANCANGAN	29
3.1 Analisis Kebutuhan	29
3.1.1 Proses Bisnis Berjalan	29
3.1.2 Gambaran Umum Sistem	30
3.2 Perancangan	32
3.2.1 Kebutuhan Fungsional	32

3.2.2 Kebutuhan Non-Fungsional.....	33
3.2.3 Diagram Use Case	34
3.2.4 Skenario Use Case	35
3.2.4 Activity Diagram	52
3.2.5 ER Diagram	68
3.2.5 Perancangan Antarmuka (Wireframe)	72
BAB IV	78
IMPLEMENTASI DAN PENGUJIAN	78
4.1 Hasil Implementasi	78
4.1.1 Implementasi Basis Data.....	78
4.1.2 Implementasi API.....	79
4.1.3 Implementasi Pada Sisi UI	85
4.1.4 Implementasi AI	95
4.2 Pengujian Sistem.....	102
4.2.1 User Acceptance Testing (UAT)	103
4.2.2 Load Testing	123
BAB V	127
KESIMPULAN	127
5.1 Kesimpulan.....	127
5.2 Saran	128
DAFTAR PUSTAKA	129
DAFTAR LAMPIRAN.....	132

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian Sebelumnya.....	16
Tabel 3.1 Kebutuhan Fungsional.....	32
Tabel 3.2 Kebutuhan Non-Fungsional.....	33
Tabel 3.3 Skenario <i>Use Case</i> Masuk.....	35
Tabel 3.4 Skenario <i>Use Case</i> Keluar.....	36
Tabel 3.5 Skenario <i>Use Case</i> Melihat Laporan.....	37
Tabel 3.6 Skenario <i>Use Case</i> Melihat Riwayat Kehadiran.....	37
Tabel 3.7 Skenario <i>Use Case</i> Mengelola Profil.....	38
Tabel 3.8 Skenario <i>Use Case</i> Mengubah Kata Sandi.....	39
Tabel 3.9 Skenario <i>Use Case</i> Mendapatkan Notifikasi.....	40
Tabel 3.10 Skenario <i>Use Case</i> Melakukan Absensi Cepat.....	41
Tabel 3.11 Skenario <i>Use Case</i> Melakukan Absensi.....	42
Tabel 3.12 Skenario <i>Use Case</i> Mengajukan Persetujuan.....	43
Tabel 3.13 Skenario <i>Use Case</i> Menambah Catatan Keterlambatan.....	44
Tabel 3.14 Skenario <i>Use Case</i> Daftar.....	44
Tabel 3.15 Skenario <i>Use Case</i> Mengelola Akun Pengguna.....	46
Tabel 3.16 Skenario <i>Use Case</i> Menambah <i>Dataset</i> Pengguna.....	47
Tabel 3.17 Skenario <i>Use Case</i> Menangani Persetujuan.....	48
Tabel 3.18 Skenario <i>Use Case Export</i> Laporan.....	50
Tabel 3.18 Skenario <i>Use Case</i> Mengelola Peraturan Absensi	51
Tabel 4.1 Implementasi API	80
Tabel 4.2 Implementasi API AI.....	101
Tabel 4.3 <i>User Acceptance Testing</i> FR-001.....	104
Tabel 4.4 <i>User Acceptance Testing</i> FR-002.....	105
Tabel 4.5 <i>User Acceptance Testing</i> FR-003.....	106
Tabel 4.6 <i>User Acceptance Testing</i> FR-004.....	107
Tabel 4.7 <i>User Acceptance Testing</i> FR-005.....	108
Tabel 4.8 <i>User Acceptance Testing</i> FR-006.....	109
Tabel 4.9 <i>User Acceptance Testing</i> FR-007.....	111
Tabel 4.10 <i>User Acceptance Testing</i> FR-008.....	112

Tabel 4.11 <i>User Acceptance Testing</i> FR-009.....	113
Tabel 4.12 <i>User Acceptance Testing</i> FR-010.....	114
Tabel 4.13 <i>User Acceptance Testing</i> FR-011.....	116
Tabel 4.14 <i>User Acceptance Testing</i> FR-012.....	117
Tabel 4.15 <i>User Acceptance Testing</i> FR-013.....	118
Tabel 4.16 <i>User Acceptance Testing</i> FR-014.....	120
Tabel 4.17 <i>User Acceptance Testing</i> FR-015.....	121
Tabel 4.18 <i>User Acceptance Testing</i> FR-016.....	122
Tabel 4.19 <i>User Acceptance Testing</i> FR-017.....	123
Tabel 4.20 Hasil Pengujian <i>Load Testing</i> Menggunakan <i>Locust</i>	125

DAFTAR GAMBAR

Gambar 2.1 Gambar Haar Cascade Classifier.....	22
Gambar 2.2 Proses LBP.....	23
Gambar 2.3 Proses Metode HOG.....	23
Gambar 2.4 Klasifikasi Model YOLO.....	24
Gambar 2.5 Metode <i>Waterfall</i>	27
Gambar 3.1 Proses Bisnis Berjalan.....	29
Gambar 3.2 Gambaran Umum Sistem.....	30
Gambar 3.3 Gambaran Umum <i>Face Recognition</i>	31
Gambar 3.4 <i>Use Case Diagram</i>	34
Gambar 3.5 <i>Activity Diagram</i> Masuk.....	52
Gambar 3.6 <i>Activity Diagram</i> Keluar.....	53
Gambar 3.7 <i>Activity Diagram</i> Melihat Laporan.....	53
Gambar 3.8 <i>Activity Diagram</i> Melihat Riwayat Kehadiran.....	54
Gambar 3.9 <i>Activity Diagram</i> Mengelola Profil.....	55
Gambar 3.10 <i>Activity Diagram</i> Mengubah Kata Sandi.....	56
Gambar 3.11 <i>Activity Diagram</i> Mendapatkan Notifikasi.....	57
Gambar 3.12 <i>Activity Diagram</i> Melakukan Absensi Cepat.....	58
Gambar 3.13 <i>Activity Diagram</i> Melakukan Absensi.....	59
Gambar 3.14 <i>Activity Diagram</i> Mengajukan Persetujuan.....	60
Gambar 3.15 <i>Activity Diagram</i> Menambah Catatan Keterlambatan.....	61
Gambar 3.16 <i>Activity Diagram</i> Daftar.....	62
Gambar 3.17 <i>Activity Diagram</i> Mengelola Akun Pengguna.....	63
Gambar 3.18 <i>Activity Diagram</i> Menambah <i>Dataset</i> Pengguna.....	64
Gambar 3.19 <i>Activity Diagram</i> Menangani Persetujuan.....	65
Gambar 3.20 <i>Activity Diagram</i> <i>Export</i> Laporan.....	66
Gambar 3.21 <i>Activity Diagram</i> Mengelola Peraturan Absensi.....	67
Gambar 3.22 ER Diagram.....	68
Gambar 3.23 <i>Wireframe</i> Halaman Masuk.....	72
Gambar 3.24 <i>Wireframe</i> Halaman Beranda	72
Gambar 3.25 <i>Wireframe</i> Halaman Masuk.....	73

Gambar 3.26 <i>Wireframe</i> Halaman Mengelola Profil dan Mengubah Kata Sandi..	73
Gambar 3.27 <i>Wireframe</i> Halaman Notifikasi.....	74
Gambar 3.28 <i>Wireframe</i> Halaman Absensi Cepat.....	74
Gambar 3.29 <i>Wireframe</i> Halaman Absensi.....	75
Gambar 3.30 <i>Wireframe</i> Halaman Daftar.....	75
Gambar 3.31 <i>Wireframe</i> Halaman Mengelola Pengguna.....	76
Gambar 3.32 <i>Wireframe</i> Halaman Mengelola <i>Dataset</i>	76
Gambar 3.33 <i>Wireframe</i> Halaman Menangani Persetujuan.....	76
Gambar 3.34 <i>Wireframe</i> Halaman Mengelola Peraturan Absensi.....	77
Gambar 4.1 Implementasi Basis Data.....	79
Gambar 4.2 Implementasi Halaman <i>Login</i>	85
Gambar 4.3 Implementasi Halaman <i>Register</i>	86
Gambar 4.4 Implementasi Halaman <i>Dashboard</i>	87
Gambar 4.5 Implementasi Halaman Pengaturan Akun.....	88
Gambar 4.6 Implementasi Halaman Notifikasi.....	88
Gambar 4.7 Implementasi Halaman Riwayat Absensi.....	89
Gambar 4.8 Implementasi Halaman Mengelola Pengguna.....	90
Gambar 4.9 Implementasi Halaman Mengelola <i>Dataset</i>	91
Gambar 4.10 Implementasi Halaman Menangani Persetujuan.....	91
Gambar 4.11 Implementasi Halaman Mengelola Pengaturan Absensi.....	91
Gambar 4.12 Implementasi Halaman Mengambil Absensi Cepat.....	93
Gambar 4.13 Implementasi Halaman Mengambil Absensi.....	94
Gambar 4.14 Implementasi Halaman Mengajukan Perizinan.....	95
Gambar 4.15 Arsitektur AI.....	96
Gambar 4.16 Implementasi deteksi dan pengenalan wajah secara <i>real-time</i> menggunakan metode YOLO dan LBPH	98
Gambar 4.17 Proses Pengunduhan Dataset.....	98
Gambar 4.18 Proses Pengunduhan Dataset.....	99
Gambar 4.19 Proses Training Dataset	99
Gambar 4.20 Struktur Folder <i>Dataset</i> Wajah	100
Gambar 4.21 Hasil Pengenalan Wajah Menggunakan LBPH.....	100
Gambar 4.23 Hasil Pengujian <i>Load Testing</i> Menggunakan <i>Locust</i>	124

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan Requirement.....	129
Lampiran B. Link Produk.....	129
Lampiran C. Dokumen Pengujian UAT.....	129

ABSTRAK

Absensi merupakan salah satu aspek penting dalam mendukung kedisiplinan dan evaluasi kinerja karyawan di sebuah perusahaan. Perusahaan Jasa Teknologi Informasi saat ini masih menggunakan aplikasi Slack untuk mencatat kehadiran, namun sistem tersebut memiliki kelemahan karena memungkinkan orang lain melakukan absensi atas nama karyawan yang tidak hadir. Hal ini menimbulkan masalah terkait keakuratan dan keamanan data absensi. Oleh karena itu, diperlukan sistem absensi berbasis *face recognition* yang lebih akurat dan sulit untuk dipalsukan. Untuk meningkatkan keamanan, sistem ini juga dilengkapi dengan verifikasi lokasi menggunakan *Global Positioning System* (GPS) sehingga absensi hanya dapat dilakukan di area yang telah ditentukan oleh perusahaan. Sistem ini dirancang dengan memanfaatkan algoritma *Haar Cascade* dan *Local Binary Pattern Histogram* (LBPH) pada *library* OpenCV untuk melakukan deteksi dan pengenalan wajah secara *real-time*. Pengembangan aplikasi dilakukan dengan metode *Agile Software Development* agar sistem dapat disesuaikan dengan kebutuhan pengguna secara bertahap. Aplikasi absensi ini dibangun berbasis *website* menggunakan *framework* NestJS untuk *backend* dan Vue.js untuk *frontend* dengan pendekatan *Progressive Web Application* (PWA). Hasil dari pengembangan ini berupa sistem absensi yang mampu mengidentifikasi wajah karyawan secara otomatis, memverifikasi lokasi kehadiran melalui *GPS*, menyimpan data absensi dengan akurat, serta membantu perusahaan dalam meningkatkan efisiensi dan mengurangi kecurangan pada proses absensi.

Kata Kunci: Absensi Karyawan, Face Recognition, OpenCV, Haar Cascade, LBPH, GPS

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi informasi yang semakin pesat telah memberikan dampak signifikan di berbagai sektor, termasuk dalam bidang pendataan kehadiran atau absensi. Penggunaan teknologi informasi dalam proses absensi dapat mempermudah pengelolaan kepegawaian. Absensi merupakan proses pendataan yang dilakukan untuk mencatat kehadiran seseorang dalam sebuah kegiatan, yang kemudian akan direkap menjadi laporan untuk disampaikan kepada ketua acara atau kegiatan (Sugeng & Mulyana, 2022). Sistem absensi merupakan proses pencatatan kehadiran yang dilakukan, sedangkan absensi adalah suatu kegiatan pengumpulan data guna mengetahui jumlah kehadiran setiap pekerja atau pelajar. Sistem presensi ini digunakan baik perusahaan, perkantoran, maupun dalam lingkup sekolah atau pendidikan (Wahyuni & Sulaeman, 2022).

Perusahaan Jasa Teknologi Informasi merupakan sebuah perusahaan yang telah berbadan usaha sejak tahun 2017. Perusahaan ini berfokus pada penyediaan produk dan jasa di bidang teknologi informasi. Secara umum, setiap perusahaan seharusnya sudah menerapkan sistem pencatatan kehadiran untuk memantau karyawan. Sistem absensi sendiri memiliki peranan penting tidak hanya sebagai alat untuk mencatat kehadiran, tetapi juga sebagai dasar evaluasi kinerja dan penilaian kedisiplinan. Penerapan sistem absensi yang efektif dapat membantu perusahaan meningkatkan efisiensi operasional, menjaga akurasi data kehadiran, serta mengurangi terjadinya kecurangan. Oleh karena itu, pemilihan metode absensi yang tepat menjadi faktor krusial dalam mendukung produktivitas dan profesionalisme karyawan (Sonny & Rizki, 2021).

Namun, metode absensi yang digunakan saat ini di Perusahaan Jasa Teknologi Informasi masih menggunakan aplikasi Slack dan memiliki kelemahan karena proses verifikasi hanya dilakukan dengan pengiriman foto *real-time* yang dicek manual oleh HR, sehingga masih memungkinkan terjadinya kecurangan. Oleh karena itu, dikembangkan sistem absensi karyawan berbasis *face recognition* yang lebih akurat dan sulit dipalsukan. Sistem ini juga dilengkapi dengan verifikasi

lokasi menggunakan GPS untuk mendeteksi dan mencatat posisi karyawan saat melakukan absensi, sehingga memberikan informasi tambahan bagi perusahaan dalam laporan kehadiran.

Sistem absensi dapat dikembangkan dengan memanfaatkan karakteristik bagian tubuh manusia seperti jari dan wajah. Wajah digunakan sebagai elemen yang dapat diidentifikasi oleh komputer. Pendeteksian dan pengenalan wajah merupakan teknik yang digunakan untuk mengenali wajah melalui proses komputer (Sugeng & Mulyana, 2022). Sistem *face recognition* adalah teknologi yang dapat mengenali wajah seseorang di dalam sebuah gambar digital. Metode yang digunakan sistem *face recognition* ada banyak, tapi secara garis besar bekerja dengan menyamakan fitur wajah yang ada di dalam basis data. Teknologi *face recognition* dapat menjadi solusi karena kemampuannya dalam memverifikasi kehadiran secara akurat berdasarkan data wajah karyawan sehingga hanya pemilik akun yang dapat melakukan absensi dan mengurangi kesalahan identifikasi (Evelyn et al., 2022). Kelebihan dari penggunaan *face recognition* adalah tidak perlunya kontak dari bagian tubuh untuk melakukan absen, efisien dan hasilnya akurat, multifungsi dan susah untuk dilakukan pemalsuan.

Berdasarkan uraian diatas, penelitian ini bertujuan untuk mengembangkan dan mengimplementasi sistem yang mampu membantu karyawan dalam absensi kehadiran karyawan, selain itu sistem ini juga diharapkan bisa memudahkan HR untuk memantau kehadiran dan laporan dalam memanfaatkan teknologi yang ada. Dengan menggunakan *face recognition*, sistem ini diharapkan mampu meningkatkan akurasi absensi, mengurangi potensi kecurangan, serta mendukung efektivitas manajemen sumber daya manusia di Perusahaan Jasa Teknologi Informasi. Peneliti membuat suatu perancangan absensi yang dinilai lebih efektif dan bermanfaat. Sistem ini diharapkan mampu meminimalisasikan kesalahan dalam proses identifikasi melalui penerapan metode *face recognition* berbasis *OpenCV* dengan memanfaatkan *Haar Cascade Classifier* dan *Local Binary Patterns Histograms* (LBPH) (Prof. Dr. Paul Mccullagh, 2023). Selain itu, sistem juga dilengkapi dengan fitur GPS untuk mendeteksi dan mencatat lokasi karyawan pada saat melakukan absensi, sehingga dapat memberikan informasi tambahan dalam laporan kehadiran.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat dirumuskan permasalahan yaitu bagaimana merancang dan membangun sistem absensi menggunakan teknologi *face recognition* berbasis *website* di Perusahaan Jasa Teknologi Informasi?

1.3. Tujuan

Tujuan umum dari penelitian ini adalah untuk membangun sistem absensi menggunakan teknologi *face recognition* berbasis *website* di Perusahaan Jasa Teknologi Informasi.

1.4. Batasan Masalah

Untuk memastikan analisis ini terfokus dan relevan, beberapa batasan masalah ditetapkan untuk memberikan fokus dan arah yang jelas. Berikut adalah batasan yang ditetapkan dalam penelitian ini:

1. Sistem dibangun untuk mendeteksi kehadiran karyawan di Perusahaan Jasa Teknologi Informasi.
2. Absensi hanya dapat dilakukan oleh karyawan Perusahaan Jasa Teknologi Informasi.
3. Pengumpulan *dataset* wajah hanya dilakukan oleh karyawan Perusahaan Jasa Teknologi Informasi.
4. Sistem menggunakan kamera perangkat pengguna untuk menangkap gambar.
5. Sistem dibangun hanya untuk melakukan absensi dan mengajukan perizinan, tidak mencakup *payroll* dan pemotongan gaji.
6. Perangkat yang digunakan untuk mengakses sistem minimal mendukung *web browser*.
7. Sistem dibangun hanya berbasis *website* dengan pendekatan *Progressive Web Application* (PWA).
8. Sistem hanya dapat mengenali 1 wajah dalam gambar saat pengguna melakukan absensi.

9. Sistem dibangun menggunakan *library* OpenCV, dan YOLO untuk membantu deteksi wajah, pengenalan wajah dan ekstraksi fitur unik pada wajah.
10. Sistem dilengkapi dengan fitur *Global Positioning System* (GPS) yang digunakan untuk mendeteksi dan mencatat lokasi karyawan saat melakukan absensi, namun tidak membatasi area tertentu.

1.5. Manfaat

Manfaat penelitian ini diharapkan memberikan kontribusi signifikan dalam beberapa aspek berikut:

1. Manfaat bagi perusahaan, penelitian ini diharapkan dapat membantu perusahaan dalam meningkatkan akurasi pencatatan kehadiran karyawan melalui absensi berbasis *face recognition*. Dengan adanya sistem ini, potensi kecurangan absensi dapat diminimalkan, efisiensi operasional meningkat, serta manajemen sumber daya manusia lebih efektif.
2. Manfaat bagi peneliti, penelitian ini memberikan pengalaman langsung dalam merancang dan mengimplementasikan sistem absensi berbasis teknologi *face recognition*. Selain itu penelitian ini memperluas pemahaman mengenai pengembangan perangkat lunak.
3. Manfaat bagi karyawan, sistem ini memberikan kemudahan bagi karyawan dalam melakukan absensi. Dengan menggunakan *face recognition*, proses absensi menjadi lebih praktis.

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Beberapa penelitian yang berhubungan dengan penelitian ini dapat dilihat pada Tabel 1. berikut ini.

Tabel 2.1 Perbandingan Penelitian Sebelumnya

Peneliti/Tahun	Judul/Topik Penelitian	Metode/Teknologi	Kelebihan	Kekurangan/Celah
(Prasetia et al., 2024)	Pengembangan Aplikasi Presensi Online Berbasis Mobile dengan Penerapan Geolocator dan Face Recognition pada CV. Global Mandiri	Metode Waterfall, Dart, framework Flutter, Geolocator, Google ML Kit, TensorFlow Lite	Integrasi geolocator dan face recognition meningkatkan keakuratan presensi dan berbasis mobile.	Hanya diuji secara fungsional belum ada pengujian akurasi face recognition di berbagai kondisi
(Putra et al., 2021)	Absensi Pengenalan Wajah Menggunakan Algoritma Eigenface Berbasis Web	Algoritma Eigenface (PCA), OpenCV, PHP & HTML, MySQL	Solusi relevan untuk pandemi menggantikan fingerprint dengan face recognition agar bebas kontak fisik, implementasi berbasis web dan terdapat beberapa sample dataset wajah yang diuji	Pengujian hanya dilakukan pada skala kecil, tampilan website kurang menarik dan akurasi sangat dipengaruhi oleh jarak kamera dan pencahayaan

(Hidayanto & Fauzi, 2025)	Implementasi Aplikasi E-Kehadiran Berbasis Face Recognition dan GPS Menggunakan Metode Agile Development	Agile, GPS, PHP, MySQL,	Integritas face recognition dan GPS meningkatkan keamanan	Tidak disebutkan model face recognition yang digunakan.
(Santoso & Kristianto, 2020)	Implementasi Penggunaan OpenCV pada Face Recognition untuk Sistem Presensi Perkuliahan Mahasiswa	OpenCV, Haar Cascade Classifier, Local Binary Patterns Histograms (LBPH).	Dalam 1 frame dapat mengenali 2 wajah,	Batasan jarak jangkauan optimal Face Recognition sampai 150 cm, sistem gagal mendeteksi jika wajah tertutup sebagian dan Hanya berbasis aplikasi lokal dengan webcam, belum diintegrasikan dengan sistem berbasis web.
(Dina Ramadani, 2025)	Perancangan dan Implementasi Sistem Absensi Berbasis Face Recognition pada Perusahaan Jasa Teknologi Informasi	Waterfall, MySQL, OpenCV, YOLOv8, Algoritma Haar Cascade Classifier, Local Binary Patterns Histograms (LBPH), GPS	Integritas face recognition dan GPS meningkatkan keamanan, sistem lebih fleksibel karena berbasis website, tidak terbatas	Ruang lingkup hanya pada Perusahaan Jasa Teknologi Informasi, belum diuji pada skala besar

			pada satu perangkat	
--	--	--	---------------------	--

Dengan melihat rangkuman dari beberapa penelitian sebelumnya, penelitian ini akan mengimplementasikan metode *face recognition* berbasis algoritma *Haar Cascade* dan *Local Binary Pattern Histogram (LBPH)* dengan dukungan *library* OpenCV untuk membangun sistem absensi karyawan pada Perusahaan Jasa Teknologi Informasi. Dari penelitian (Prasetia et al., 2024), integrasi *geolocator* dan *face recognition* terbukti meningkatkan keakuratan presensi, namun pengujiannya masih terbatas secara fungsional. Penelitian (Putra et al., 2021) menggunakan algoritma *Eigenface* dengan basis web yang relevan untuk menggantikan fingerprint, tetapi akurasi sangat dipengaruhi oleh kondisi pencahayaan dan jarak kamera. Sementara itu, penelitian (Hidayanto & Fauzi, 2025) berhasil mengintegrasikan *face recognition* dengan GPS menggunakan metode *Waterfall*, tetapi tidak menyebutkan model pengenalan wajah yang digunakan sehingga kurang jelas dari sisi teknis. Penelitian (Santoso & Kristianto, 2020) menunjukkan bahwa metode *Haar Cascade* dan LBPH mampu mendeteksi wajah secara *real-time* pada sistem presensi mahasiswa, namun jangkauan deteksinya terbatas dan belum diintegrasikan ke platform berbasis *web*.

Berdasarkan celah dari penelitian-penelitian tersebut, penelitian ini berkontribusi dengan membangun sistem absensi berbasis web menggunakan pendekatan *Progressive Web Application (PWA)* yang dilengkapi algoritma *Haar Cascade* untuk deteksi wajah dan LBPH untuk pengenalan wajah. Selain itu, metode *Agile* diterapkan agar sistem dapat dikembangkan secara fleksibel sesuai kebutuhan pengguna. Sebagai tambahan, sistem ini juga dilengkapi dengan fitur GPS untuk mendeteksi dan mencatat lokasi saat absensi dilakukan, sehingga memberikan informasi yang lebih lengkap dalam laporan kehadiran. Dengan pendekatan ini, diharapkan sistem absensi yang dihasilkan lebih akurat, efisien, serta mampu mengurangi potensi kecurangan dalam proses pencatatan kehadiran karyawan.

2.2. Landasan Teori

Adapun landasan teori dasar yang mendukung penelitian ini adalah sebagai berikut:

2.2.1 Pencatatan Kehadiran Karyawan

Kehadiran merupakan salah satu cara untuk melacak waktu kedatangan dan kepulangan seseorang. Presensi umumnya diterapkan di lingkungan resmi, seperti sekolah maupun tempat kerja. Untuk mencatat statistik sekaligus menilai tingkat kedisiplinan individu (Sonny Sonny, 2021).

2.2.2 Presensi Online

Presensi *online* merupakan sistem pencatatan kehadiran berbasis internet yang dirancang untuk menggantikan metode tradisional seperti daftar hadir manual atau mesin absensi fisik (Andre & Suciadi, 2022). Sistem ini menawarkan pencatatan kehadiran yang lebih efisien, akurat, serta dapat diakses secara fleksibel melalui jaringan internet. Dalam implementasinya, presensi *online* dapat dikembangkan melalui aplikasi khusus yang mengintegrasikan teknologi pengenalan wajah untuk autentikasi identitas (Ukamaka Betrand et al., 2023).

2.2.3 Face Detection

Proses *face detection* bekerja dengan cara menganalisis gambar yang dimasukkan untuk menentukan apakah di dalamnya terdapat wajah atau tidak. Jika wajah terdeteksi, sistem akan melakukan pemisahan dengan memotong area wajah dari latar belakang gambar. Apabila data masukan berupa video, sistem tidak hanya mendeteksi tetapi juga melakukan *face tracking* untuk mengikuti pergerakan wajah pada setiap *frame*. Secara umum, *face detection* dan *face tracking* memiliki tujuan yang sama, yaitu mengenali keberadaan wajah, namun keduanya berbeda pada metode pendeteksiannya (Susim & Darujati, 2021).

2.2.4 Face Recognition

Face recognition merupakan teknologi yang digunakan untuk mengidentifikasi atau memverifikasi seseorang melalui gambar maupun video (Yusuf et al., 2016). Teknologi ini bekerja dengan algoritma khusus

untuk mendeteksi wajah, kemudian mencocokkannya dengan data wajah yang telah tersimpan dalam basis data. Saat ini, *face recognition* banyak diterapkan dalam berbagai bidang, seperti keamanan, sistem pengawasan, manajemen akses, hingga aplikasi sosial. Secara umum, sistem *face recognition* melibatkan empat tahap utama, yaitu deteksi, penyesuaian, ekstraksi fitur, dan perbandingan (Oloyede et al., 2020).

Face recognition dilakukan dengan mengekstraksi fitur-fitur khas wajah seperti mata, hidung, dan mulut untuk membedakan wajah dari objek lain. Sistem *face recognition* bekerja dengan mengekstraksi fitur-fitur struktural tersebut, kemudian menggunakan *classifier* untuk membedakan antara objek wajah dan non-wajah (Oloyede et al., 2020).

2.2.5 Computer Vision

Computer vision merupakan salah satu cabang dari ilmu komputer yang berfokus pada penggunaan komputer untuk mengenali dan mengklasifikasikan isyarat visual, dengan tujuan memperoleh informasi dari gambar digital. Selain pada gambar, *computer vision* juga dapat diterapkan pada video, karena pada dasarnya video merupakan rangkaian gambar yang ditampilkan secara berurutan. Terdapat berbagai pendekatan dalam *computer vision*, mulai dari metode berbasis model matematis hingga teknik modern yang memanfaatkan *machine learning*. Pada era modern, *computer vision* telah dilakukan secara luas, mulai dari pengenalan dan klasifikasi objek seperti tanaman dan hewan, hingga penerapan pada teknologi kendaraan otonom. Salah satu penetapan penting dari *computer vision* adalah pada sistem pengenalan wajah (*face recognition*), yang memungkinkan komputer untuk mengidentifikasi dan memverifikasi identitas seseorang berdasarkan gambar wajah (Suwarno & Kevin, 2020).

2.2.6 OpenCV

OpenCV merupakan singkatan dari *open source computer vision*. Arsitekturnya terdiri dari perangkat lunak, basis data dan *plugin* yang telah diprogram sebelumnya dengan dukungan untuk integrasi *computer vision* (Adusumalli et al., 2021). Pendeteksi wajah dengan *Python* dapat dilakukan dengan menggunakan *library* OpenCV. Metode yang diterapkan untuk

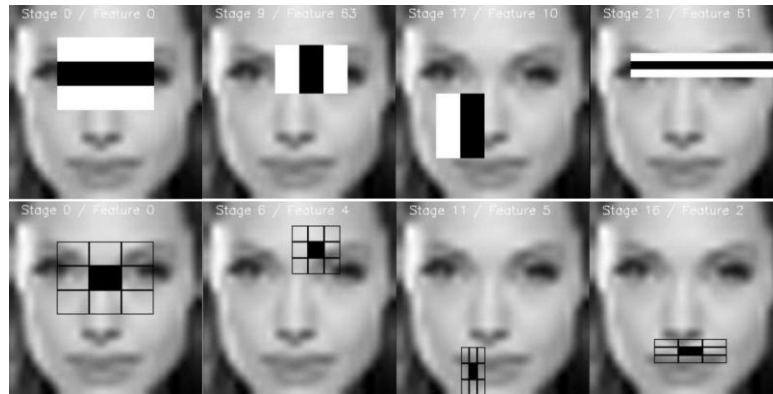
deteksi objek pada *library* OpenCV adalah metode *Convolutional Neural Networks* (CNN). (Boulila et al., 2021).

2.2.7 *Haar Cascade Classifier*

Algoritma haar cascade classifier digunakan untuk melakukan proses pendeteksian wajah atau objek yang berupa gambar digital. Algoritma ini menampilkan fungsi matematika yang berupa kotak dengan menampilkan nilai RGB pada setiap *pixel*, setelah itu Viola-Jones mengembangkan algoritma ini, di mana setiap kotak diproses dan menghasilkan beberapa nilai yang berupa daerah gelap dan terang. Dan nilai-nilai tersebut akan dijadikan sebagai dasar dalam pemrosesan gambar, sehingga dikenal dengan *haar-like feature* (Santoso & Kristianto, 2020).

Proses perhitungan nilai fitur pada algoritma *haar* dilakukan dengan cara mengurangi nilai *pixel* pada area berwarna putih dengan area berwarna hitam. *Algoritma* ini memanfaatkan *integral image* dari sebuah citra dalam bentuk *grayscale*, di mana setiap nilai *pixel* diperoleh dari penjumlahan kumulatif *pixel* mulai dari sudut kiri atas hingga kanan bawah citra (Santoso & Kristianto, 2020).

Haar cascade classifier dilatih menggunakan kumpulan data yang besar terdiri dari *positive images* dan *negative images*. *Positive image* berisi objek yang memang ingin dideteksi oleh algoritma. Sedangkan *negative images* berisi objek lain atau elemen latar belakang. Selama proses pelatihan, algoritma membangun sebuah model yang mampu membedakan antara citra positif dan citra negatif. Setelah dilatih, *haar cascade classifier* dapat mendeteksi objek pada citra baru dengan membandingkan fitur objek tersebut dengan model yang sudah dimilikinya. Algoritma ini bekerja dengan cara memindai citra menggunakan *sliding window* pada berbagai ukuran dan posisi. Untuk setiap *sliding window*, algoritma akan membandingkan fitur objek dengan model yang telah dilatih. Jika fitur yang ditemukan sesuai dengan model, maka algoritma memberikan hasil deteksi positif (Prof. Dr. Paul Mccullagh, 2023).



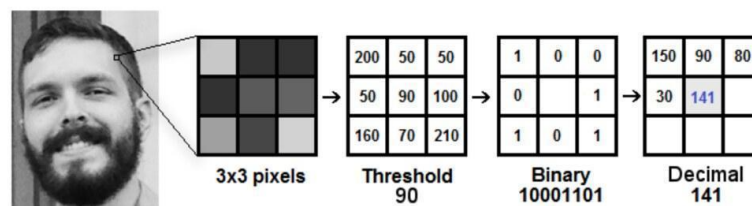
Gambar 2.1 Haar cascade classifier (Hasan & Sallow, 2021)

Gambar di atas menunjukkan beberapa contoh *haar-like feature* yang berperan dalam proses deteksi wajah, di mana kotak hitam dan putih mewakili area perhitungan nilai *pixel*.

2.2.8 Local Binary Pattern Histogram (LBPH)

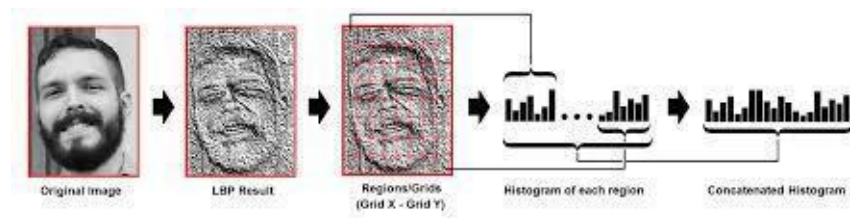
Local binary pattern histogram merupakan fitur yang berfungsi untuk mengklasifikasikan citra dengan menggabungkan pola biner lokal dan histogram, yang bertujuan untuk meningkatkan kinerja pengenalan wajah. Metode ini bekerja dengan menganalisis tekstur wajah dan menghasilkan histogram pola biner lokal sebagai dasar untuk mengenali wajah. LBPH merupakan salah satu dari tiga metode pengenalan wajah yang tersedia dalam *library* OpenCV (Muhammad, M., & Maulana, H 2023).

Metode ini sering digunakan dalam *computer vision*, pengolahan citra, dan pengenalan pola, karena kinerjanya yang baik pada gambar beresolusi rendah dengan hanya mengandalkan pola lokal. Algoritma ini mendeteksi pola tekstur dengan membandingkan setiap *pixel* dengan *pixel* di sekitarnya, kemudian menghitung histogram untuk mencocokkan wajah yang terdeteksi kamera dengan wajah dalam basis data.



Gambar 2.2 Proses LBP (Yulianti et al., 2022)

Pada metode *local binary pattern histogram* (LBPH), citra terlebih dahulu diproses dengan mengubah citra berwarna (RGB) menjadi citra skala abu-abu (*grayscale*). Pada citra *grayscale*, setiap *pixel* hanya memiliki satu nilai intensitas dengan rentang 0 sampai 255, di mana semakin mendekati 255 berarti *pixel* semakin terang. Selanjutnya, nilai intensitas *pixel* dibandingkan dengan nilai tengah (*center pixel*). Jika nilai *pixel* lebih besar atau sama dengan nilai tengah, maka diberi kode biner 1 sebaliknya jika nilai *pixel* lebih kecil, diberi kode biner 0. Hasil perbandingan dari *pixel-pixel* ini sekelilingnya membentuk sebuah deret biner. Deret biner tersebut kemudian dikonversi menjadi nilai desimal yang digunakan sebagai nilai baru bagi *pixel* tengah. Deret biner ini dikenal sebagai kode LBP (Sunardi et al., 2022).



Gambar 2.3 Proses Metode HOG (Yulianti et al., 2022)

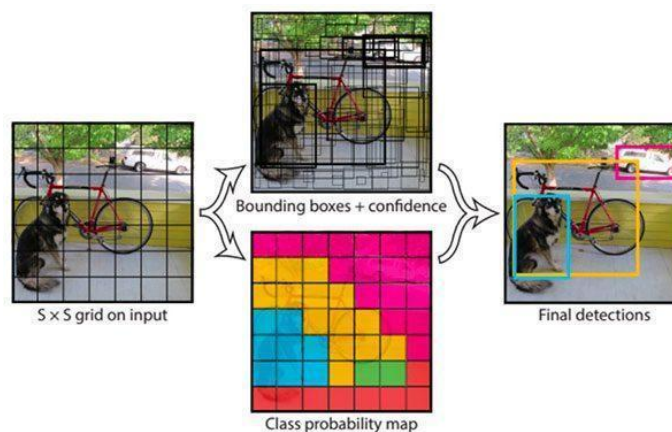
Setelah diperoleh citra biner lokal, langkah berikutnya adalah melakukan ekstraksi fitur menggunakan metode *Histogram of Oriented Gradients* (HOG). Metode ini menganalisis distribusi arah dan magnitudo gradien pada citra untuk menangkap informasi bentuk serta tekstur. Citra dibagi menjadi sel (kelompok *pixel* kecil) yang kemudian digabung menjadi blok. Setiap blok memiliki sejumlah bin orientasi yang merepresentasikan arah gradien. Parameter seperti ukuran sel, blok, dan jumlah bin sangat mempengaruhi kualitas vektor fitur dan tingkat akurasi pengenalan wajah (Pradista, R., Darmanto, T., & Putra, A. Y. A. 2024).

2.2.9 YOLO

YOLO singkatan dari *you only look once* merupakan salah satu sistem deteksi objek *real-time* terkini yang menggunakan jaringan saraf (*neural network*) untuk memproses keseluruhan gambar (Gupta et al., 2020). Jaringan ini membagi gambar ke dalam beberapa bagian kemudian

melakukan prediksi *bounding box* beserta probabilitas dari setiap objek yang terdeteksi. Nilai probabilitas tersebut digunakan untuk menimbang dan memperbaiki hasil prediksi *bounding box*.

Berbeda dengan metode konvensional YOLO menganalisis gambar secara menyeluruh pada tahap pengujian, sehingga prediksi yang dihasilkan mempertimbangkan konteks keseluruhan dari gambar. Dengan pendekatan ini YOLO mampu mendeteksi objek baik pada gambar statis maupun pada video secara efisien (Das et al., 2020).



Gambar 2.4 Klasifikasi Model YOLO (Hasan & Sallow, 2021)

YOLO secara khusus menghasilkan koordinat *bounding box* dan probabilitas untuk setiap kelas melalui pendekatan regresi langsung. Hal ini secara signifikan meningkatkan kecepatan deteksi dibandingkan dengan *faster R-CNN*. Pada data *training*, setiap citra dibagi menjadi grid berukuran $S \times S$ menggunakan jaringan YOLO. Ketika pusat objek (*ground truth center*) berada pada suatu *grid* tertentu, maka *grid* tersebut bertanggung jawab untuk mendeteksi objek tersebut. Selain itu, setiap *grid* juga memprediksi *bounding box*, probabilitas kondisional, serta tingkat kepercayaan (*confidence score*) untuk *bounding box* yang dihasilkan.

2.2.10 NestJs

NestJS adalah sebuah *framework* pemrograman yang terus dikembangkan dan ditingkatkan, serta terutama berbasis pada bahasa TypeScript. Dengan menggunakan *framework* ini, dapat melakukan pemrograman menggunakan bahasa JavaScript. *Framework* ini digunakan

untuk membangun aplikasi sisi server. Kode yang ditulis dalam *framework* ini merupakan kombinasi dari *Object-Oriented Programming* (OOP), *Functional Programming* (FP) dan *Functional Reactive Programming* (FRP). (Golec & Plechawska-Wójcik, 2022)

2.2.11 VueJs

Vue Js yang umumnya disebut sebagai Vue, adalah *framework JavaScript progressive* dan *Open Source* yang dibuat oleh Evan You pada tahun 2014 sebagai alternatif dari *framework* yang lebih berat seperti AngularJs dan React. Vue menggabungkan pendekatan yang dipengaruhi oleh *Angular* serta fitur-fitur yang lebih ringkas untuk antarmuka *frontend* dan pengembangan aplikasi. *Core library* Vue berfokus hanya pada lapisan tampilan (*view layer*) dan dirancang agar dapat diadopsi ke dalam proyek secara bertahap (Aryputra et al., 2020).

2.2.12 Global Positioning System

GPS adalah sistem navigasi berbasis satelit yang digunakan untuk menentukan lokasi geografis secara presisi. GPS bekerja dengan menerima sinyal dari jaringan satelit yang mengorbit bumi dan menghitung posisi di permukaan bumi berdasarkan jarak dari satelit-satelit tersebut (Amrial Khoir & Yudhana, 2020).

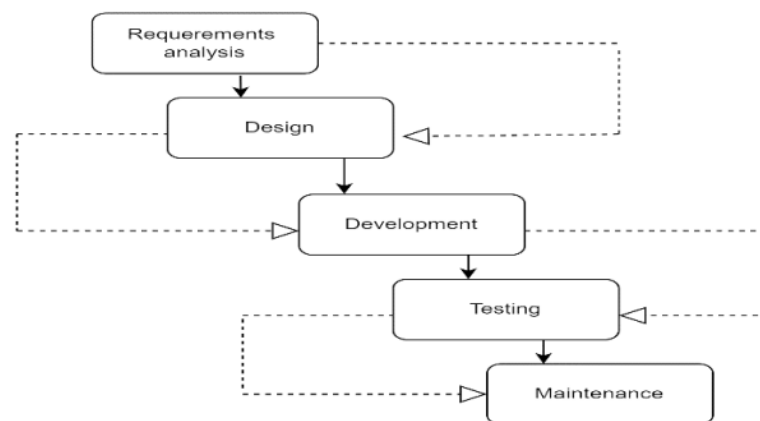
2.3. Metode Pengembangan Produk

Dalam proses pengembangan perangkat lunak, diperlukan suatu metode yang dikenal dengan *Software Development Life Cycle* (SDLC). Menurut penelitian yang dilakukan oleh (Nova et al., 2022) dalam jurnal berjudul “Analisis Metode Agile pada Pengembangan Sistem Informasi Berbasis Website: *Systematic Literature Review*”, SDLC merupakan kerangka kerja yang digunakan untuk merancang (*planning*), menganalisis (*analysis*), mendesain (*design*), mengimplementasi (*implementation*), uji coba (*testing*), dan memelihara (*maintenance*) perangkat lunak maupun sistem informasi. SDLC memiliki berbagai pendekatan yang dapat digunakan dalam proses pengembangan perangkat lunak, diantaranya *waterfall*, *scrum*, *agile*, dan *spiral*.

Metode *waterfall* merupakan salah satu model pengembangan perangkat lunak yang paling banyak digunakan. Model ini bersifat linier, dimulai dari tahap

perencanaan hingga pemeliharaan, di mana setiap tahapan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya, dan tidak memungkinkan untuk kembali ke tahap sebelumnya (Pricillia & Zulfachmi, 2021). Metode *waterfall* sering digunakan karena relatif mudah diterapkan dan memiliki alur proses yang berkelanjutan. Hal ini ditunjukkan pada penelitian yang dilakukan oleh (Agustini & Dewi, 2022) dengan judul “Perancangan Sistem Informasi Berbasis Web Menggunakan *Waterfall* Model (Studi Kasus : LPD Desa Adat Sumerta)” dan juga penelitian oleh (Septyanto et al., 2020) berjudul “Pengembangan E-Learning Berbasis Website menggunakan *Metode Waterfall*”.

Metode *waterfall* dipilih dalam penelitian ini karena proses pengembangan sistem dilakukan secara terstruktur dan memiliki tahapan yang jelas mulai dari analisis kebutuhan, perancangan sistem, implementasi, pengujian, hingga pemeliharaan. Model *waterfall* cocok digunakan karena kebutuhan sistem sudah dapat didefinisikan sejak awal sehingga setiap tahapan dapat dilakukan secara berurutan tanpa banyak perubahan selama proses pengembangan. Selain itu, metode ini memudahkan peneliti dalam melakukan dokumentasi pada setiap tahap pengembangan sehingga proses pengerjaan menjadi lebih terarah dan sistematis. (Pricillia & Zulfachmi, 2021)



Gambar 2.5 Metode Waterfall (Novia Ramadani et al., n.d.)

Tahapan dalam siklus metode *waterfall* adalah sebagai berikut :

1. *Requirements Analysis* : Sebelum proses implementasi perangkat lunak dilakukan, pengembang perlu memahami terlebih dahulu kebutuhan informasi dari pengguna sistem. Untuk memperoleh informasi tersebut,

proses pengumpulan data dapat dilakukan melalui beberapa metode seperti observasi, diskusi, survei, maupun wawancara. Pada penelitian ini, data utama diperoleh melalui proses wawancara dengan pihak terkait, kemudian hasil wawancara tersebut diolah dan dianalisis sehingga menghasilkan informasi yang lengkap mengenai kebutuhan pengguna terhadap sistem yang akan dikembangkan. Informasi tersebut selanjutnya digunakan sebagai dasar dalam pengembangan sistem.

2. *Design* : Proses perancangan sistem berdasarkan hasil analisis kebutuhan yang telah dilakukan pada tahap sebelumnya dalam metode waterfall. Pada tahap ini, seluruh kebutuhan pengguna diterjemahkan ke dalam bentuk rancangan sistem yang lebih terstruktur sebagai acuan pada proses pengembangan. Perancangan sistem meliputi pembuatan *Unified Modelling Language* (UML), perancangan basis data (*database*), serta desain antarmuka pengguna (*user interface*) agar sistem yang dikembangkan sesuai dengan kebutuhan pengguna dan mudah digunakan.
3. *Development* : Setelah tahap perancangan selesai dilakukan, tahapan berikutnya dalam metode *waterfall* adalah *development* sistem atau proses pembuatan kode program (*coding*). Pada tahap ini, rancangan sistem yang telah dibuat sebelumnya diterjemahkan ke dalam bahasa pemrograman sehingga dapat dijalankan sebagai sebuah aplikasi atau *website*. Proses implementasi dilakukan dengan mengembangkan fitur-fitur sistem sesuai dengan desain yang telah dirancang, mulai dari pembuatan antarmuka, pengelolaan *database*, hingga fungsi utama sistem. Tahap implementasi juga mencakup proses pengujian awal terhadap sistem untuk memastikan seluruh fungsi berjalan sesuai dengan kebutuhan dan rancangan yang telah ditentukan sebelumnya. Selain itu, dilakukan pengecekan terhadap kemungkinan terjadinya kesalahan (*error*) agar sistem dapat berjalan dengan baik sebelum digunakan oleh pengguna.
4. *Testing* : *Testing* pada tahapan ini pengembang akan menguji sistem yang telah dibangun sebelum digunakan oleh pengguna. Pada tahap ini

peneliti akan melakukan uji unit terhadap model *Face Detection*, *Face Recognition*, serta API backend NestJS. Selanjutnya dilakukan uji integrasi antara *Frontend* Vue.js dengan *Backend* untuk memastikan seluruh alur login, proses absensi, dan laporan absensi berjalan dengan benar. Peneliti juga akan melaksanakan *User Acceptance Test* (UAT) untuk menilai apakah sistem telah memenuhi kebutuhan pengguna sebelum diterapkan lebih luas.

5. *Maintenance* : Tahap pemeliharaan sistem merupakan tahap akhir dalam metode *waterfall* yang dilakukan setelah aplikasi selesai dikembangkan dan digunakan. Pada tahap ini, dilakukan proses perbaikan terhadap kesalahan, *error*, atau *bug* yang ditemukan selama pengujian maupun saat sistem digunakan oleh pengguna. Selain itu, hasil evaluasi dari penggunaan sistem juga dijadikan sebagai bahan pengembangan dan penyempurnaan aplikasi di masa mendatang agar sistem dapat berjalan lebih optimal dan sesuai dengan kebutuhan pengguna.

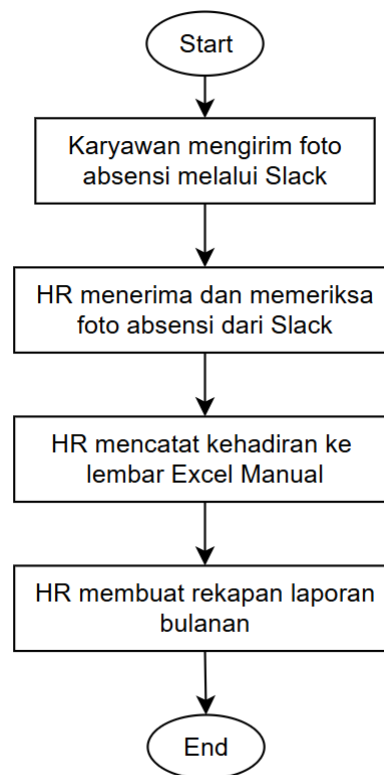
BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk memahami permasalahan sistem absensi yang berjalan dan menentukan solusi yang tepat. Saat ini, proses absensi di Perusahaan Jasa Teknologi Informasi masih dilakukan melalui Slack dengan pengiriman foto manual, sehingga rawan kecurangan dan kurang efisien. Berdasarkan kondisi tersebut, dirancang sistem absensi berbasis *face recognition* dan GPS.

3.1.1 Proses Bisnis Berjalan



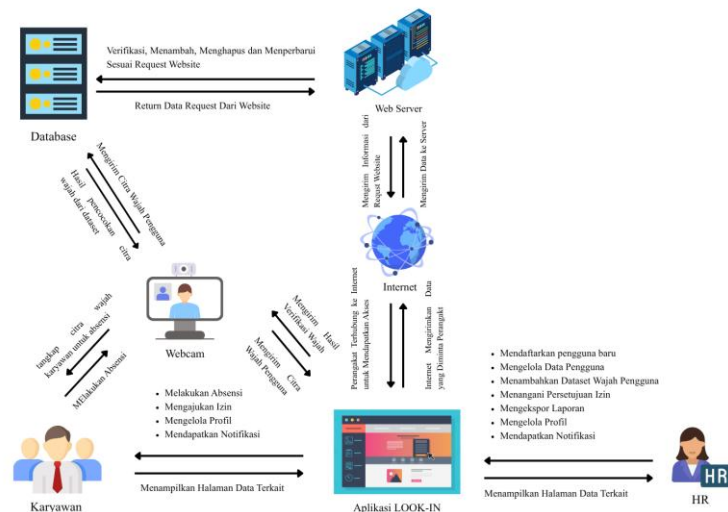
Gambar 3.1 Proses Bisnis Berjalan

Setiap karyawan diwajibkan mengirimkan foto sebagai bukti kehadiran pada grup atau kanal yang telah ditentukan oleh *Human Resource* (HR) setiap hari kerja. Foto tersebut berfungsi sebagai bukti bahwa karyawan telah hadir dan melakukan pekerjaan sesuai jadwal. Setelah

karyawan mengirimkan foto, pihak HR akan memeriksa satu per satu foto yang masuk untuk memastikan bahwa foto tersebut sesuai dengan identitas karyawan. HR kemudian mencatat kehadiran secara manual ke dalam lembar kerja (Microsoft Excel) sebagai rekap harian. Proses ini juga mencakup pencatatan izin, keterlambatan, dan ketidakhadiran. Apabila ada karyawan yang sakit atau tidak dapat hadir, maka mereka diwajibkan memberikan pemberitahuan melalui pesan pribadi atau grup Slack dengan menyertakan alasan dan bukti seperti surat dokter. HR akan menyimpan bukti tersebut sebagai arsip dan mencatat status izin secara manual di *file* absensi.

3.1.2 Gambaran Umum Sistem

Rancang Bangun Aplikasi LOOK-IN dirancang untuk mempermudah proses absensi dan pengelolaan data karyawan secara otomatis menggunakan teknologi *face recognition*. Sistem ini mengintegrasikan perangkat *webcam*, *web server*, dan *database* melalui koneksi internet untuk memastikan seluruh aktivitas absensi dan manajemen data dapat dilakukan secara *real-time*. Aplikasi ini ditujukan bagi dua peran utama, yaitu karyawan dan HR, yang masing-masing memiliki fungsi berbeda namun saling terhubung dalam satu sistem terpusat.



Gambar 3.2 Gambaran Umum Sistem

hasil ekstraksi dibandingkan dengan dataset wajah yang telah diproses dan disimpan sebelumnya. Proses pencocokan dilakukan menggunakan algoritma LBPH (*Local Binary Pattern Histogram*) yang dijalankan melalui OpenCV dan Python. Hasil akhir dari proses ini berupa identitas wajah yang berhasil dikenali secara otomatis dan real-time.

3.2 Perancangan

Perancangan sistem meliputi kebutuhan fungsional dan kebutuhan non-fungsional. Berikut adalah perancangan sistem yang akan dikembangkan

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan fitur-fitur utama yang harus dimiliki sistem agar dapat berjalan sesuai dengan tujuan yang telah diterapkan. Kebutuhan fungsional pada sistem dapat dilihat pada tabel berikut:

Tabel 3.1 Kebutuhan Fungsional

No	Kode	Fungsional	Keterangan	Aktor
1	FR-001	Masuk	Karyawan dan HR dapat masuk ke sistem.	Karyawan dan HR
2	FR-002	Keluar	Karyawan dan HR dapat keluar dari sistem untuk mengakhiri sesi.	
3	FR-003	Melihat Laporan	Karyawan dan HR dapat melihat laporan yang tersedia sesuai hak akses.	
4	FR-004	Melihat Riwayat Absen	Karyawan dan HR dapat melihat riwayat absensi yang tersedia sesuai hak akses.	
5	FR-005	Mengelola Profil	Karyawan dan HR dapat mengubah dan melengkapi data pribadi di menu profil.	
6	FR-006	Mengubah Kata Sandi	Karyawan dan HR dapat mengubah kata sandi.	
7	FR-007	Mendapatkan Notifikasi	Karyawan dan HR mendapatkan notifikasi.	
8	FR-008	Melakukan Absensi Cepat	Karyawan dapat melakukan absensi cepat tanpa masuk ke sistem.	Karyawan
9	FR-009	Melakukan Absen	Karyawan dapat melakukan absensi	

			dengan <i>webcam</i> dan berdasarkan area radius lokasi tertentu..	
10	FR-010	Mengajukan Persetujuan	Karyawan dapat membuat pengajuan izin masuk kerja dengan menyertakan alasan dan durasi hari.	
11	FR-011	Menambah Catatan Keterlambatan	Karyawan dapat menambah catatan alasan keterlambatan sesuai dengan durasi telat	
12	FR-012	Daftar	HR dapat melakukan pendaftaran akun ke sistem	HR
13	FR-013	Mengelola Pengguna	HR dapat menambahkan, menghapus atau memperbarui data pengguna dalam sistem	
14	FR-014	Menambah <i>Dataset</i> Pengguna	HR dapat menambahkan dan menghapus <i>dataset</i> untuk <i>training</i> data,	
15	FR-015	Menangani Persetujuan	HR dapat menerima atau menolak persetujuan izin yang diajukan oleh karyawan.	
16	FR-016	<i>Export</i> Riwayat	HR dapat mengeksport absensi dalam format tertentu.	
17	FR-017	Mengelola Peraturan Absensi	HR dapat mengelola pengaturan untuk absensi yaitu berupa lokasi dan jam kerja.	

3.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional yang terdapat pada sistem dapat dilihat pada tabel berikut:

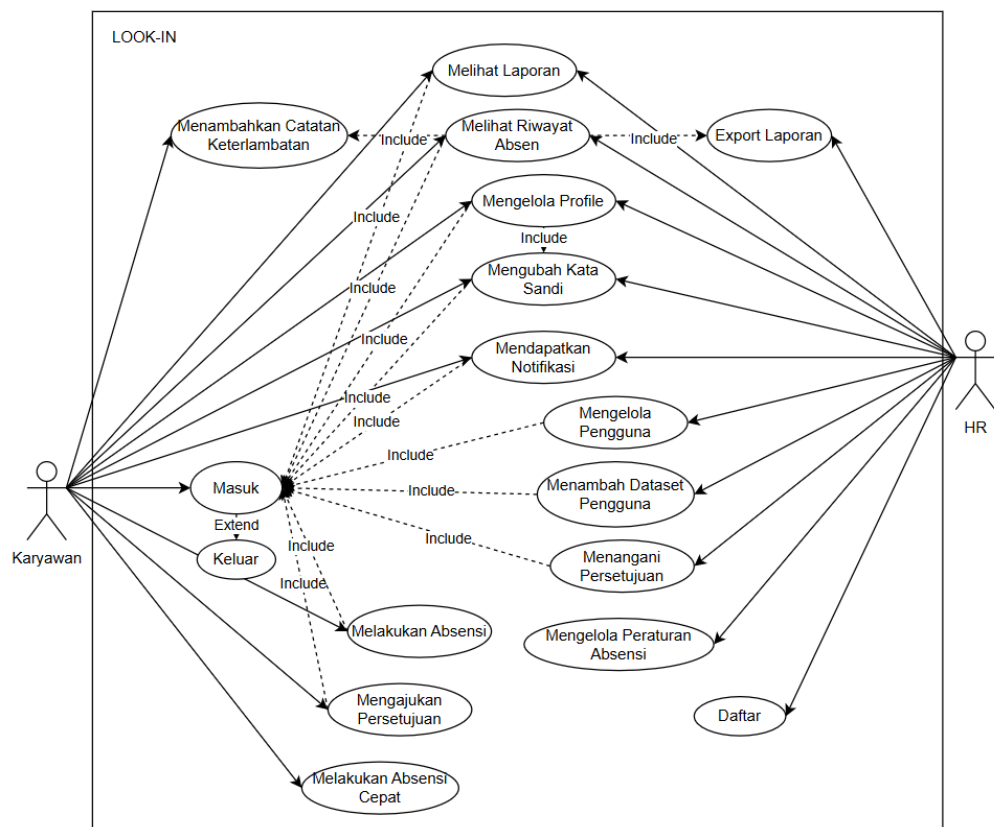
Tabel 3.2 Kebutuhan Non-Fungsional

No	Kode	Kebutuhan Non-Fungsional
1	NFR-001	Sistem memiliki tampilan UI/UX yang mudah dipahami dan <i>responsive</i> .
3	NFR-003	Sistem dibangun berbasis <i>website</i> dengan pendekatan PWA.
4	NFR-004	Sistem harus mampu menampilkan data laporan dan riwayat absensi dengan waktu respon maksimal 2 detik.

5	NFR-005	Sistem menyimpan kata sandi pengguna menggunakan metode enkripsi hashing untuk menjaga keamanan data akun.
6	NFR-006	Sistem menyimpan path dataset wajah pengguna dalam bentuk terenkripsi untuk menghindari akses tidak sah terhadap data dataset.
7	NFR-007	Sistem hanya dapat melakukan absensi apabila pengguna berada dalam radius lokasi yang telah ditentukan oleh HR.
8	NFR-008	Sistem melakukan validasi lokasi GPS dan data wajah sebelum absensi berhasil disimpan ke sistem.

3.2.3 Diagram Use Case

Use case diagram adalah gambaran interaksi antara komponen-komponen sistem yang memperkenalkan bagaimana interaksinya dengan pengguna. *Use case diagram* digunakan untuk menggambarkan secara ringkas siapa saja yang menggunakan sistem dan apa saja yang dilakukan (Hidayat, 2022)



Gambar 3.4 Use Case Diagram

Pada Gambar 3.4 memperlihatkan *use case diagram* sistem LOOK-IN. Diagram ini menggambarkan interaksi antara dua aktor utama, yaitu

karyawan dan HR, dengan sistem. Karyawan dapat melakukan proses masuk, keluar, melihat laporan, melihat riwayat absensi, mengelola profil, mengubah kata sandi, melakukan absensi, melakukan absensi cepat, mengajukan persetujuan, menambahkan catatan keterlambatan serta mendapatkan notifikasi dari sistem. Sedangkan aktor HR memiliki hak akses tambahan, seperti mendaftarkan pengguna baru, mengelola data pengguna, menambah *dataset* pengguna, menangani persetujuan izin, melihat laporan, mengekspor laporan, serta mengelola peraturan absensi.

3.2.4 Skenario *Use Case*

Use case skenario masuk berikut menggambarkan proses bagaimana pengguna (karyawan dan HR) dapat masuk ke akun masing-masing. Proses masuk dijelaskan melalui serangkaian langkah dalam skenario normal, sebagaimana ditunjukkan pada Tabel 3.3 dibawah ini.

Tabel 3.3 Skenario *Use Case* Masuk

<i>Use Case</i> Masuk	
Kode	FR-001
Deskripsi	Pengguna (karyawan dan HR) dapat masuk ke sistem dengan kredensial yang valid.
Aktor	Karyawan dan HR
Kondisi Awal	Pengguna (karyawan dan HR) sudah memiliki akun
Kondisi Akhir	Pengguna berhasil masuk ke sistem sesuai dengan peran masing-masing.
Normal Skenario	
Aktor	Sistem
1. Karyawan dan HR membuka halaman 'Masuk'.	2. Sistem menampilkan halaman masuk
3. Karyawan dan HR memasuki email dan kata sandi.	
4. Karyawan dan HR menekan tombol 'Masuk'.	5. Sistem melakukan validasi kredensial.
	6. Sistem mengarahkan pengguna ke beranda sesuai dengan peran.
Alur Kerja Alternatif	

1. Data yang dimasukkan tidak valid, maka sistem menampilkan pesan *error* dan meminta pengguna untuk mengulang memasukkan data.
2. Terjadi *error* saat verifikasi kredensial dengan database, maka sistem akan menampilkan pesan *error* dan meminta pengguna untuk mencoba lagi.

Use case skenario keluar bertujuan untuk mengakhiri sesi pengguna sehingga data pribadi dan akses sistem tetap terlindungi. Alur keluar dijelaskan melalui langkah-langkah dalam skenario normal, sebagaimana ditunjukkan pada tabel 3.4 berikut ini

Tabel 3.4 Skenario *Use Case* Keluar

<i>Use Case</i> Keluar	
Kode	FR-002
Deskripsi	Pengguna (karyawan dan HR) keluar dari sistem.
Aktor	Karyawan dan HR
Kondisi Awal	Pengguna (karyawan dan HR) sudah masuk ke sistem dengan kredensial valid.
Kondisi Akhir	Pengguna berhasil keluar dari sistem
Normal Skenario	
Aktor	Sistem
1. Karyawan dan HR menekan tombol 'Keluar'.	2. Sistem menerima permintaan keluar.
	3. Sistem menghapus token autentikasi.
	4. Sistem menampilkan pesan sukses bahwa pengguna telah keluar.
Alur Kerja Alternatif	
Jika terjadi kesalahan dalam proses keluar pengguna tetap berada di sesi yang aktif dan dapat mencoba untuk keluar lagi.	

Use case skenario menampilkan laporan untuk memperlihatkan data statistik pengguna. Alur melihat laporan dijelaskan melalui langkah-langkah dalam skenario normal, sebagaimana ditunjukkan pada tabel 3.5 dibawah ini

Tabel 3.5 Skenario *Use Case* Melihat Laporan

<i>Use Case</i> Melihat Laporan

Kode	FR-003
Deskripsi	Pengguna (karyawan dan HR) dapat melihat laporan sistem yang mencakup statistik pengguna
Aktor	Karyawan dan HR
Kondisi Awal	Pengguna (karyawan dan HR) sudah masuk ke sistem dengan kredensial valid.
Kondisi Akhir	Sistem menampilkan laporan sistem.
Normal Skenario	
Aktor	Sistem
1. Karyawan dan HR masuk ke sistem.	2. Sistem mengambil data dari <i>database</i> terkait laporan.
	3. Sistem menampilkan laporan sistem.
Alur Kerja Alternatif	
Jika terjadi kesalahan saat mengambil data laporan, sistem menampilkan pesan <i>error</i> dan meminta untuk mencoba lagi nanti.	

Use case skenario melihat riwayat kehadiran digunakan untuk menampilkan data statistik kehadiran pengguna. Alur proses melihat riwayat kehadiran dijelaskan melalui langkah-langkah pada skenario normal sebagaimana ditunjukkan pada Tabel 3.6 berikut

Tabel 3.6 Skenario *Use Case* Melihat Riwayat Kehadiran

<i>Use Case</i> Melihat Riwayat Kehadiran	
Kode	FR-004
Deskripsi	Pengguna (karyawan dan HR) dapat melihat riwayat absensi yang menampilkan data kehadiran, keterlambatan, izin, dan status persetujuan
Aktor	Karyawan dan HR
Kondisi Awal	Pengguna (karyawan dan HR) sudah masuk ke sistem dengan kredensial valid.
Kondisi Akhir	Sistem menampilkan daftar riwayat kehadiran beserta ringkasan statistik kehadiran dan status izin
Normal Skenario	
Aktor	Sistem
1. Karyawan dan HR masuk ke sistem.	

2. Karyawan dan HR memilih menu 'Riwayat Kehadiran'.	3. Sistem mengambil data absensi, keterlambatan dan izin dari <i>database</i> .
	4. Sistem menampilkan ringkasan statistik seperti jumlah absen hari ini, terlambat, sakit, dan menunggu perizinan.
	5. Sistem menampilkan tabel berisi data kehadiran seperti nama lengkap, tanggal, keterangan, status, serta koordinat lokasi
6. Pengguna dapat mengurutkan data berdasarkan bulan.	7. Sistem menampilkan hasil penyaringan sesuai pilihan pengguna
Alur Kerja Alternatif	
Jika sistem gagal memuat data, maka akan muncul pesan <i>error</i> dan pengguna diminta mencoba kembali.	

Use case skenario mengelola digunakan untuk pengguna melihat dan memperbarui data profil seperti nama, email, dan nomor telepon, Alur proses mengelola profil dijelaskan melalui langkah-langkah pada skenario normal sebagaimana ditunjukkan pada Tabel 3.7 berikut

Tabel 3.7 Skenario *Use Case* Mengelola Profil

<i>Use Case</i> Mengelola Profil	
Kode	FR-005
Deskripsi	Pengguna (karyawan dan HR) dapat mengelola profil pribadi seperti nama, email, dan nomor telepon.
Aktor	Karyawan dan HR
Kondisi Awal	Pengguna (karyawan dan HR) sudah masuk ke sistem dengan kredensial valid.
Kondisi Akhir	Data profil berhasil diperbarui dan tersimpan ke dalam sistem.
Normal Skenario	
Aktor	Sistem
1. Karyawan dan HR masuk ke sistem.	
2. Karyawan dan HR memilih menu 'Pengaturan'.	3. Sistem menampilkan data profil pengguna seperti nama, email, nomor

	telepon, peran dan foto profil.
4. Pengguna dapat mengubah informasi pribadi seperti nama atau nomor telepon	5. Sistem melakukan validasi terhadap data yang dimasukkan.
6. Pengguna menekan tombol 'Perbarui Profil'.	7. Sistem menyimpan perubahan.
Alur Kerja Alternatif	
Jika sistem gagal menyimpan perubahan profil maka akan muncul pesan <i>error</i> .	

Use case skenario mengubah kata sandi digunakan untuk pengguna memperbarui kata sandi melalui halaman pengaturan profil. Alur proses melihat mengubah kata sandi dijelaskan melalui langkah-langkah pada skenario normal sebagaimana ditunjukkan pada Tabel 3.8 berikut

Tabel 3.8 Skenario *Use Case* Mengubah Kata Sandi

<i>Use Case</i> Mengubah Kata Sandi	
Kode	FR-006
Deskripsi	Pengguna (karyawan dan HR) dapat mengubah kata sandi melalui halaman pengaturan profil.
Aktor	Karyawan dan HR
Kondisi Awal	Pengguna (karyawan dan HR) sudah masuk ke sistem dengan kredensial valid.
Kondisi Akhir	Sistem memperbarui kata sandi.
Normal Skenario	
Aktor	Sistem
1. Karyawan dan HR masuk ke sistem.	
2. Karyawan dan HR memilih menu 'Pengaturan'.	
3. Pengguna memasukkan kata sandi lama, baru, dan konfirmasi	4. Sistem memverifikasi kata sandi lama dan memastikan kata sandi baru memenuhi syarat keamanan.
	5. Sistem kata sandi baru ke dalam database.
	6. Sistem memperbarui kata sandi.
Alur Kerja Alternatif	
Jika kata sandi lama tidak sesuai atau konfirmasi kata sandi baru tidak cocok, sistem menampilkan pesan kesalahan dan meminta pengguna untuk mengulang proses.	

Use case skenario mendapatkan notifikasi digunakan untuk menampilkan pemberitahuan kepada pengguna terkait persetujuan izin. Alur proses melihat mendapatkan notifikasi dijelaskan melalui langkah-langkah pada skenario normal sebagaimana ditunjukkan pada Tabel 3.9 berikut.

Tabel 3.9 Skenario *Use Case* Mendapatkan Notifikasi

<i>Use Case</i> Mendapatkan Notifikasi	
Kode	FR-007
Deskripsi	Pengguna (karyawan dan HR) dapat menerima dan melihat notifikasi yang dikirim oleh sistem terkait status persetujuan izin.
Aktor	Karyawan dan HR
Kondisi Awal	Pengguna (karyawan dan HR) sudah masuk ke sistem dengan kredensial valid.
Kondisi Akhir	Sistem menampilkan daftar notifikasi sesuai aktivitas pengguna.
Normal Skenario	
Aktor	Sistem
1. Karyawan dan HR masuk ke sistem.	
2. Karyawan dan HR memilih menu 'Notifikasi'.	
	3. Sistem menampilkan daftar notifikasi persetujuan izin.
4. Pengguna membuka notifikasi untuk melihat detail.	5. Sistem menampilkan detail persetujuan izin.
Alur Kerja Alternatif	
Jika sistem gagal mengambil data notifikasi maka akan muncul pesan <i>error</i> .	

Use case skenario melakukan absensi cepat digunakan untuk karyawan melakukan proses absensi secara instan tanpa perlu masuk ke sistem. Alur proses melihat melakukan absensi cepat dijelaskan melalui langkah-langkah pada skenario normal sebagaimana ditunjukkan pada Tabel 3.10 berikut

Tabel 3.10 Skenario *Use Case* Melakukan Absensi Cepat

<i>Use Case</i> Melakukan Absensi Cepat	
Kode	FR-008
Deskripsi	Karyawan dapat melakukan absensi instan menggunakan <i>face recognition</i> tanpa perlu masuk ke sistem.
Aktor	Karyawan
Kondisi Awal	Karyawan membuka halaman 'Masuk'
Kondisi Akhir	Data absensi karyawan tersimpan di sistem hasil verifikasi wajah, dan lokasi ditampilkan
Normal Skenario	
Aktor	Sistem
1. Karyawan membuka halaman 'Masuk'.	
2. Karyawan memencet ikon kamera.	3. Sistem meminta persetujuan akses kamera dan lokasi.
4. Pengguna memberikan akses kamera dan lokasi	5. Sistem mengaktifkan kamera dan menampilkan tampilan pengambilan wajah.
	6. Sistem memverifikasi wajah berdasarkan <i>dataset</i> yang tersimpan.
	7. Sistem menampilkan detail absensi seperti hari, nama, status, dan lokasi
Alur Kerja Alternatif	
Jika wajah tidak dikenali atau akses kamera dan lokasi belum diizinkan, sistem akan menampilkan pesan <i>error</i> dan meminta pengguna untuk mencoba kembali.	

Use case skenario melakukan absensi digunakan untuk karyawan melakukan absensi setelah masuk ke dalam sistem menggunakan akun yang valid. Alur proses melakukan absensi dijelaskan melalui langkah-langkah pada skenario normal sebagaimana ditunjukkan pada Tabel 3.11 berikut

Tabel 3.11 Skenario *Use Case* Melakukan Absensi

<i>Use Case</i> Melakukan Absensi	
Kode	FR-009
Deskripsi	Karyawan dapat melakukan absensi menggunakan <i>face</i>

	<i>recognition</i> setelah masuk ke dalam sistem.
Aktor	Karyawan
Kondisi Awal	Karyawan telah masuk ke sistem menggunakan akun yang valid.
Kondisi Akhir	Data absensi berhasil tersimpan dalam sistem dan menampilkan detail absen ke karyawan.
Normal Skenario	
Aktor	Sistem
1. Karyawan masuk ke sistem	
2. Karyawan memilih menu 'Absensi'.	3. Sistem meminta persetujuan akses kamera dan lokasi.
4. Pengguna memberikan akses kamera dan lokasi	5. Sistem mengaktifkan kamera dan menampilkan tampilan pengambilan wajah.
	6. Sistem memverifikasi wajah berdasarkan <i>dataset</i> yang tersimpan.
	7. Jika cocok, sistem menyimpan data absensi dan menampilkan detail absensi.
Alur Kerja Alternatif	
Jika wajah tidak dikenali atau akses kamera dan lokasi ditolak, sistem menampilkan pesan <i>error</i> dan meminta pengguna untuk mencoba kembali.	

Use case skenario mengajukan persetujuan digunakan untuk karyawan melakukan perizinan sakit dengan formulir perizinan dan mengunggah bukti pendukung melalui sistem. Alur proses mengajukan persetujuan dijelaskan melalui langkah-langkah pada skenario normal sebagaimana ditunjukkan pada Tabel 3.12 berikut

Tabel 3.12 Skenario *Use Case* Mengajukan Persetujuan

<i>Use Case</i> Mengajukan Persetujuan	
Kode	FR-010
Deskripsi	Karyawan dapat mengajukan izin dengan mengisi formulir perizinan dan mengunggah bukti pendukung melalui sistem.
Aktor	Karyawan

Kondisi Awal	Karyawan telah masuk ke sistem menggunakan akun yang valid.
Kondisi Akhir	Data pengajuan tersimpan di sistem dan status izin ditampilkan pada daftar perizinan dengan status “Menunggu Persetujuan”.
Normal Skenario	
Aktor	Sistem
1. Karyawan masuk ke sistem	
2. Karyawan memilih menu ‘Perizinan’.	
3. Karyawan menekan tombol ‘Tambah Data’ untuk membuat pengajuan baru.	4. Sistem menampilkan formulir pengajuan izin yang berisi kolom pilih alasan izin, deskripsi dan <i>upload</i> bukti foto.
5. Karyawan mengisi pengajuan izin.	
	6. Sistem memvalidasi semua <i>inputan</i> dan menyimpan data ke <i>database</i> .
	7. Menampilkan data izin baru di daftar perizinan.
Alur Kerja Alternatif	
1. Jika karyawan tidak mengisi semua kolom wajib sistem akan menampilkan pesan <i>error</i> . 2. Format <i>file</i> bukti tidak sesuai (bukan .jpg atau .png) sistem menampilkan pesan <i>error</i> . 3. Karyawan membatalkan proses sebelum menekan tombol ‘Tambah Perizinan’ sistem tidak menyimpan data apa pun dan kembali ke halaman daftar perizinan.	

Use case skenario mengajukan persetujuan digunakan untuk karyawan melakukan perizinan sakit dengan formulir perizinan dan mengunggah bukti pendukung melalui sistem. Alur proses mengajukan persetujuan dijelaskan melalui langkah-langkah pada skenario normal sebagaimana ditunjukkan pada Tabel 3.13 berikut

Tabel 3.13 Skenario *Use Case* Menambah Catatan Keterlambatan

<i>Use Case</i> Menambah Catatan Keterlambatan	
Kode	FR-011
Deskripsi	Karyawan dapat menambahkan catatan keterlambatan sesuai dengan ketentuan durasi telat
Aktor	Karyawan

Kondisi Awal	Karyawan telah masuk ke sistem menggunakan akun yang valid.
Kondisi Akhir	Data catatan tersimpan di sistem dan dapat dilihat oleh HR
Normal Skenario	
Aktor	Sistem
1. Karyawan masuk ke sistem	
2. Karyawan memilih menu 'Riwayat Absensi'	3. Sistem menampilkan tampilan halaman riwayat absensi
4. Karyawan memilih tombol tambah catatan	5. Sistem menampilkan <i>pop up</i> untuk mengisi catatan disertai dengan jumlah kata yang harus dituliskan
6. Karyawan mengisi catatan keterlambatan	
	7. Sistem memvalidasi semua <i>inputan</i> dan menyimpan data ke <i>database</i> .
Alur Kerja Alternatif	
Jika karyawan tidak mengisi catatan dengan kata yang sesuai dengan durasi keterlambatan data tidak dapat disimpan	

Use case skenario daftar digunakan untuk HR melakukan pendaftaran akun agar dapat mengelola data karyawan dan proses absensi. Alur proses daftar dijelaskan melalui langkah-langkah pada skenario normal sebagaimana ditunjukkan pada Tabel 3.14 berikut

Tabel 3.14 Skenario *Use Case* Daftar

<i>Use Case</i> Daftar	
Kode	FR-012
Deskripsi	HR dapat melakukan pendaftaran akun ke dalam sistem untuk mengelola data karyawan. Pendaftaran akun HR pertama dapat dilakukan tanpa kode undangan. Setelah akun HR tersedia di dalam sistem, pendaftaran HR berikutnya wajib menggunakan kode undangan yang valid.
Aktor	HR
Kondisi Awal	HR membuka halaman daftar.
Kondisi Akhir	Akun HR berhasil dibuat.
Normal Skenario	

Aktor	Sistem
1. HR membuka halaman pendaftaran sistem.	2. Sistem menampilkan <i>form</i> pendaftaran yang berisi kolom username, name lengkap, email, dan kata sandi.
3. HR mengisi seluruh kolom pendaftaran dengan data valid dan menekan tombol 'Daftar'.	4. Sistem memvalidasi <i>input</i> data.
	5. Jika belum terdapat akun HR pada database, sistem mengabaikan validasi kode undangan dan melanjutkan proses pendaftaran.
	6. Jika sudah terdapat akun HR pada database, sistem memvalidasi kode undangan yang dimasukkan.
	7. Sistem menyimpan data yang valid ke dalam database.
	8. Sistem menampilkan pesan bahwa pendaftaran berhasil.
	9. Sistem mengarahkan HR ke halaman masuk (<i>login</i>).
Alur Kerja Alternatif	
1. Jika salah satu kolom wajib tidak diisi, sistem menampilkan pesan: "Harap lengkapi semua kolom sebelum melanjutkan." 2. Jika format email tidak valid, sistem menampilkan pesan: "Format email tidak valid." 3. Jika kata sandi tidak memenuhi ketentuan keamanan, sistem menampilkan pesan validasi kata sandi. 4. Jika username sudah terdaftar, sistem menampilkan pesan: "Username sudah digunakan." 5. Jika akun HR sudah tersedia dan kode undangan tidak diisi, sistem menampilkan pesan: "Kode undangan wajib diisi." 6. Jika akun HR sudah tersedia dan kode undangan tidak valid, sistem menampilkan pesan: "Kode undangan tidak valid." 7. Jika terjadi kegagalan penyimpanan data, sistem menampilkan pesan: "Pendaftaran gagal, silakan coba kembali."	

Use case skenario mengelola pengguna digunakan untuk HR menambahkan, menghapus, atau memperbaiki data karyawan dalam sistem. Alur proses mengelola pengguna dijelaskan melalui langkah-langkah pada skenario normal sebagaimana ditunjukkan pada Tabel 3.15 berikut

Tabel 3.15 Skenario *Use Case* Mengelola Pengguna

<i>Use Case</i> Mengelola Pengguna	
Kode	FR-013
Deskripsi	HR dapat menambah, mengedit, dan menghapus data pengguna pada sistem melalui halaman pengguna.
Aktor	HR
Kondisi Awal	HR telah masuk ke dalam sistem dan berada pada halaman pengguna.
Kondisi Akhir	Data pengguna berhasil diperbarui, ditambahkan, atau dihapus dari sistem.
Normal Skenario	
Aktor	Sistem
1. HR membuka halaman 'Pengguna'.	2. Sistem menampilkan seluruh data karyawan beserta informasi.
3. HR menekan tombol 'Tambah Data' untuk menambahkan pengguna baru.	4. Sistem menampilkan <i>form</i> penambahan data yang berisi kolom email, nama pengguna, kata sandi, dan peran pengguna.
5. HR mengisi data pengguna sesuai kebutuhan dan menekan tombol 'Tambah data pengguna'	6. Sistem memvalidasi <i>input</i> data, kemudian menyimpan data karyawan baru ke dalam <i>database</i> .
7. HR dapat memilih ikon 'Edit' untuk memperbaiki data karyawan yang sudah ada.	8. Sistem menampilkan <i>form</i> edit dengan data pengguna yang dipilih, lalu memperbaiki data setelah disimpan.
9. HR dapat memilih ikon 'Hapus' untuk menghapus data pengguna dari sistem.	10. Sistem memunculkan konfirmasi penghapusan, lalu menghapus data jika dikonfirmasi.
Alur Kerja Alternatif	

1. Jika salah satu kolom tidak diisi, sistem akan menampilkan pesan *error* “Harap lengkapi semua kolom sebelum melanjutkan”.
2. Format email tidak valid, sistem menampilkan pesan “Format email tidak valid”.
3. Jika peran pengguna belum dipilih, sistem menampilkan pesan “Pilih peran pengguna sebelum melanjutkan.”

Use case skenario menambah *dataset* pengguna digunakan untuk HR menambahkan, menghapus, atau memperbarui data karyawan dalam sistem. Alur proses menambah *dataset* pengguna dijelaskan melalui langkah-langkah pada skenario normal sebagaimana ditunjukkan pada Tabel 3.16 berikut

Tabel 3.16 Skenario *Use Case* Menambah *Dataset* Pengguna

<i>Use Case</i> Menambah <i>Dataset</i> Pengguna	
Kode	FR-014
Deskripsi	HR dapat menambah, mengedit, dan menghapus data pengguna pada sistem melalui halaman pengguna.
Aktor	HR
Kondisi Awal	HR telah masuk ke dalam sistem dan berada pada halaman pengguna.
Kondisi Akhir	Data pengguna berhasil diperbarui, ditambahkan, atau dihapus dari sistem.
Normal Skenario	
Aktor	Sistem
1. HR membuka halaman ‘Pengguna’.	2. Sistem menampilkan seluruh data karyawan beserta informasi.
3. HR menekan tombol ‘Tambah Data’ untuk menambahkan pengguna baru.	4. Sistem menampilkan <i>form</i> penambahan data yang berisi kolom email, nama pengguna, kata sandi, dan peran pengguna.
5. HR mengisi data pengguna sesuai kebutuhan dan menekan tombol ‘Tambah data pengguna’	6. Sistem memvalidasi <i>input</i> data, kemudian menyimpan data karyawan baru ke dalam <i>database</i> .
7. HR dapat memilih ikon ‘Edit’ untuk memperbarui data karyawan yang sudah ada.	8. Sistem menampilkan <i>form</i> edit dengan data pengguna yang dipilih, lalu

	memperbarui data setelah disimpan.
9. HR dapat memilih ikon 'Hapus' untuk menghapus data pengguna dari sistem.	10. Sistem memunculkan konfirmasi penghapusan, lalu menghapus data jika dikonfirmasi.
Alur Kerja Alternatif	
1. Jika salah satu kolom tidak diisi, sistem akan menampilkan pesan <i>error</i> "Harap lengkapi semua kolom sebelum melanjutkan". 2. Format email tidak valid, sistem menampilkan pesan "Format email tidak valid". 3. Jika peran pengguna belum dipilih, sistem menampilkan pesan "Pilih peran pengguna sebelum melanjutkan."	

Use case skenario menangani persetujuan izin digunakan untuk HR dalam meninjau, menyetujui, atau menolak pengajuan izin yang diajukan oleh karyawan melalui sistem. Alur proses menangani persetujuan izin dijelaskan melalui langkah-langkah pada skenario normal sebagaimana ditunjukkan pada Tabel 3.17 berikut

Tabel 3.17 Skenario *Use Case* Menangani Persetujuan

<i>Use Case</i> Menangani Persetujuan	
Kode	FR-015
Deskripsi	HR dapat meninjau, menyetujui, atau menolak pengajuan izin yang diajukan oleh karyawan melalui halaman persetujuan.
Aktor	HR
Kondisi Awal	HR telah masuk ke dalam sistem dan berada pada halaman persetujuan.
Kondisi Akhir	Status pengajuan izin karyawan berhasil diperbarui menjadi disetujui atau ditolak, dan sistem mengirimkan notifikasi kepada karyawan.
Normal Skenario	
Aktor	Sistem
1. HR membuka halaman 'Persetujuan'.	2. Sistem menampilkan daftar pengajuan izin karyawan dengan informasi nama,

	alasan, tanggal, status, dan aksi.
3. HR memilih salah satu pengajuan izin untuk ditinjau.	4. Sistem menampilkan <i>pop-up</i> detail pengajuan izin yang berisi nama pengguna, alasan izin, deskripsi, tanggal izin, bukti perizinan, dan status saat ini.
5. HR memeriksa detail pengajuan serta bukti perizinan.	6. Sistem menampilkan opsi tombol ‘Setuju’ dan ‘Tolak’.
7. HR menekan tombol ‘Setuju’ jika menyetujui atau ‘Tolak’ jika menolak pengajuan izin.	
8. HR mengonfirmasi keputusan tersebut.	9. Sistem memperbarui status pengajuan menjadi disetujui atau ditolak, menyimpan hasil keputusan ke <i>database</i> , dan mengirim notifikasi ke karyawan terkait.
Alur Kerja Alternatif	
Jika HR membatalkan proses setelah menekan tombol <i>Setujui</i> atau <i>Tolak</i> , sistem tidak mengubah status pengajuan dan tetap menandainya sebagai <i>Menunggu Persetujuan</i> .	

Use case skenario *export* laporan digunakan untuk HR dalam mengekspor data laporan kehadiran atau aktivitas karyawan dari sistem ke dalam format tertentu seperti PDF atau Excel. Alur proses *export* laporan dijelaskan melalui langkah-langkah pada skenario normal sebagaimana ditunjukkan pada Tabel 3.18 berikut

Tabel 3.18 Skenario *Use Case Export* Laporan

<i>Use Case Export</i> Laporan	
Kode	FR-016
Deskripsi	HR dapat mengekspor data laporan kehadiran atau aktivitas karyawan ke dalam format tertentu seperti PDF atau Excel melalui halaman riwayat absensi.

Aktor	HR
Kondisi Awal	HR telah masuk ke dalam sistem dan berada pada halaman riwayat laporan.
Kondisi Akhir	File laporan berhasil diekspor dan tersimpan di perangkat HR dalam format yang dipilih (PDF atau Excel).
Normal Skenario	
Aktor	Sistem
1. HR membuka halaman 'Riwayat Absen'.	2. Sistem menampilkan data kehadiran seluruh karyawan yang meliputi nama lengkap, jam absen, keterangan, status, dan lokasi absen.
3. HR memilih periode atau tanggal tertentu untuk ditampilkan.	4. Sistem menampilkan data kehadiran sesuai tanggal yang dipilih.
5. HR menekan tombol 'Download' yang berada di atas tabel.	6. Sistem memproses dan menghasilkan file laporan sesuai format pilihan HR.
7. HR menekan tombol 'Setuju' jika menyetujui atau 'Tolak' jika menolak pengajuan izin.	
8. HR menyimpan atau mengunduh file laporan ke perangkat.	
Alur Kerja Alternatif	
1. Jika data kehadiran kosong sesuai filter atau tanggal yang dipilih, sistem menampilkan pesan "Tidak ada data yang dapat diekspor." 2. Jika HR membatalkan proses ekspor sebelum selesai, sistem tidak menyimpan atau menghasilkan file apa pun.	

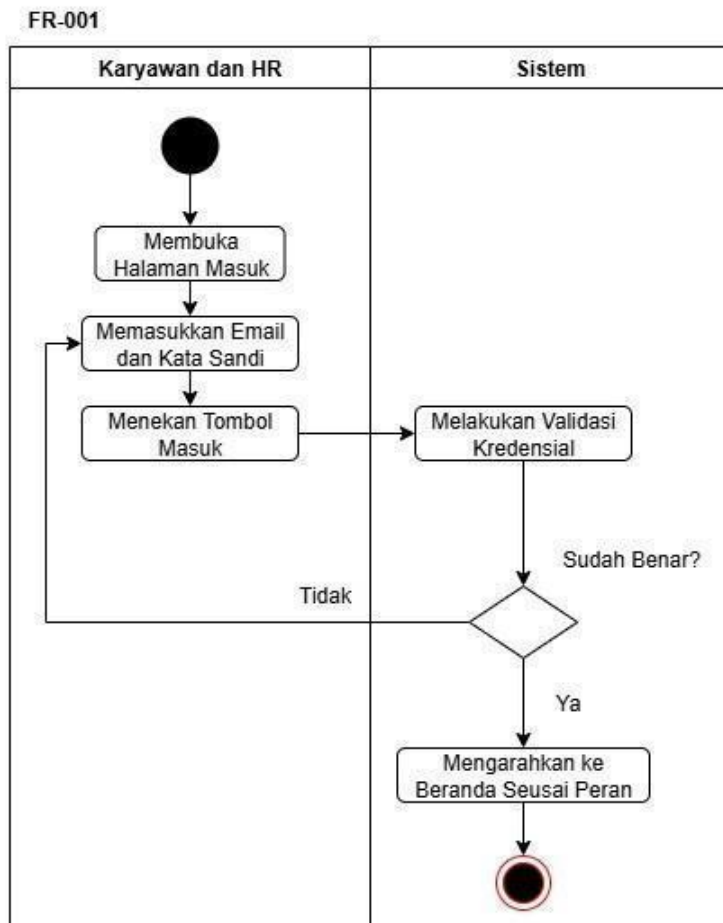
Use case skenario mengelola peraturan absensi digunakan untuk HR untuk mengubah jam shift kerja dan mengatur lokasi melalui GPS. Alur mengelola peraturan absensi dijelaskan melalui langkah-langkah pada skenario normal sebagaimana ditunjukkan pada Tabel 3.19 berikut

Tabel 3.19 Skenario *Use Case* Mengelola Peraturan Absensi

<i>Use Case</i> Mengelola Peraturan Absensi	
Kode	FR-017
Deskripsi	HR dapat mengelola peraturan absensi seperti mengganti <i>shift time</i> dan mengatur lokasi untuk absensi

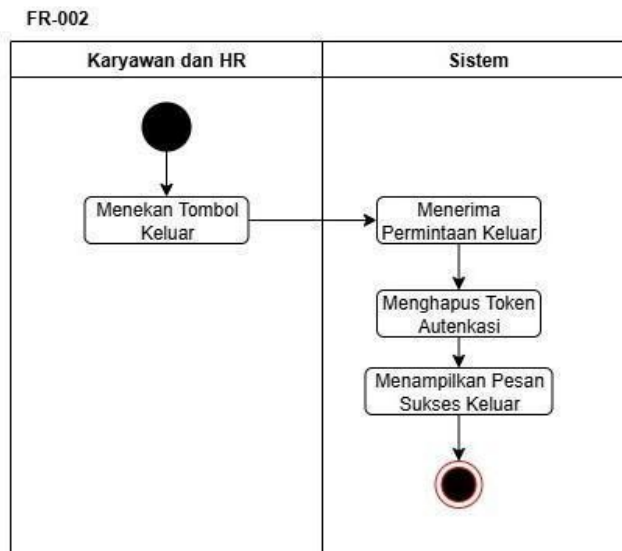
Aktor	HR
Kondisi Awal	HR telah masuk ke dalam sistem dan berada pada halaman pengaturan absensi.
Kondisi Akhir	Peraturan absensi berhasil disimpan
Normal Skenario	
Aktor	Sistem
1. HR memilih menu pengaturan absensi	2. Sistem menampilkan halaman pengaturan absensi
3. HR memilih lokasi dan radius GPS absensi	
4. HR mengubah jam shift kerja	
	5. Sistem melakukan validasi data input
	6. Sistem menyimpan data pengaturan absensi ke sistem
Alur Kerja Alternatif	
Jika HR membatalkan proses perubahan pengaturan absensi, sistem tidak menyimpan perubahan data ke dalam sistem.	

3.2.4 Activity Diagram



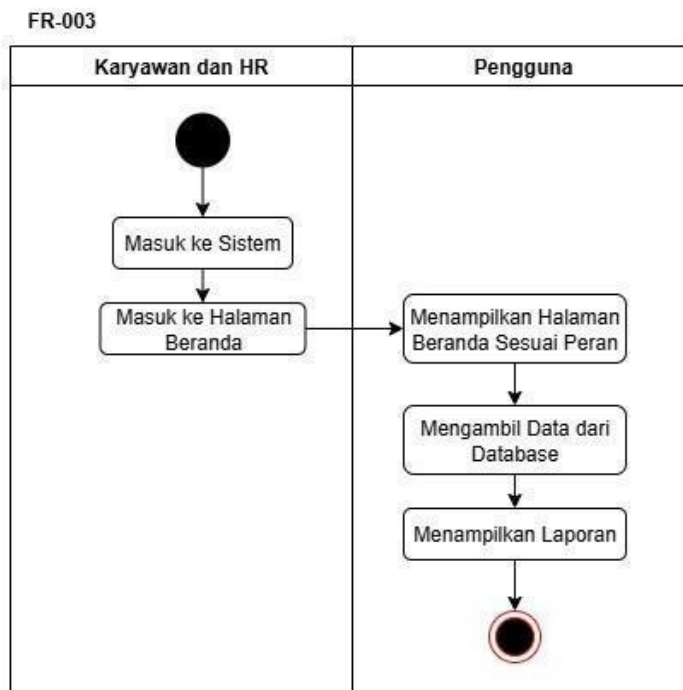
Gambar 3.5 Activity Diagram Masuk

Gambar 3.5 menjelaskan alur proses aktivitas masuk ke sistem, di mana karyawan atau HR membuka halaman masuk, memasukkan email dan kata sandi, lalu sistem melakukan validasi kredensial. Jika data yang dimasukkan benar, pengguna akan diarahkan ke beranda sesuai perannya, sedangkan jika salah, sistem akan meminta pengguna untuk mengulangi proses masuk.



Gambar 3.6 Activity Diagram Keluar

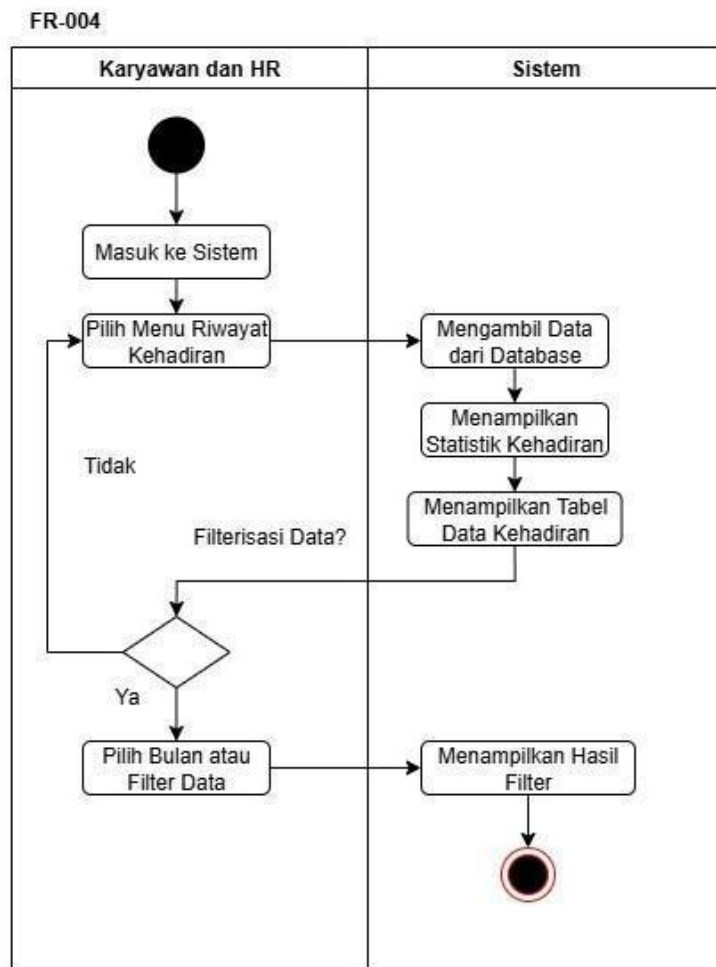
Gambar 3.6 menjelaskan alur proses aktivitas keluar dari sistem, di mana karyawan atau HR menekan tombol keluar, kemudian sistem menerima permintaan tersebut, menghapus token autentikasi pengguna, dan menampilkan pesan bahwa proses keluar berhasil.



Gambar 3.7 Activity Diagram Melihat Laporan

Gambar 3.7 menjelaskan alur proses melihat laporan, di mana karyawan atau HR yang telah masuk ke sistem akan diarahkan ke halaman

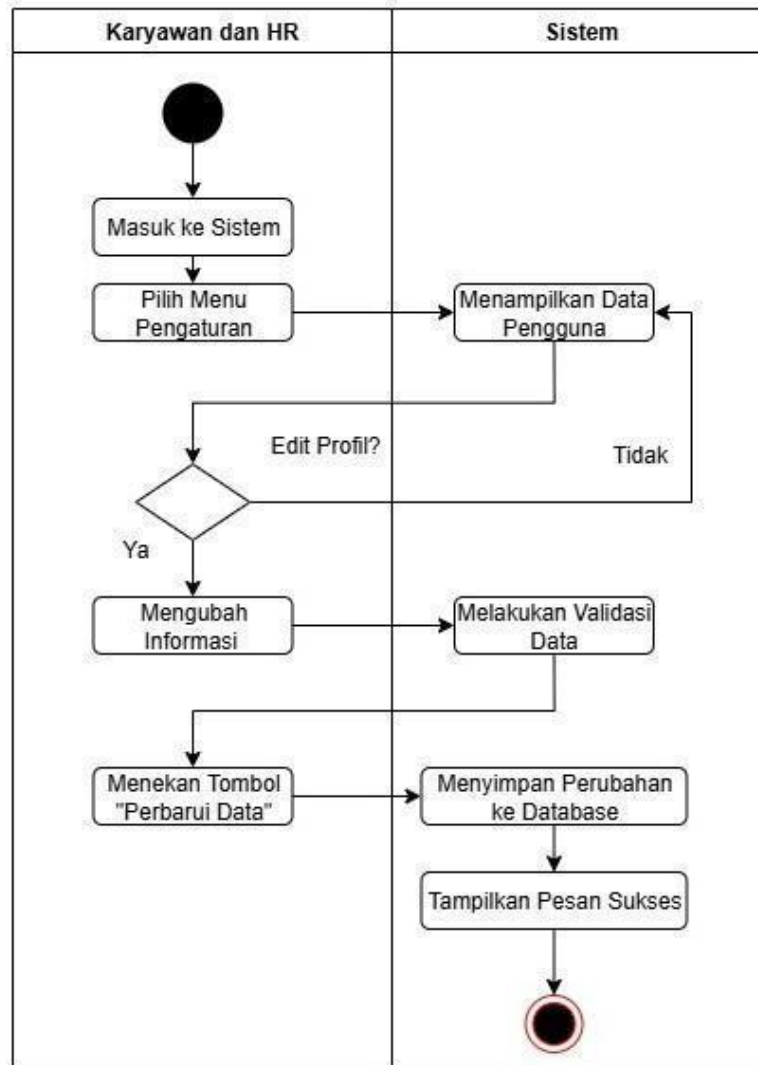
beranda sesuai peran, kemudian sistem mengambil data dari *database* dan menampilkan laporan yang sesuai.



Gambar 3.8 Activity Diagram Melihat Riwayat Kehadiran

Gambar 3.8 menjelaskan alur proses melihat riwayat kehadiran, di mana karyawan atau HR masuk ke sistem lalu memilih menu riwayat kehadiran. Sistem akan mengambil data dari *database* dan menampilkan statistik serta tabel kehadiran. Jika pengguna ingin melakukan filterisasi, mereka dapat memilih bulan atau kriteria tertentu, kemudian sistem akan menampilkan hasil filter sesuai dengan data yang dipilih.

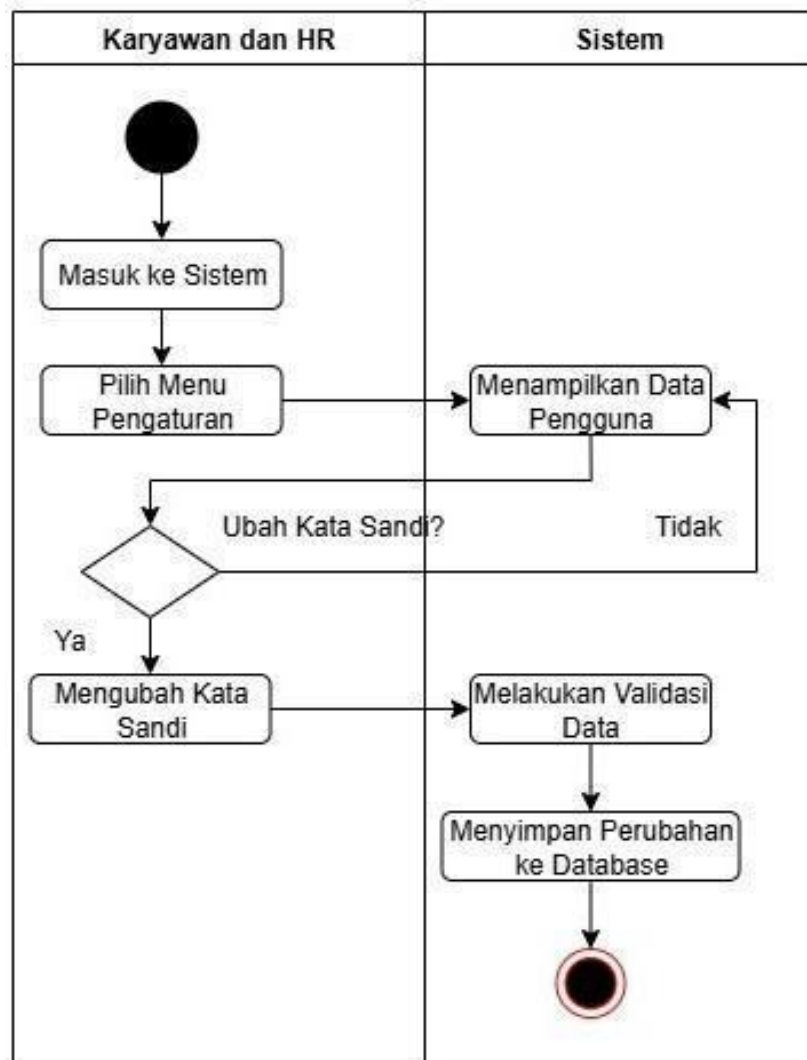
FR-005



Gambar 3.9 Activity Diagram Mengelola Profil

Gambar 3.9 menjelaskan alur proses mengelola profil, di mana karyawan atau HR masuk ke sistem dan memilih menu pengaturan untuk melihat serta mengedit data profil. Setelah melakukan perubahan, sistem memvalidasi dan menyimpan data ke *database*, lalu menampilkan pesan sukses.

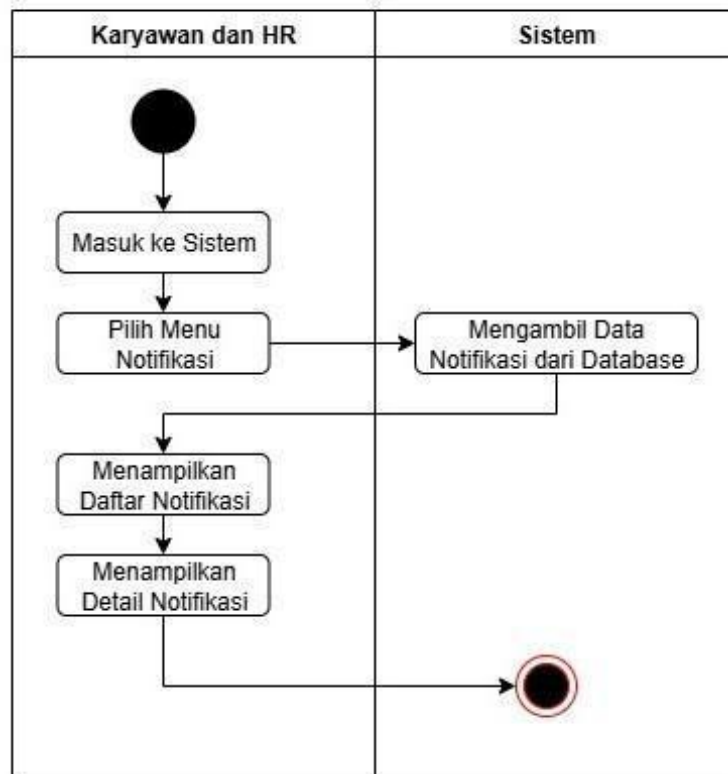
FR-006



Gambar 3.10 Activity Diagram Mengubah Kata Sandi

Gambar 3.10 menjelaskan alur proses mengubah kata sandi, di mana karyawan atau HR masuk ke sistem dan memilih menu pengaturan. Jika ingin mengubah kata sandi, pengguna memasukkan data baru, kemudian sistem memvalidasi dan menyimpan perubahan ke *database*.

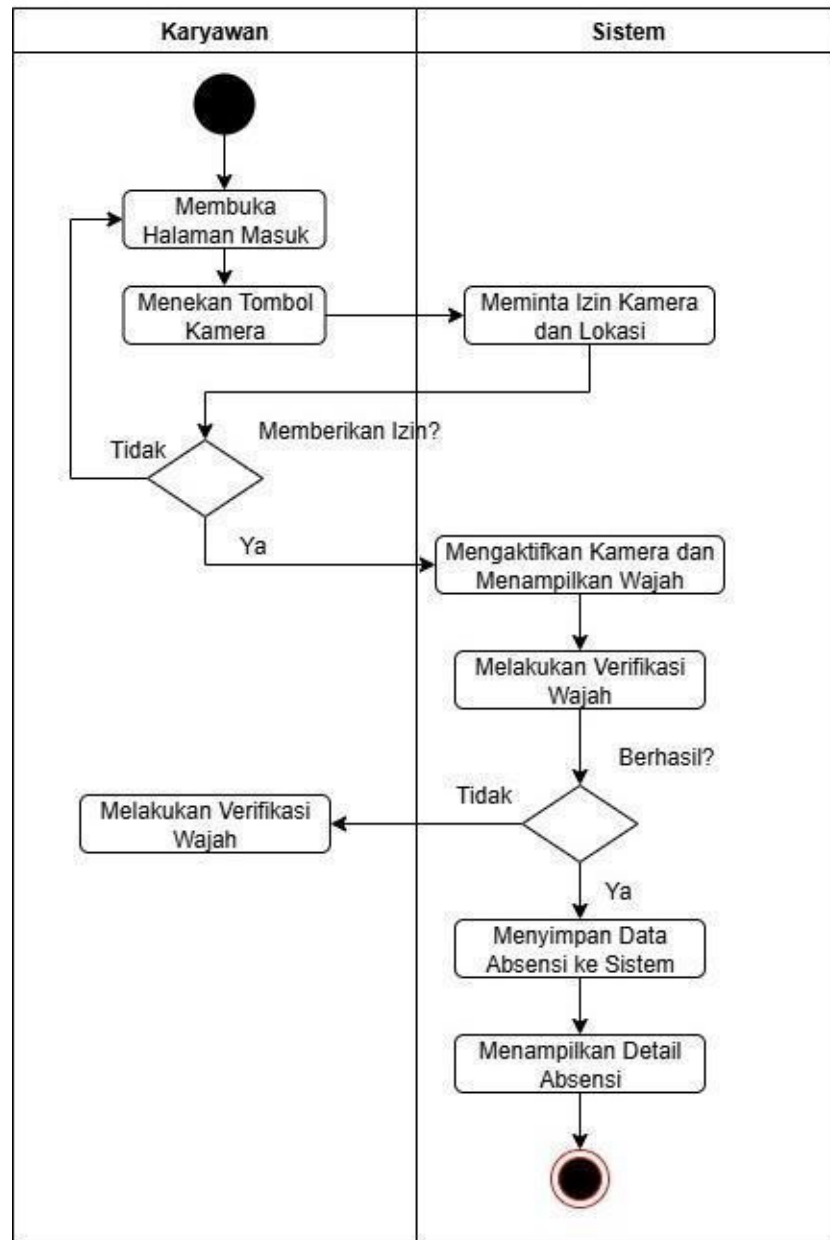
FR-007



Gambar 3.11 Activity Diagram Mendapatkan Notifikasi

Gambar 3.11 menjelaskan alur proses mendapatkan notifikasi, di mana karyawan atau HR masuk ke sistem dan memilih menu notifikasi. Sistem akan mengambil data notifikasi dari *database*, lalu menampilkan daftar serta detail notifikasi kepada pengguna.

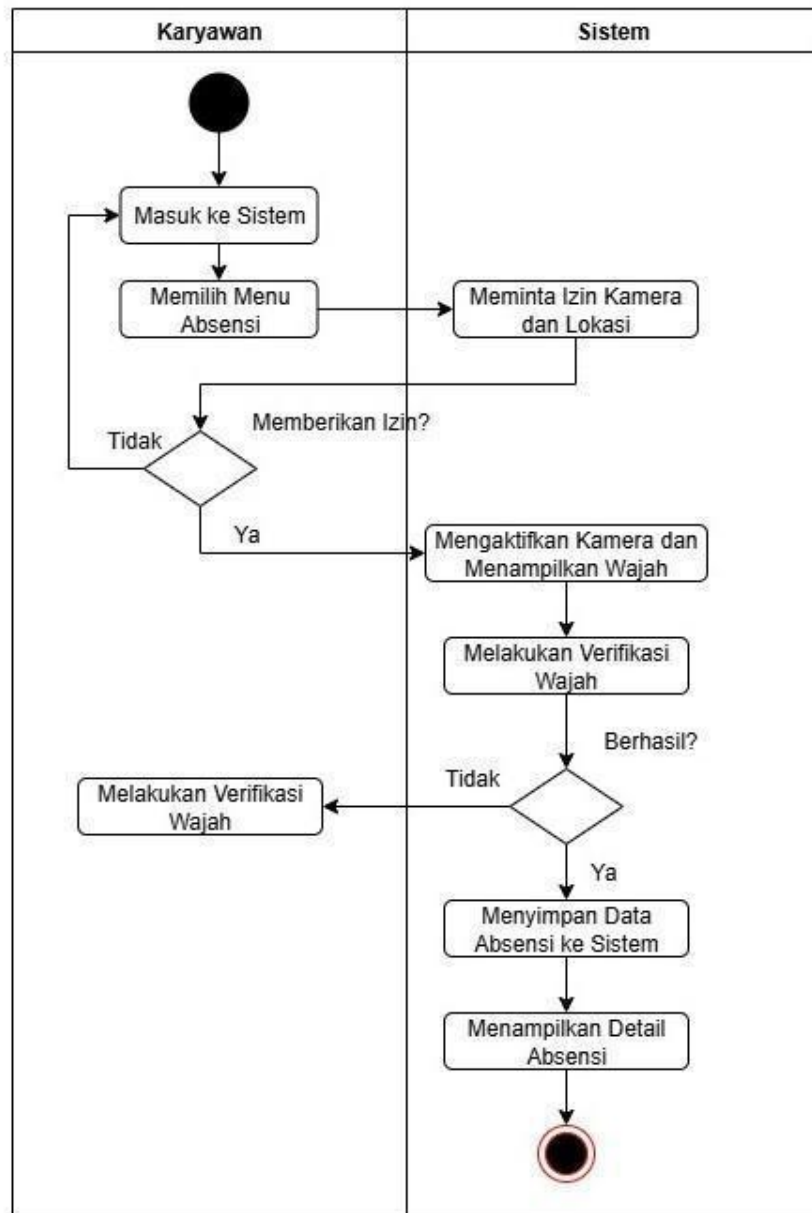
FR-008



Gambar 3.12 Activity Diagram Melakukan Absensi Cepat

Gambar 3.12 menjelaskan alur proses melakukan absensi cepat, di mana karyawan membuka menu absensi dan menyalakan kamera. Sistem akan mendeteksi wajah untuk verifikasi. Jika wajah terverifikasi, sistem menyimpan data kehadiran dan menampilkan detail absensi.

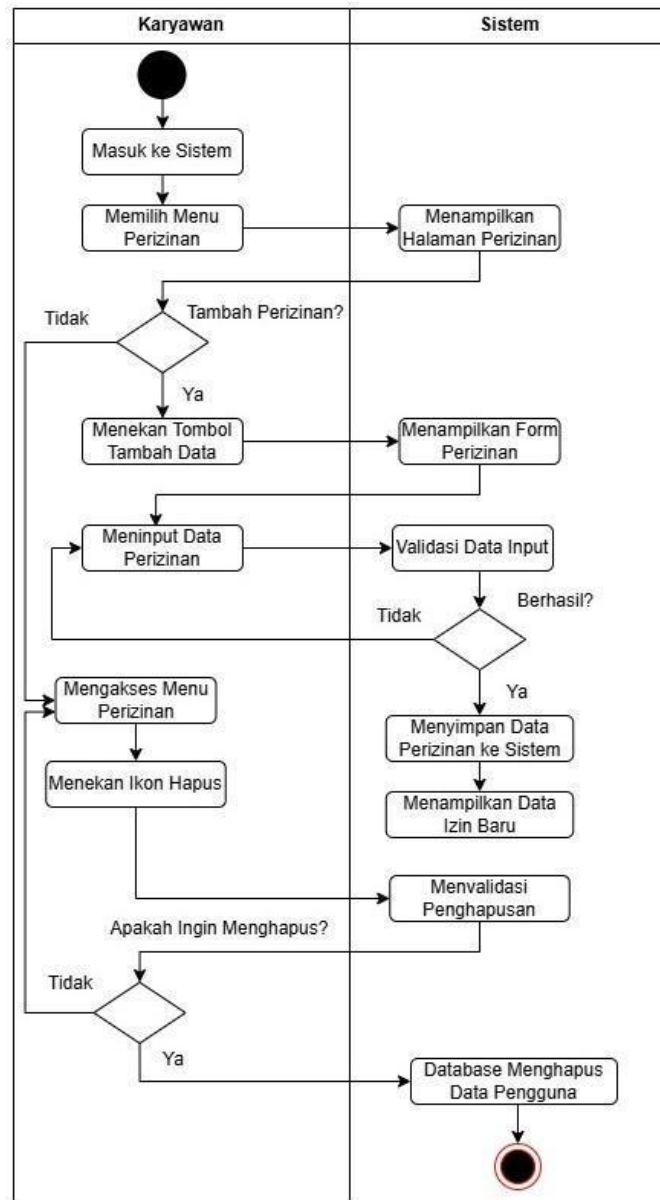
FR-009



Gambar 3.13 Activity Diagram Melakukan Absensi

Gambar 3.13 menjelaskan alur proses melakukan absensi, di mana karyawan masuk ke sistem dan membuka menu absensi. Setelah kamera aktif, sistem melakukan verifikasi wajah. Jika berhasil, sistem menyimpan data kehadiran ke *database* dan menampilkan detail absensi.

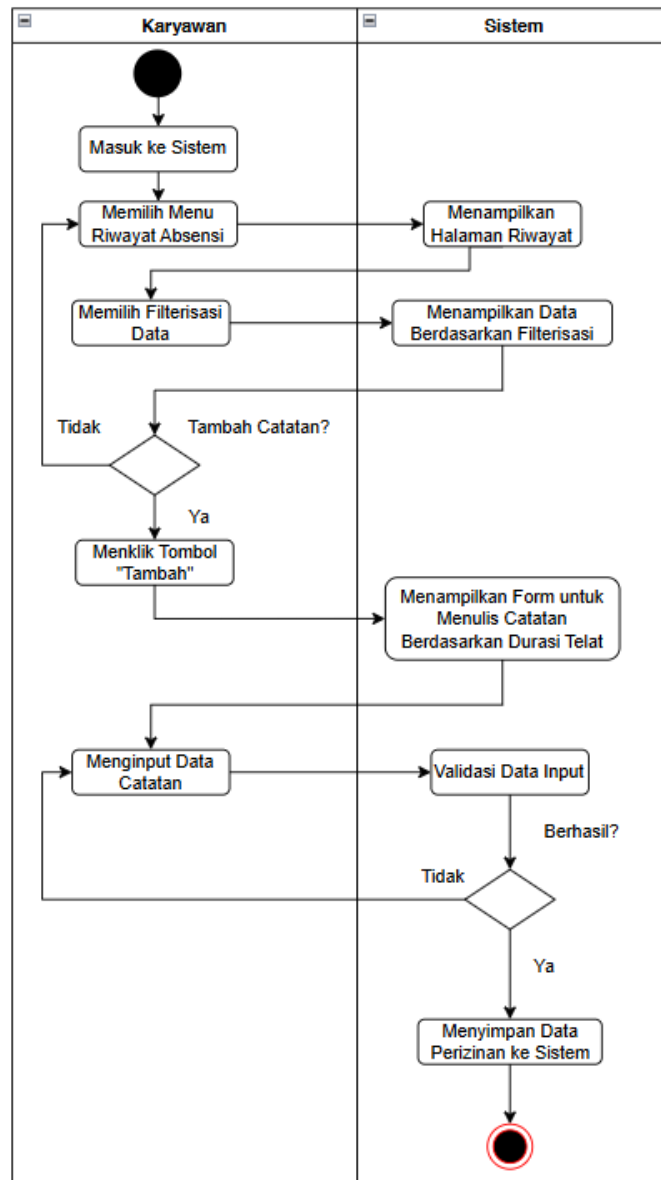
FR-010



Gambar 3.14 Activity Diagram Mengajukan Persetujuan

Gambar 3.14 menjelaskan alur proses mengajukan persetujuan, dimana karyawan masuk ke sistem dan memilih menu perizinan. Jika ingin menambah data izin, pengguna mengisi *form* perizinan, kemudian sistem memvalidasi dan menyimpan data ke sistem. Selain itu, pengguna juga dapat menghapus data izin melalui menu perizinan.

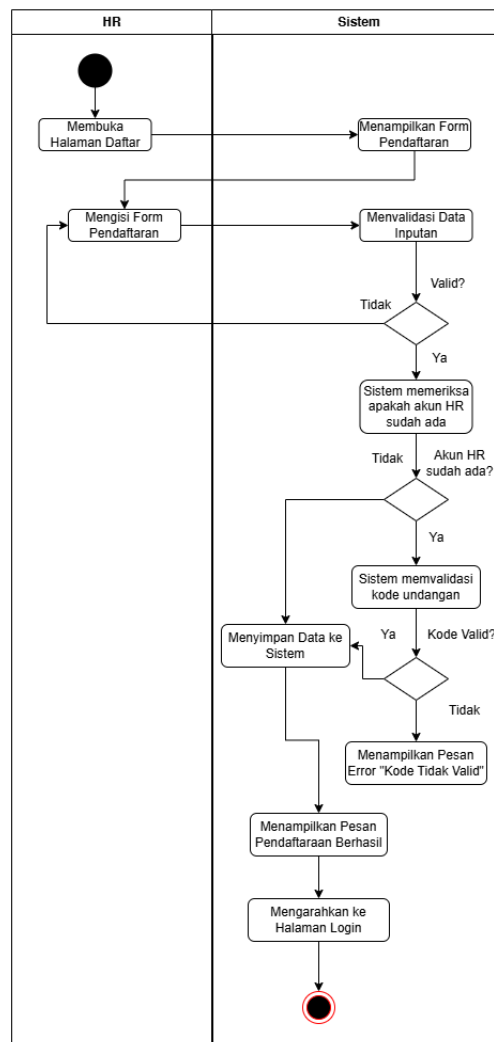
FR-011



Gambar 3.15 Activity Diagram Menambah Catatan Keterlambatan

Gambar 3.15 menjelaskan alur proses penambahan catatan keterlambatan, dimana karyawan masuk ke sistem dan memilih menu riwayat absensi. Jika ingin menambah catatan, pengguna mengisi form catatan keterlambatan, kemudian sistem memvalidasi dan menyimpan data ke sistem.

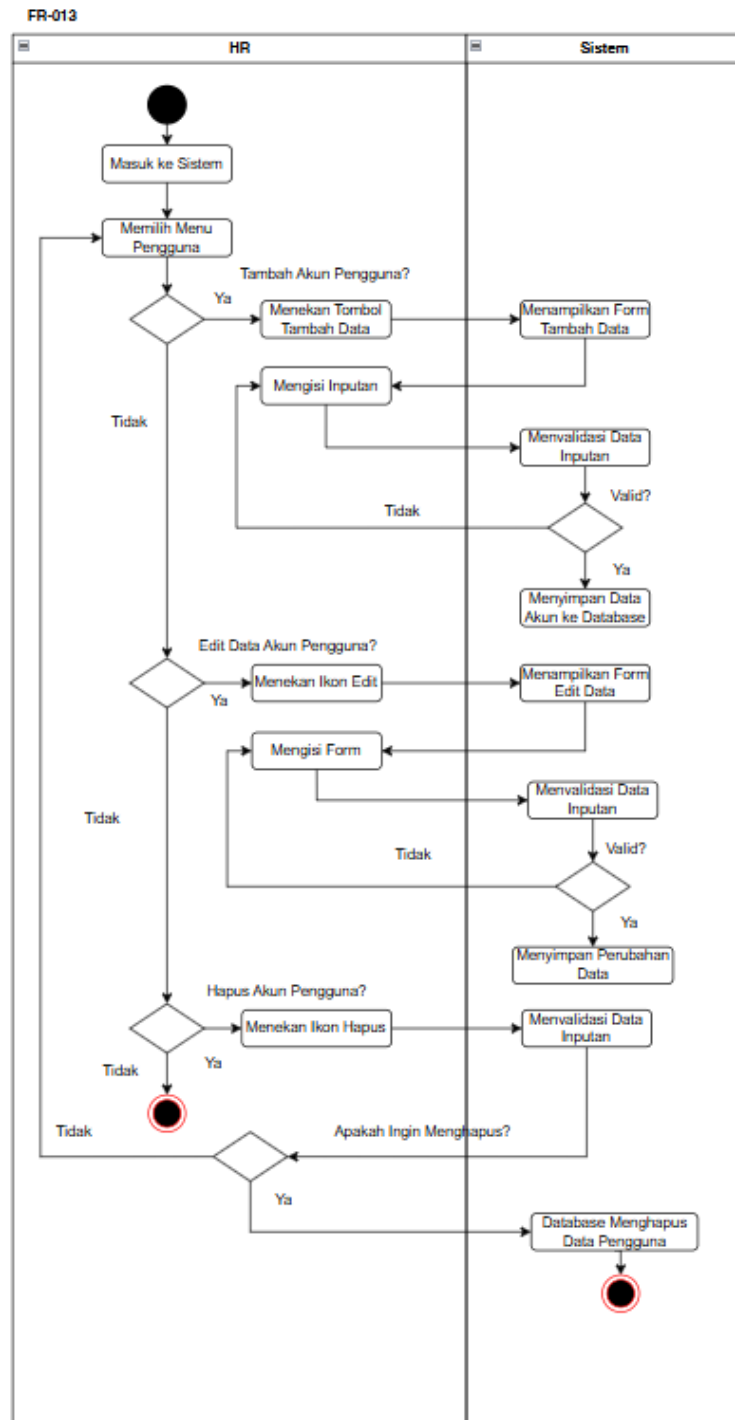
FR-012



Gambar 3.16 Activity Diagram Daftar

Gambar 3.16 menjelaskan alur proses pendaftaran akun HR. HR membuka halaman pendaftaran dan mengisi formulir yang tersedia. Sistem kemudian melakukan validasi terhadap data yang dimasukkan. Jika data tidak valid, sistem akan menampilkan pesan kesalahan dan meminta pengguna memperbaiki data. Jika data valid, sistem memeriksa apakah akun HR telah tersedia di dalam database. Apabila belum terdapat akun HR, sistem langsung menyimpan data pendaftaran. Namun, jika akun HR sudah tersedia, sistem akan melakukan validasi kode undangan yang dimasukkan. Jika kode undangan valid, data pendaftaran disimpan ke database dan pengguna diarahkan ke halaman login. Jika kode undangan tidak valid,

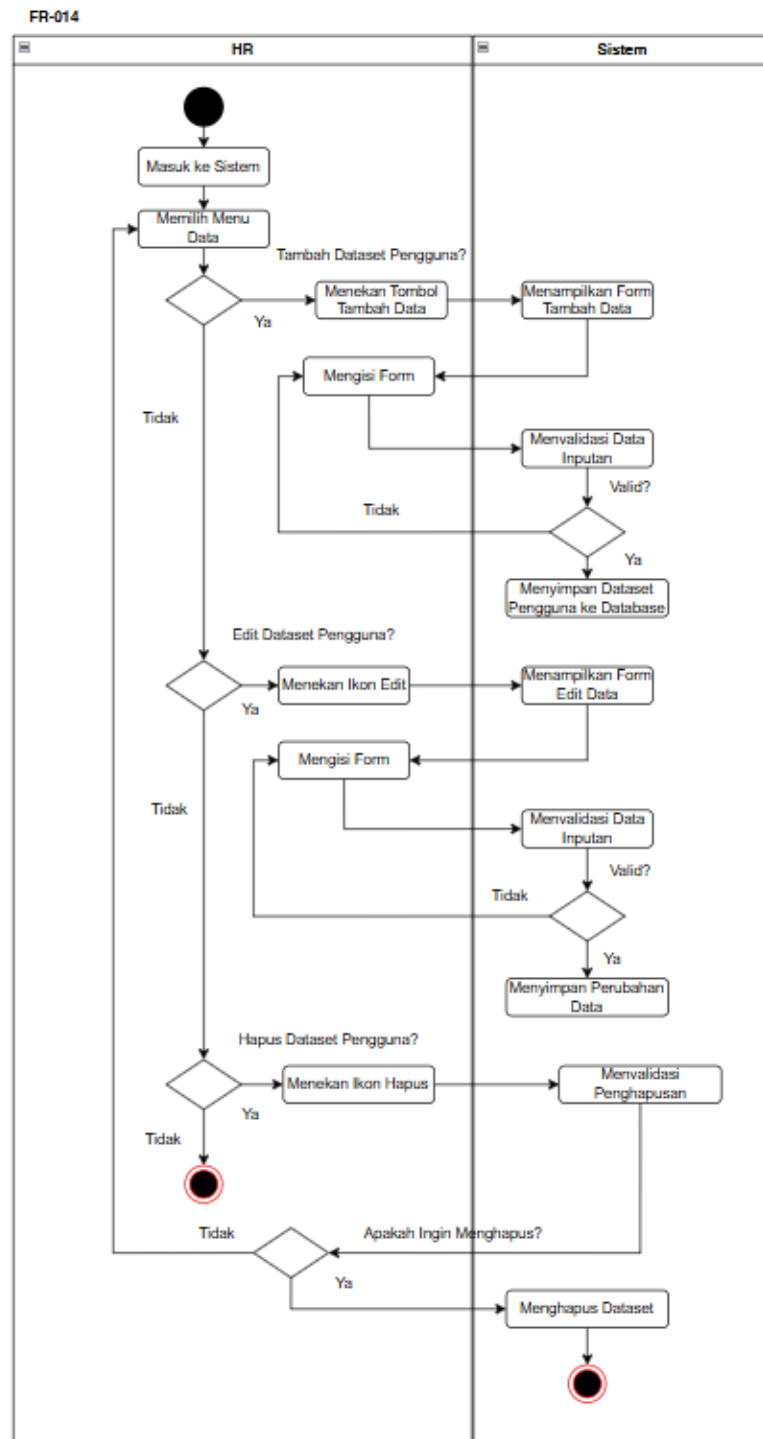
sistem menampilkan pesan kesalahan dan pengguna diminta mengulangi proses pendaftaran.



Gambar 3.17 Activity Diagram Mengelola Akun Pengguna

Gambar 3.17 menjelaskan alur proses mengelola akun pengguna, di mana HR masuk ke sistem dan memilih menu pengguna. HR dapat

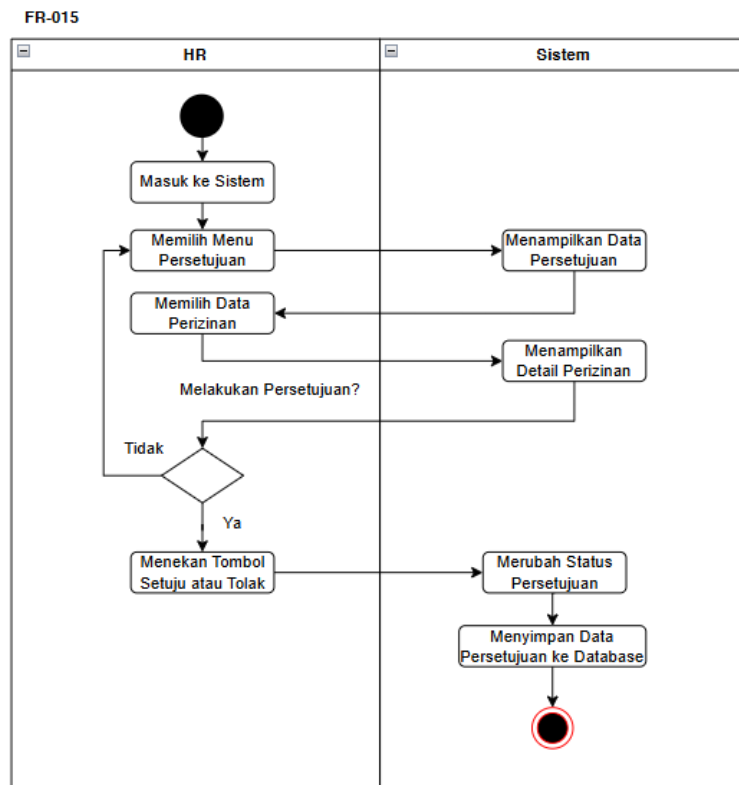
menambah, mengedit, atau menghapus akun. Setiap input akan divalidasi oleh sistem sebelum disimpan atau dihapus dari *database*.



Gambar 3.18 Activity Diagram Menambah *Dataset* Pengguna

Gambar 3.18 menjelaskan alur proses menambah *dataset* pengguna, di mana HR masuk ke sistem dan memilih menu data. HR dapat menambah, mengedit, atau menghapus *dataset* pengguna. Sistem akan melakukan

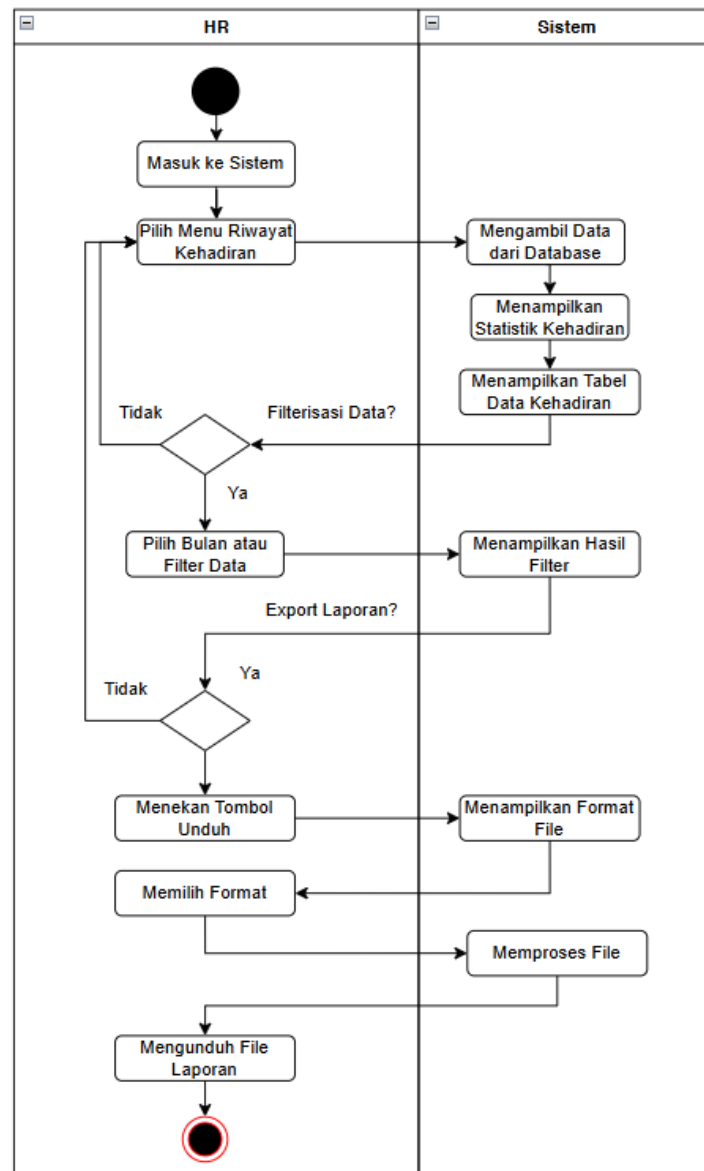
validasi terhadap setiap input dan menyimpan perubahan ke dalam *database* jika data valid.



Gambar 3.19 Activity Diagram Menangani Persetujuan

Gambar 3.19 menjelaskan alur proses menangani persetujuan, dimana HR masuk ke sistem dan memilih menu persetujuan. HR kemudian memilih data perizinan yang akan diproses. Jika melakukan persetujuan, sistem akan mengubah status perizinan dan menyimpan data ke *database*.

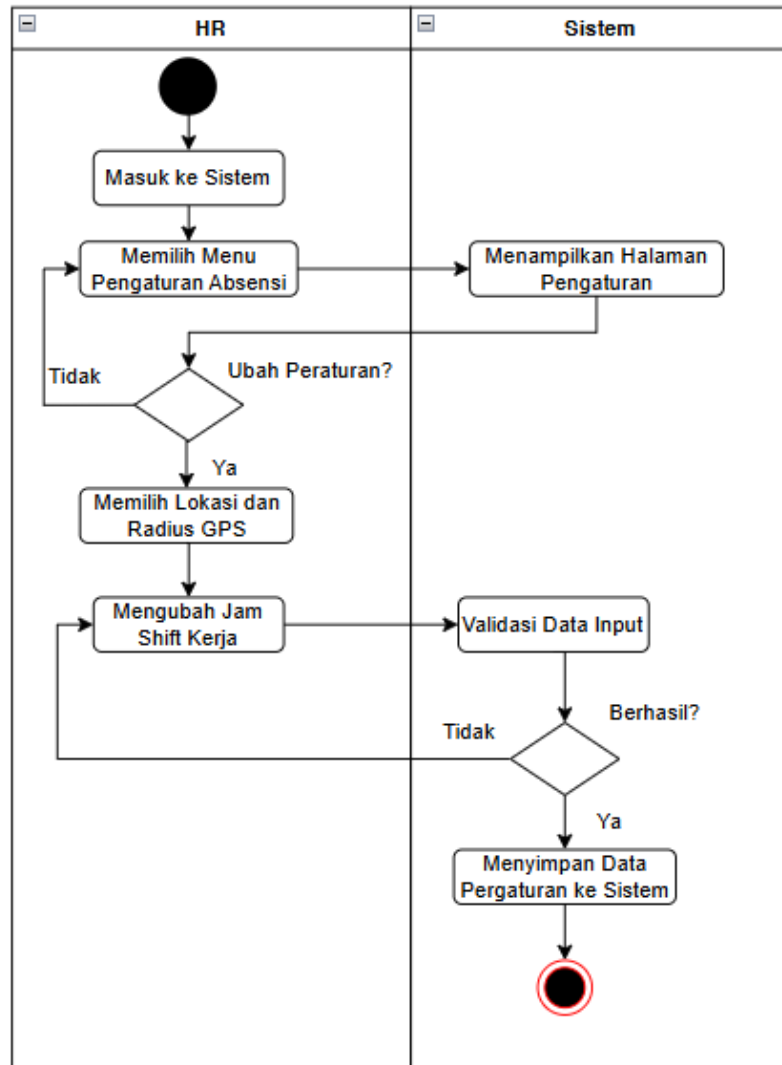
FR-016



Gambar 3.20 Activity Diagram *Export Laporan*

Gambar 3.20 menjelaskan alur proses *export* laporan, di mana HR masuk ke sistem dan memilih menu riwayat kehadiran. Sistem menampilkan data kehadiran dan memungkinkan HR melakukan filterisasi berdasarkan bulan atau kriteria tertentu. Setelah memilih format *file*, sistem memproses dan mengunduh laporan yang telah diekspor.

FR-017

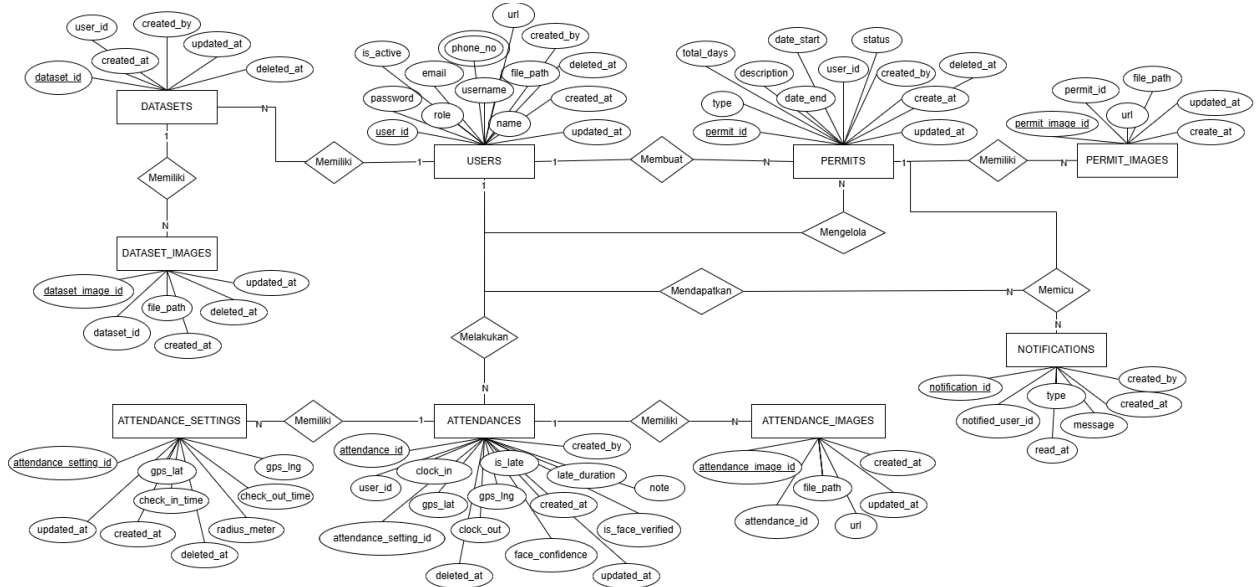


Gambar 3.21 Activity Diagram Mengelola Peraturan Absensi

Gambar 3.21 menjelaskan alur proses mengelola pengaturan absensi, dimana HR masuk ke sistem dan memilih menu pengaturan absensi. Jika ingin mengubah pengaturan, HR dapat memilih lokasi dan radius GPS serta mengubah jam shift kerja, kemudian sistem memvalidasi dan menyimpan data pengaturan ke sistem.

3.2.5 ER Diagram

Entity Relationship Diagram (ERD) yang terdapat pada sistem LOOK-IN dapat dilihat pada gambar berikut:



Gambar 3.22 ER Diagram

Gambar 3.22 Menunjukkan ER diagram pada sistem LOOK-IN. Diagram ini menggambarkan struktur basis data yang digunakan dalam sistem absensi berbasis face recognition pada Perusahaan Jasa Teknologi Informasi. Diagram ini memperlihatkan hubungan antar entitas yang saling terintegrasi untuk mendukung proses absensi, pengelolaan izin, pengelolaan dataset wajah, penyimpanan gambar pendukung, serta pengiriman notifikasi dalam sistem.

Diagram ini menjelaskan bahwa sistem memiliki beberapa entitas utama, yaitu *Users*, *Attendances*, *Attendance_Settings*, *Attendance_Images*, *Permits*, *Permit_Images*, *Notifications*, *Datasets*, dan *Dataset_Images*. Setiap entitas saling berhubungan untuk mendukung proses absensi berbasis face recognition, pengelolaan izin, penyimpanan dataset wajah, serta pengiriman notifikasi dalam sistem. Berikut penjelasan tiap entitas dan relasinya:

1. **Users** : merupakan entitas utama yang merepresentasikan seluruh pengguna sistem baik karyawan maupun HR. Atribut pada entitas

ini meliputi *user_id* sebagai *primary key*, *name*, *username*, *email*, *password*, *role*, *phone_no*, *is_active*, *file_path*, *url*, *created_at*, *updated_at*, *deleted_at*, dan *created_by*. Entitas *Users* berelasi *one-to-many* dengan *Attendances*, *Permits*, *Notifications*, dan *Datasets* karena satu pengguna dapat memiliki banyak data absensi, pengajuan izin, notifikasi, dan dataset wajah yang digunakan dalam proses *face recognition* pada sistem.

2. *Attendances* : digunakan untuk mencatat data kehadiran karyawan yang diperoleh dari proses verifikasi wajah dan geolokasi. Atributnya terdiri dari *attendance_id* sebagai *primary key*, *timestamp*, *status*, *type*, *gps_lat*, *gps_lng*, *file_path*, *created_at*, dan *created_by*. Entitas ini berelasi *many-to-one* dengan *Users*, karena setiap catatan kehadiran hanya dimiliki oleh satu pengguna. Data pada entitas ini menjadi bukti kehadiran yang diverifikasi melalui teknologi *face recognition*.
3. *Attendance Images* : digunakan untuk mencatat data kehadiran pengguna yang diperoleh dari proses absensi berbasis *face recognition* dan geolokasi. Atribut pada entitas ini terdiri dari *attendance_id* sebagai *primary key*, *user_id*, *attendance_setting_id*, *clock_in*, *clock_out*, *gps_lat*, *gps_lng*, *is_late*, *late_duration*, *note*, *face_confidence*, *is_face_verified*, *created_at*, *updated_at*, *deleted_at*, dan *created_by*. Entitas *Attendances* berelasi *many-to-one* dengan *Users* karena setiap data absensi dimiliki oleh satu pengguna, serta berelasi *many-to-one* dengan *Attendance_Settings* karena setiap absensi menggunakan satu pengaturan absensi tertentu. Selain itu, entitas ini juga berelasi *one-to-many* dengan *Attendance_Images* karena satu data absensi dapat memiliki beberapa gambar pendukung sebagai bukti kehadiran.
4. *Attendance Settings* : digunakan untuk menyimpan pengaturan absensi yang diterapkan pada sistem. Atribut pada entitas ini meliputi *attendance_setting_id* sebagai *primary key*, *gps_lat*, *gps_lng*, *radius_meter*, *check_in_time*, *check_out_time*, *created_at*,

updated_at, *deleted_at*, dan *created_by*. Entitas *Attendance_Settings* berelasi *one-to-many* dengan *Attendances* karena satu pengaturan absensi dapat digunakan oleh banyak data absensi pengguna. Entitas ini berfungsi untuk mengatur lokasi absensi, radius validasi GPS, serta waktu masuk dan pulang yang digunakan dalam proses pencatatan kehadiran.

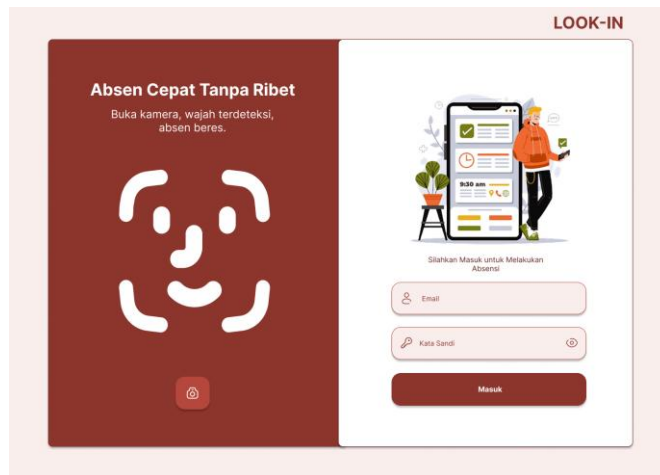
5. *Permits* : berfungsi untuk menyimpan data pengajuan izin yang diajukan oleh pengguna dalam sistem. Atribut pada entitas ini meliputi *permit_id* sebagai *primary key*, *user_id*, *description*, *type*, *status*, *date_start*, *date_end*, *total_day*, *created_at*, *updated_at*, *deleted_at*, dan *created_by*. Entitas *Permits* berelasi *many-to-one* dengan *Users* karena setiap pengguna dapat memiliki banyak pengajuan izin. Selain itu, entitas ini juga berelasi *one-to-many* dengan *Permit_Images* karena satu data izin dapat memiliki beberapa gambar atau dokumen pendukung sebagai bukti pengajuan izin.
6. *Permit Images* : digunakan untuk menyimpan gambar atau dokumen pendukung pada pengajuan izin pengguna. Atribut pada entitas ini meliputi *permit_image_id* sebagai *primary key*, *permit_id*, *file_path*, *url*, *created_at*, *updated_at*, *deleted_at*, dan *created_by*. Entitas *Permit_Images* berelasi *many-to-one* dengan *Permits* karena satu data izin dapat memiliki beberapa file gambar atau dokumen pendukung yang digunakan sebagai bukti dalam proses pengajuan izin pada sistem.
7. *Notifications* : digunakan untuk menyimpan pesan pemberitahuan kepada pengguna, seperti status izin, hasil verifikasi absensi, atau informasi sistem lainnya. Atributnya meliputi *notification_id* sebagai *primary key*, *type*, *message*, *read_at*, *created_at*, *created_by*, dan *notified_user_id*. Entitas ini berelasi *many-to-one* dengan *Users*, karena satu pengguna dapat menerima banyak notifikasi dari berbagai aktivitas sistem.

8. *Datasets* : berisi kumpulan data wajah yang digunakan dalam proses pelatihan dan pengenalan wajah pada sistem *face recognition*. Atribut pada entitas ini terdiri dari *dataset_id* sebagai *primary key*, *user_id*, *created_at*, *updated_at*, *deleted_at*, dan *created_by*. Entitas *Datasets* berelasi *many-to-one* dengan *Users* karena setiap *dataset* wajah dimiliki oleh satu pengguna. Selain itu, entitas ini juga berelasi *one-to-many* dengan *Dataset_Images* karena satu *dataset* dapat memiliki banyak gambar wajah yang digunakan sebagai data pelatihan sistem pengenalan wajah.
9. *Dataset Images* : digunakan untuk menyimpan gambar wajah yang menjadi bagian dari *dataset* pelatihan pada sistem *face recognition*. Atribut pada entitas ini terdiri dari *dataset_image_id* sebagai *primary key*, *dataset_id*, *file_path*, *url*, *created_at*, *updated_at*, *deleted_at*, dan *created_by*. Entitas *Dataset_Images* berelasi *many-to-one* dengan *Datasets* karena satu *dataset* wajah dapat memiliki banyak gambar wajah yang digunakan sebagai data pelatihan dan proses pengenalan wajah pada sistem.

Secara keseluruhan, ERD ini menggambarkan bahwa sistem absensi berbasis face recognition di Perusahaan Jasa Teknologi Informasi dirancang secara terintegrasi, di mana entitas *Users* menjadi pusat utama yang menghubungkan proses absensi, pengelolaan izin, pengaturan absensi, notifikasi, serta dataset wajah dan gambar pendukung. Struktur basis data ini dirancang untuk memastikan proses pencatatan kehadiran, verifikasi wajah, pengelolaan dokumen, dan penyampaian informasi dapat berjalan secara efisien, terstruktur, aman, dan akurat dalam mendukung operasional sistem absensi karyawan.

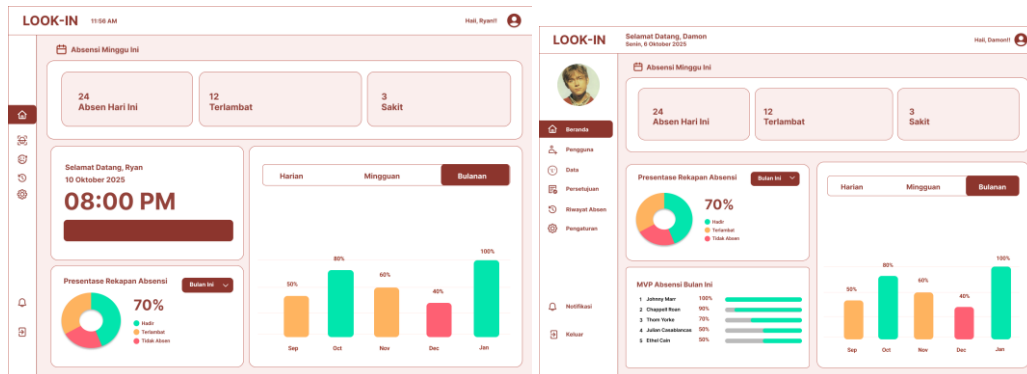
3.2.5 Perancangan Antarmuka (*Wireframe*)

Perancangan antarmuka yang terdapat pada sistem LOOK-IN dapat dilihat pada gambar berikut:



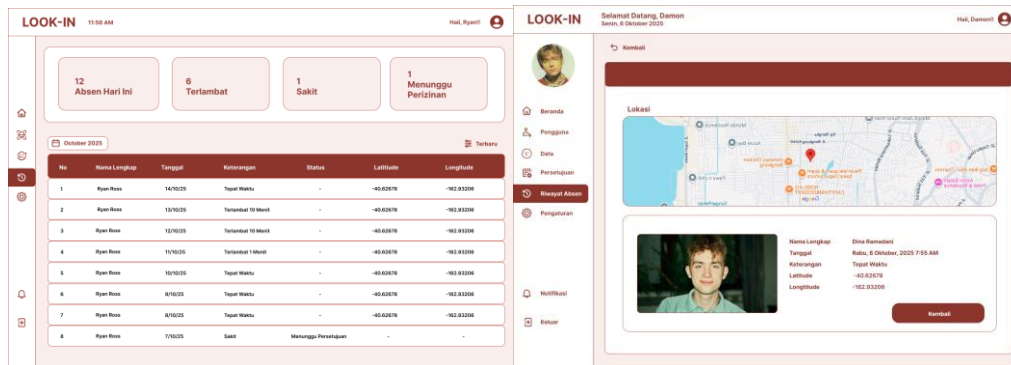
Gambar 3.23 *Wireframe* Halaman Masuk

Pada Gambar 3.23 menampilkan *wireframe* halaman masuk yang berisi kolom *email*, *kata sandi*, dan tombol *masuk*. Halaman ini dirancang sederhana agar pengguna dapat dengan mudah mengakses sistem sebelum melakukan absensi berbasis *face recognition*.



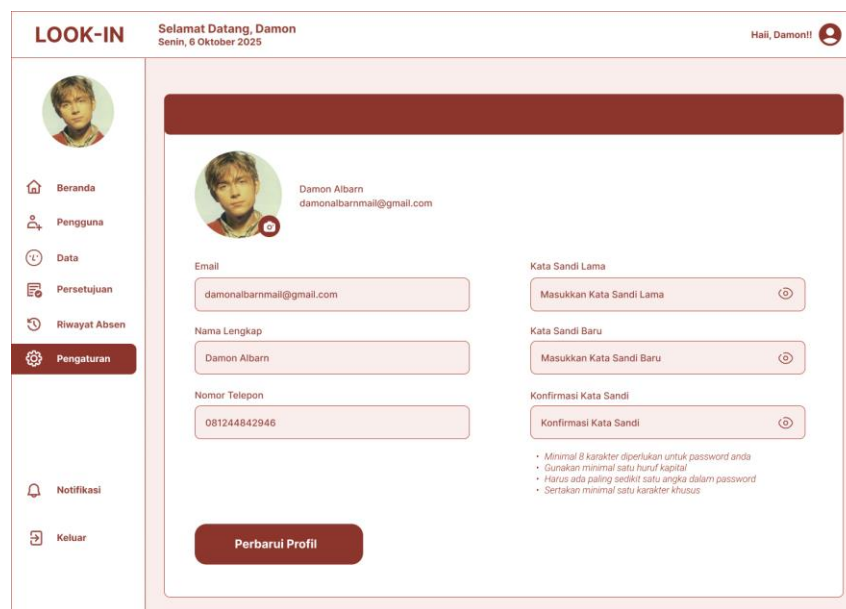
Gambar 3.24 *Wireframe* Halaman Beranda

Pada Gambar 3.24 menampilkan *wireframe* halaman beranda yang digunakan untuk menampilkan ringkasan data absensi karyawan. Halaman ini menampilkan informasi seperti jumlah absen hari ini, keterlambatan, izin sakit, serta grafik persentase kehadiran mingguan dan bulanan untuk memudahkan pemantauan kehadiran secara *real-time*.



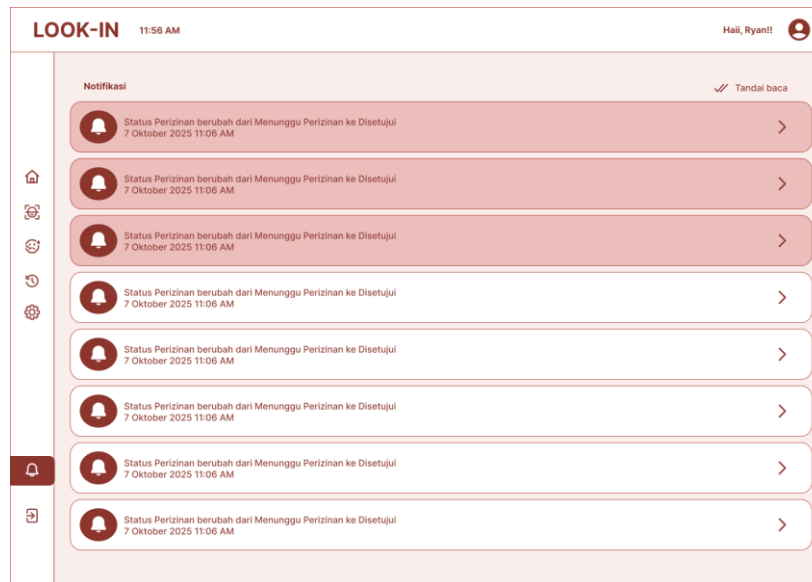
Gambar 3.25 Wireframe Halaman Riwayat Kehadiran

Pada Gambar 3.25 menampilkan *wireframe* halaman riwayat kehadiran yang berfungsi untuk menampilkan data absensi karyawan secara detail. Halaman ini berisi daftar kehadiran lengkap dengan tanggal, status, serta lokasi absensi yang ditampilkan melalui peta untuk memverifikasi posisi karyawan saat melakukan absensi.



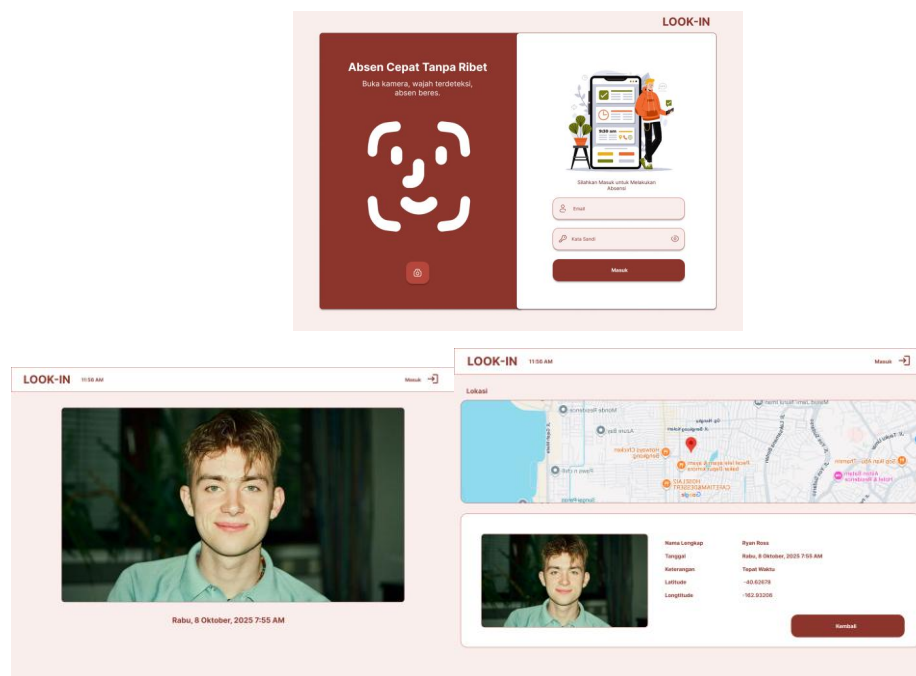
Gambar 3.26 Wireframe Halaman Mengelola Profil dan Mengubah Kata Sandi

Pada Gambar 3.26 menampilkan *wireframe* halaman mengelola profil dan mengubah kata sandi yang memungkinkan pengguna memperbarui data pribadi seperti email, nama, dan nomor telepon, serta mengganti kata sandi lama dengan yang baru untuk menjaga keamanan akun.



Gambar 3.27 *Wireframe* Halaman Notifikasi

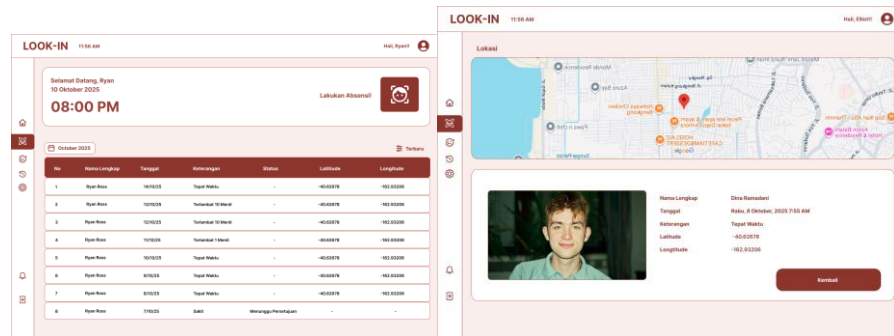
Pada Gambar 3.27 menampilkan *wireframe* halaman notifikasi yang berfungsi untuk menampilkan informasi terkait aktivitas pengguna, seperti status pengajuan izin yang telah disetujui atau ditolak. Halaman ini membantu pengguna memantau pembaruan sistem secara cepat dan efisien.



Gambar 3.28 *Wireframe* Halaman Absensi Cepat

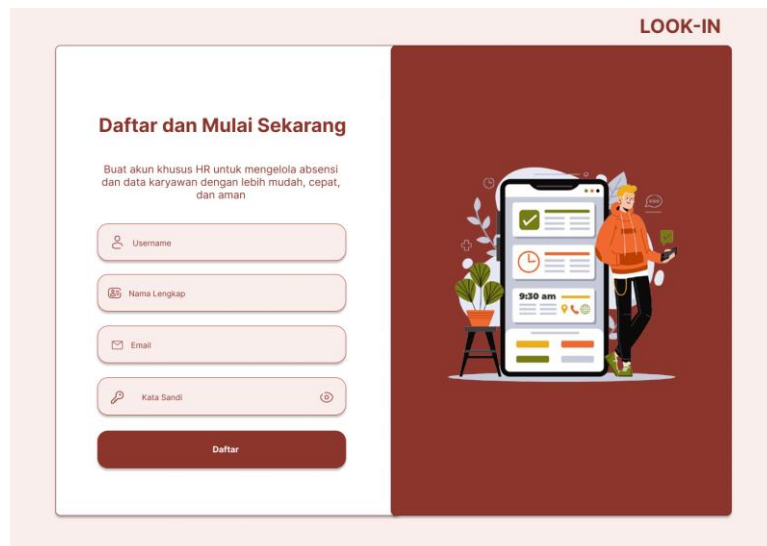
Pada Gambar 3.28 menampilkan *wireframe* halaman absensi cepat yang digunakan untuk proses absensi otomatis berbasis *face recognition*. Halaman ini menampilkan hasil deteksi wajah pengguna beserta lokasi

absensi pada peta, sehingga memastikan kehadiran tercatat secara akurat dan sesuai dengan posisi sebenarnya.



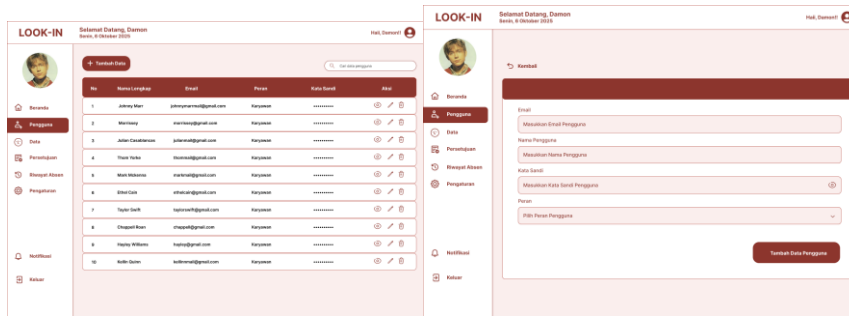
Gambar 3.29 Wireframe Halaman Absensi

Pada Gambar 3.29 menampilkan *wireframe* halaman absensi yang digunakan untuk menampilkan daftar kehadiran karyawan serta proses absensi secara langsung. Halaman ini menampilkan waktu absensi, foto hasil deteksi wajah, dan lokasi pengguna pada peta untuk memastikan keakuratan data kehadiran.



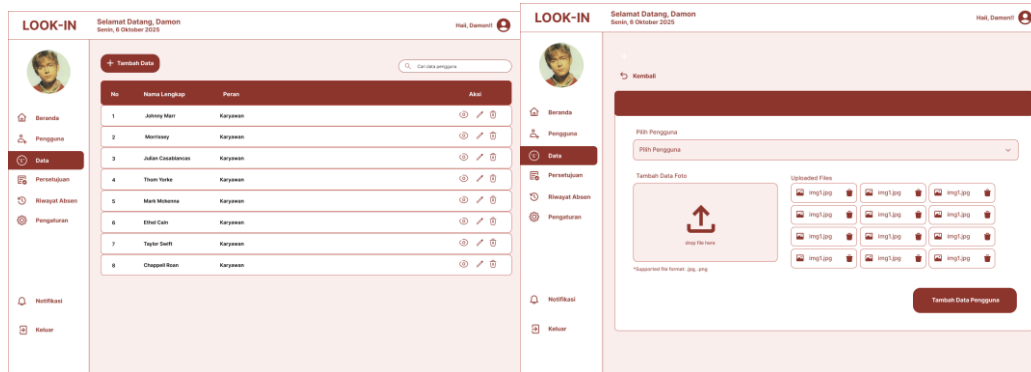
Gambar 3.30 Wireframe Halaman Daftar

Pada Gambar 3.30 menampilkan *wireframe* halaman daftar yang digunakan khusus oleh HR untuk membuat akun baru. Halaman ini berisi *form* pendaftaran dengan kolom *username*, *nama lengkap*, *email*, dan *kata sandi* yang memungkinkan HR mengelola data absensi dan pengguna dalam sistem.



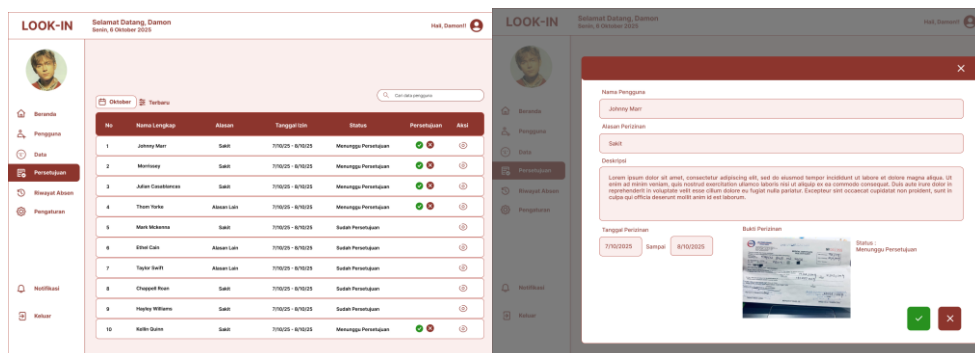
Gambar 3.31 Wireframe Halaman Mengelola Pengguna

Pada Gambar 3.31 menampilkan *wireframe* halaman mengelola pengguna yang digunakan oleh HR untuk menambah, mengubah, dan menghapus akun karyawan dalam sistem.



Gambar 3.32 Wireframe Halaman Mengelola Dataset

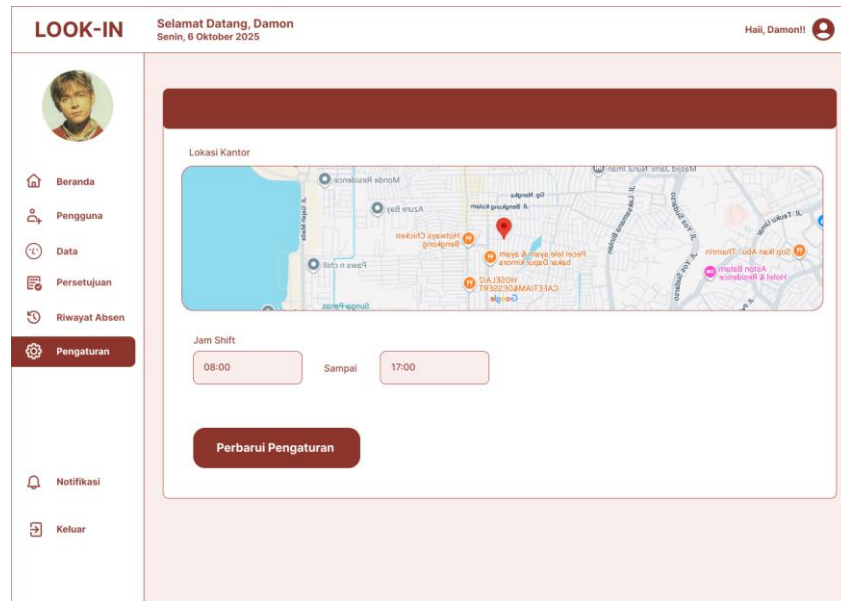
Pada Gambar 3.32 menampilkan *wireframe* halaman mengelola *dataset* yang digunakan oleh HR untuk menambahkan dan mengatur data wajah karyawan yang akan digunakan sebagai data *training* pada model *face recognition*.



Gambar 3.33 Wireframe Halaman Menangani Persetujuan

Pada Gambar 3.33 menampilkan *wireframe* halaman menangani persetujuan yang digunakan oleh HR untuk memeriksa dan memverifikasi pengajuan izin sakit dari karyawan. Pada halaman ini, HR dapat melihat

detail data karyawan, alasan izin, serta lampiran bukti pendukung seperti surat keterangan dokter. Selain itu, HR juga dapat memberikan keputusan berupa persetujuan atau penolakan terhadap pengajuan tersebut secara langsung melalui sistem.



Gambar 3.34 *Wireframe* Halaman Mengelola Peraturan Absensi

Pada Gambar 3.34 menampilkan *wireframe* halaman pengaturan absensi yang digunakan oleh HR untuk mengatur lokasi dan jam kerja pada sistem absensi. Pada halaman ini, HR dapat melihat lokasi kantor melalui peta, mengatur jam masuk dan jam pulang kerja, serta memperbarui pengaturan absensi sesuai kebutuhan perusahaan. Pengaturan ini digunakan sebagai acuan dalam proses validasi absensi berbasis GPS dan *face recognition* pada sistem.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Hasil Implementasi

Proses implementasi sistem dilakukan setelah tahap analisis kebutuhan dan perancangan selesai dikerjakan. Pada tahap ini, sistem absensi berbasis *face recognition* mulai dibangun dengan menyiapkan lingkungan pengembangan, *database*, *backend*, *frontend*, serta model AI agar seluruh komponen dapat saling terhubung dan berjalan dengan baik.

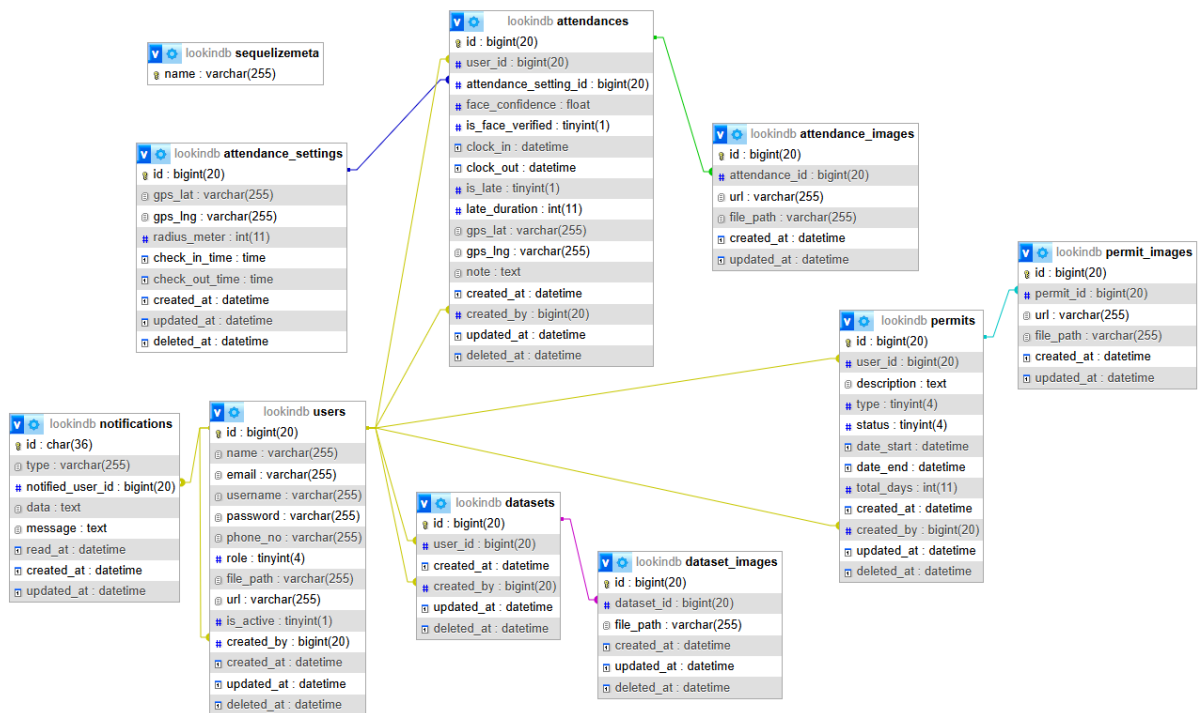
Pada bagian *backend*, *framework NestJS* digunakan untuk mengembangkan API yang berfungsi sebagai penghubung antara *frontend* dengan *database*. Pembuatan database dilakukan menggunakan MySQL dengan bantuan *Sequelize* ORM dan migration untuk membangun struktur tabel seperti *users*, *attendances*, *permits*, *datasets*, *notifications*, dan *attendance settings*. Setelah seluruh endpoint selesai dibuat, pengujian API dilakukan menggunakan *Postman* untuk memastikan setiap fitur dapat berjalan sesuai kebutuhan sistem.

Pada sisi *frontend*, sistem dikembangkan menggunakan *Vue.js* untuk membangun tampilan antarmuka website yang digunakan oleh HR maupun karyawan. *Frontend* dihubungkan dengan backend menggunakan API sehingga data absensi, izin, notifikasi, dan pengaturan dapat ditampilkan secara *real-time*. Selama proses pengembangan, *frontend* dijalankan menggunakan port 5173.

Selain itu, sistem juga mengimplementasikan fitur *face recognition* menggunakan *Python* dan *library OpenCV*. Metode *Haar Cascade* digunakan untuk mendeteksi wajah, sedangkan metode *Local Binary Pattern Histogram* (LBPH) digunakan untuk proses pengenalan wajah pengguna. Service AI dijalankan secara terpisah menggunakan port 8000 dan digunakan untuk proses *training dataset* wajah serta verifikasi identitas pengguna ketika melakukan absensi.

4.1.1 Implementasi Basis Data

Implementasi basis data yang terdapat pada sistem LOOK-IN dapat dilihat pada gambar berikut:



Gambar 4.1 Implementasi Basis Data

Gambar 4.1 menunjukkan implementasi basis data pada sistem LOOK-IN yang menggunakan MySQL dan Sequelize ORM pada *framework NestJS*. Basis data ini digunakan untuk mendukung proses absensi berbasis *face recognition*, pengelolaan pengguna, izin, dataset wajah, dan notifikasi sistem.

Pada implementasinya terdapat beberapa tabel utama seperti *users*, *attendances*, *attendance_settings*, *attendance_images*, *permits*, *permit_images*, *datasets*, *dataset_images*, *notifications*, dan *sequelizemeta*. Seluruh tabel saling terhubung menggunakan *foreign key* sehingga proses pengelolaan data dapat berjalan secara terintegrasi, terstruktur, dan efisien.

4.1.2 Implementasi API

Implementasi API pada sistem LOOK-IN digunakan sebagai penghubung antara *frontend*, *backend*, *database*, serta modul *face recognition* agar proses pertukaran data dapat berjalan secara *real-time*. *Backend* sistem dikembangkan menggunakan *NestJS* dan layanan AI menggunakan *FastAPI* dengan pendekatan REST API. Setiap respons API menggunakan format JSON yang memuat informasi seperti *message*, *data*,

token, dan status agar mudah dibaca serta diintegrasikan dengan frontend berbasis web. Implementasi API menggunakan *Postman* yang terdapat pada sistem LOOK-IN dapat dilihat pada gambar berikut:

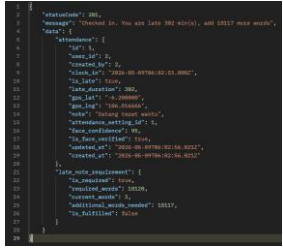
Tabel 4.1 Implementasi API

No	Folders	Endpoints	Methods	Responses
1	Auth	<i>/api/v1/auth/profile</i>	GET	<pre> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 26</pre>

		/api/v1/user/{id}/image	POST	<pre> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 "statusCode": 200, "message": "Successfully update photo profile", "data": { "role_name": "Hiring Manager", "id": 1, "name": "Hiring Manager", "email": "hiringmanager@gmail.com", "username": "hiring manager", "phone_no": null, "role": 1, "file_path": null, "url": null, "is_active": true, "created_by": null, "created_at": "2020-06-07T12:09:18.000Z", "updated_at": "2020-06-07T12:09:18.000Z", "deleted_at": null } </pre>
		/api/v1/user	PUT	<pre> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 "statusCode": 200, "message": "Successfully updated user", "data": { "user": { "role_name": "Hiring Manager", "id": 1, "name": "Hiring Manager 1", "email": "hiringmanager@gmail.com", "username": "Hiring manager", "phone_no": "9876543210", "role": 1, "file_path": null, "url": null, "is_active": true, "created_by": null, "created_at": "2020-06-07T12:09:18.000Z", "updated_at": "2020-06-07T12:09:18.000Z", "deleted_at": null } } </pre>
		/api/v1/user/change-password	PUT	<pre> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 "statusCode": 200, "message": "Successfully change password", "data": { "role_name": "Hiring Manager", "id": 1, "name": "Hiring Manager 1", "email": "hiringmanager@gmail.com", "username": "Hiring manager", "password": "123456789", "role": 1, "file_path": null, "url": null, "is_active": true, "created_by": null, "created_at": "2020-06-07T12:09:18.000Z", "updated_at": "2020-06-07T12:09:18.000Z", "deleted_at": null } </pre>
		/api/v1/user/{id}	DEL	<pre> 1 2 3 4 5 "statusCode": 200, "message": "Successfully delete user", "data": {} </pre>
3	Dataset	/api/v1/dataset	GET	<pre> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 "statusCode": 200, "message": "Successfully retrieved datasets", "data": { "count": 1, "datasets": [{ "id": 1, "user": { "id": 2, "name": "Johnny Marx" }, "image_count": 1, "created_at": "2020-06-07T14:17:35.000Z" }] } </pre>
		/api/v1/dataset/{id}	GET	<pre> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 "statusCode": 200, "message": "Successfully get dataset", "data": { "dataset": { "id": 1, "user": { "id": 2, "name": "Johnny Marx" }, "image_count": 1, "created_at": "2020-06-07T14:17:35.000Z" } } </pre>
		/api/v1/dataset	POST	<pre> 1 2 3 4 5 6 7 8 9 10 "statusCode": 201, "message": "Successfully created dataset", "data": { "id": 1, "user_id": 2, "created_by": 1, "image_count": 1 } </pre>
		/api/v1/dataset/{id}	DEL	<pre> 1 2 3 4 5 "statusCode": 200, "message": "Successfully delete dataset", "data": {} </pre>
4	Dataset Image	/api/v1/dataset	GET	<pre> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 "statusCode": 200, "message": "Successfully retrieved datasets", "data": { "count": 1, "datasets": [{ "id": 1, "user": { "id": 2, "name": "Johnny Marx" }, "image_count": 1, "created_at": "2020-06-07T14:17:35.000Z" }] } </pre>

5	Attendance Setting	/api/v1/attendance-setting	GET	<pre> 1 { 2 "statusCode": 200, 3 "message": "Successfully get attendance settings", 4 "data": [5 { 6 "id": 1, 7 "gps_lat": "-0.200000", 8 "gps_lng": "100.016666", 9 "radius_meter": 100, 10 "check_in_time": "00:00:00", 11 "check_out_time": "17:00:00", 12 "created_at": "2020-05-07T14:44:55.000Z", 13 "updated_at": "2020-05-07T14:44:55.000Z", 14 "deleted_at": null 15 } 16] 17 } </pre>
		/api/v1/attendance-setting	GET	<pre> 1 { 2 "statusCode": 200, 3 "message": "Successfully get attendance setting", 4 "data": { 5 "id": 1, 6 "gps_lat": "-0.200000", 7 "gps_lng": "100.016666", 8 "radius_meter": 100, 9 "check_in_time": "00:00:00", 10 "check_out_time": "17:00:00", 11 "created_at": "2020-05-07T14:44:55.000Z", 12 "updated_at": "2020-05-07T14:44:55.000Z", 13 "deleted_at": null 14 } 15 } </pre>
		/api/v1/attendance-setting	POST	<pre> 1 { 2 "statusCode": 201, 3 "message": "Successfully create attendance setting", 4 "data": { 5 "id": 1, 6 "gps_lat": "-0.200000", 7 "gps_lng": "100.016666", 8 "radius_meter": 100, 9 "check_in_time": "00:00:00", 10 "check_out_time": "17:00:00", 11 "created_at": "2020-05-07T14:44:55.000Z", 12 "updated_at": "2020-05-07T14:44:55.000Z" 13 } 14 } </pre>
		/api/v1/attendance-setting/{id}	PUT	<pre> 1 { 2 "statusCode": 200, 3 "message": "Successfully update attendance setting", 4 "data": { 5 "id": 1, 6 "gps_lat": "-0.200000", 7 "gps_lng": "100.016666", 8 "radius_meter": 0, 9 "check_in_time": "00:00:00", 10 "check_out_time": "17:00:00", 11 "created_at": "2020-05-07T14:44:55.000Z", 12 "updated_at": "2020-05-07T14:49:28.130Z", 13 "deleted_at": null 14 } 15 } </pre>
		/api/v1/attendance-setting/{id}	DEL	<pre> 1 { 2 "statusCode": 200, 3 "message": "Successfully delete attendance setting", 4 "data": {} 5 } </pre>
6	Permit	/api/v1/permit	GET	<pre> 1 { 2 "statusCode": 200, 3 "message": "Successfully get permit", 4 "data": { 5 "permit_name": "Permit", 6 "type": "Permit", 7 "area_id": 1, 8 "description": "Right to use a part of the road for a vehicle", 9 "radius": 0, 10 "type_id": "000-00-000000-0000", 11 "area_name": "000-00-000000-0000", 12 "created_at": "2020-05-07T14:44:55.000Z", 13 "updated_at": "2020-05-07T14:44:55.000Z", 14 "deleted_at": null, 15 "area_name": "Permit", 16 "type": "Permit", 17 "radius": "0000000000000000", 18 "description": "Right to use a part of the road for a vehicle", 19 "radius": 0, 20 "type_id": "000-00-000000-0000", 21 "area_name": "000-00-000000-0000", 22 "created_at": "2020-05-07T14:44:55.000Z", 23 "updated_at": "2020-05-07T14:44:55.000Z", 24 "deleted_at": null 25 } 26 } </pre>
		/api/v1/permit/{id}	GET	<pre> 1 { 2 "statusCode": 200, 3 "message": "Successfully get permit", 4 "data": { 5 "permit_name": "Permit", 6 "type": "Permit", 7 "area_id": 1, 8 "description": "Right to use a part of the road for a vehicle", 9 "radius": 0, 10 "type_id": "000-00-000000-0000", 11 "area_name": "000-00-000000-0000", 12 "created_at": "2020-05-07T14:44:55.000Z", 13 "updated_at": "2020-05-07T14:44:55.000Z", 14 "deleted_at": null 15 } 16 } </pre>
		/api/v1/permit	POST	<pre> 1 { 2 "statusCode": 201, 3 "message": "Successfully create permit", 4 "data": { 5 "permit_name": "Permit", 6 "type": "Permit", 7 "area_id": 1, 8 "description": "Right to use a part of the road for a vehicle", 9 "radius": 0, 10 "type_id": "000-00-000000-0000", 11 "area_name": "000-00-000000-0000", 12 "created_at": "2020-05-07T14:44:55.000Z", 13 "updated_at": "2020-05-07T14:44:55.000Z", 14 "deleted_at": null 15 } 16 } </pre>
		/api/v1/permit/{id}	PUT	<pre> 1 { 2 "statusCode": 200, 3 "message": "Successfully update permit", 4 "data": { 5 "permit_name": "Permit", 6 "type": "Permit", 7 "area_id": 1, 8 "description": "Right to use a part of the road for a vehicle", 9 "radius": 0, 10 "type_id": "000-00-000000-0000", 11 "area_name": "000-00-000000-0000", 12 "created_at": "2020-05-07T14:44:55.000Z", 13 "updated_at": "2020-05-07T14:44:55.000Z", 14 "deleted_at": null 15 } 16 } </pre>

		/api/v1/permit/{id}	PUT	<pre> 1 { 2 "message": "Successfully update permit", 3 "data": { 4 "id": 1, 5 "status_name": "Approved", 6 "perm_name": "Permit", 7 "id": 1, 8 "created_at": "2020-05-07T10:02:53.000Z", 9 "updated_at": "2020-05-07T10:02:53.000Z", 10 "deleted_at": null, 11 "type": 1, 12 "clock_in": "2020-05-07T09:00:00.000Z", 13 "clock_out": null, 14 "is_gate": true, 15 "last_distance": 200, 16 "gps_lat": "-4.200000", 17 "gps_lng": "100.010000", 18 "note": "Datang tepat waktu", 19 "face_confidence": 90, 20 "is_face_verified": true, 21 "attendance_setting_id": 1, 22 "created_by": 1, 23 "updated_at": "2020-05-07T10:02:53.000Z", 24 "deleted_at": null, 25 "user": { 26 "id": 1, 27 "name": "Johnny Raza", 28 "email": "johnny@examle.com", 29 "username": "johnny_raza", 30 "phone_no": "0821546789", 31 "role": 1, 32 "is_active": true, 33 "password": "12345678", 34 "created_at": "2020-05-07T10:02:53.000Z", 35 "updated_at": "2020-05-07T10:02:53.000Z", 36 "deleted_at": null 37 } 38 } 39 } </pre>
		/api/v1/permit/{id}	DEL	<pre> 1 { 2 "statusCode": 200, 3 "message": "Successfully delete permit", 4 "data": {} 5 } </pre>
7	Notification	/api/v1/notification	GET	<pre> 1 { 2 "message": "Successfully get notification", 3 "data": { 4 "id": 1, 5 "type": 1, 6 "status": "Approved", 7 "perm_name": "Permit", 8 "id": 1, 9 "created_at": "2020-05-07T10:02:53.000Z", 10 "updated_at": "2020-05-07T10:02:53.000Z", 11 "deleted_at": null, 12 "type": 1, 13 "clock_in": "2020-05-07T09:00:00.000Z", 14 "clock_out": null, 15 "is_gate": true, 16 "last_distance": 200, 17 "gps_lat": "-4.200000", 18 "gps_lng": "100.010000", 19 "note": "Datang tepat waktu", 20 "face_confidence": 90, 21 "is_face_verified": true, 22 "attendance_setting_id": 1, 23 "created_by": 1, 24 "updated_at": "2020-05-07T10:02:53.000Z", 25 "deleted_at": null, 26 "user": { 27 "id": 1, 28 "name": "Johnny Raza", 29 "email": "johnny@examle.com", 30 "username": "johnny_raza", 31 "phone_no": "0821546789", 32 "role": 1, 33 "is_active": true, 34 "password": "12345678", 35 "created_at": "2020-05-07T10:02:53.000Z", 36 "updated_at": "2020-05-07T10:02:53.000Z", 37 "deleted_at": null 38 } 39 } 40 } </pre>
8	Attendance	/api/v1/attendances	GET	<pre> 1 { 2 "statusCode": 200, 3 "message": "Successfully get attendances", 4 "data": { 5 "user": { 6 "id": 1, 7 "name": "Johnny Raza", 8 "email": "johnny@examle.com", 9 "username": "johnny_raza", 10 "phone_no": "0821546789", 11 "role": 1, 12 "is_active": true, 13 "password": "12345678", 14 "created_at": "2020-05-07T10:02:53.000Z", 15 "updated_at": "2020-05-07T10:02:53.000Z", 16 "deleted_at": null, 17 "user": { 18 "id": 1, 19 "name": "Johnny Raza", 20 "email": "johnny@examle.com", 21 "username": "johnny_raza", 22 "phone_no": "0821546789", 23 "role": 1, 24 "is_active": true, 25 "password": "12345678", 26 "created_at": "2020-05-07T10:02:53.000Z", 27 "updated_at": "2020-05-07T10:02:53.000Z", 28 "deleted_at": null 29 } 30 } 31 } 32 } </pre>
		/api/v1/attendances/{id}	GET	<pre> 1 { 2 "statusCode": 200, 3 "message": "Successfully get attendance", 4 "data": { 5 "id": 1, 6 "user_id": 1, 7 "clock_in": "2020-05-07T09:02:53.000Z", 8 "clock_out": null, 9 "is_gate": true, 10 "last_distance": 200, 11 "gps_lat": "-4.200000", 12 "gps_lng": "100.010000", 13 "note": "Datang tepat waktu", 14 "face_confidence": 90, 15 "is_face_verified": true, 16 "attendance_setting_id": 1, 17 "created_at": "2020-05-07T10:02:53.000Z", 18 "updated_at": "2020-05-07T10:02:53.000Z", 19 "deleted_at": null, 20 "user": { 21 "id": 1, 22 "name": "Johnny Raza", 23 "email": "johnny@examle.com", 24 "username": "johnny_raza", 25 "phone_no": "0821546789", 26 "role": 1, 27 "is_active": true, 28 "password": "12345678", 29 "created_at": "2020-05-07T10:02:53.000Z", 30 "updated_at": "2020-05-07T10:02:53.000Z", 31 "deleted_at": null 32 } 33 } 34 } </pre>
		/api/v1/attendances/today	GET	<pre> 1 { 2 "statusCode": 200, 3 "message": "Successfully get today attendance", 4 "data": { 5 "has_checked_in": true, 6 "has_checked_out": false, 7 "attendance": { 8 "id": 1, 9 "user_id": 1, 10 "clock_in": "2020-05-07T09:02:53.000Z", 11 "clock_out": null, 12 "is_gate": true, 13 "last_distance": 200, 14 "gps_lat": "-4.200000", 15 "gps_lng": "100.010000", 16 "note": "Datang tepat waktu", 17 "face_confidence": 90, 18 "is_face_verified": true, 19 "attendance_setting_id": 1, 20 "created_at": "2020-05-07T10:02:53.000Z", 21 "updated_at": "2020-05-07T10:02:53.000Z", 22 "deleted_at": null, 23 "attendance_setting": { 24 "id": 1, 25 "gps_lat": "-4.200000", 26 "gps_lng": "100.010000", 27 "radius_meter": 50, 28 "check_in_time": "06:00:00", 29 "check_out_time": "17:00:00", 30 "updated_at": "2020-05-07T10:02:53.000Z", 31 "deleted_at": null 32 } 33 } 34 } 35 } </pre>

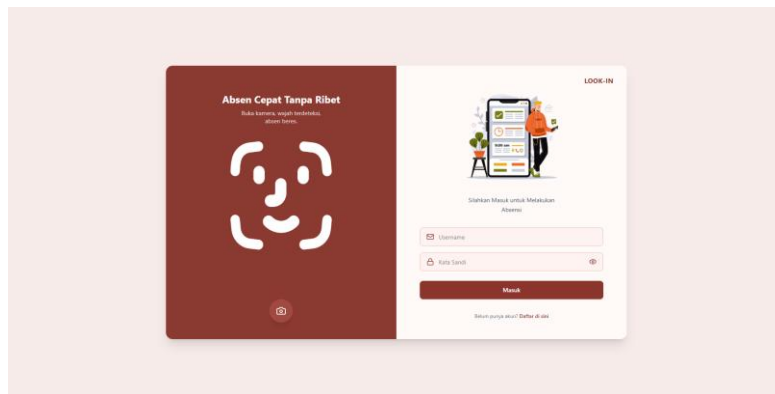
		/api/v1/attendances/history?start_date=2026-05-01&end_date=2026-05-31	GET	
		/api/v1/attendances/check-in?gps_lat=-6.200000&gps_lng=106.816666&face_confidence=95	POST	
		/api/v1/attendances/check-out?gps_lat=-6.200000&gps_lng=106.816666	POST	
		/api/v1/attendances/{id}/late-note	PATCH	
		/api/v1/attendances/{id}	DEL	
9	Attendance Image	/api/v1/attendances/{id}/images	GET	
		/api/v1/attendances/{id}/images	POST	

Implementasi API pada sistem LOOK-IN digunakan sebagai penghubung antara *frontend*, *backend*, *database*, serta modul *face recognition* agar proses pertukaran data dapat berjalan secara *real-time*. *Backend* sistem dikembangkan menggunakan *NestJS* dan AI menggunakan *FastAPI* dengan pendekatan REST API. Setiap respons API menggunakan format JSON yang memuat informasi seperti *message*.

4.1.3 Implementasi Pada Sisi UI

Implementasi antarmuka pengguna (*User Interface/UI*) pada sistem LOOK-IN dilakukan untuk menyediakan tampilan yang interaktif, responsif, dan mudah digunakan oleh pengguna. Pengembangan tampilan sistem menggunakan *framework Vue.js* dengan pendekatan *Progressive Web App* sehingga aplikasi dapat diakses melalui *browser* pada berbagai perangkat baik *desktop* maupun *mobile*. Implementasi UI dirancang berdasarkan kebutuhan fungsional sistem yang telah dianalisis sebelumnya agar setiap fitur dapat digunakan dengan mudah oleh karyawan maupun HR.

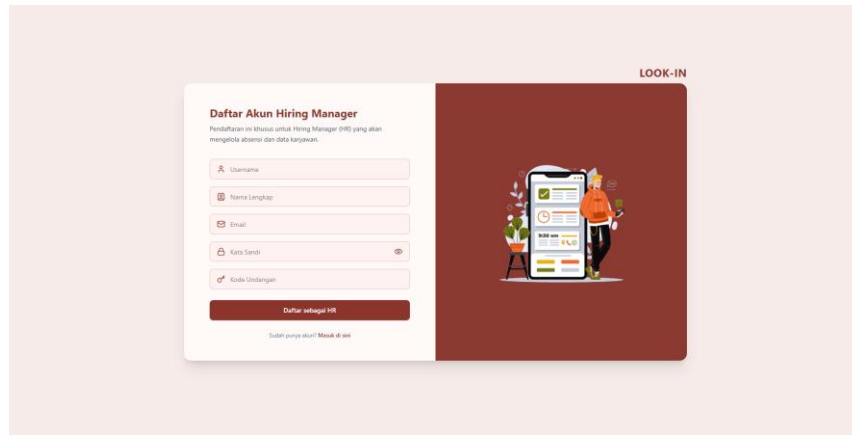
1. Halaman *Login*



Gambar 4.2 Implementasi Halaman *Login*

Gambar 4.2 merupakan halaman login aplikasi LOOK-IN yang dirancang untuk memfasilitasi akses pengguna melalui pengisian *username* dan kata sandi. Dalam sistem ini, terdapat perbedaan prosedur akses dimana pihak HR dapat langsung masuk setelah melakukan registrasi mandiri, sementara akun untuk karyawan harus didaftarkan terlebih dahulu oleh HR sebelum bisa mengakses layanan absensi tersebut.

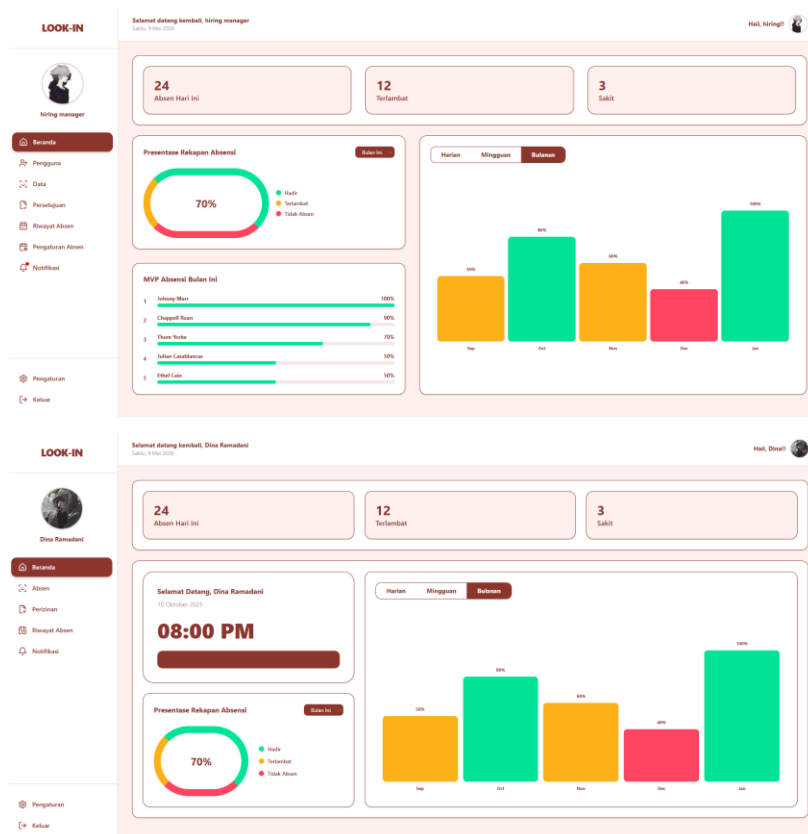
2. Halaman *Register*



Gambar 4.3 Implementasi Halaman *Register*

Gambar 4.3 merupakan halaman pendaftaran akun HR pada aplikasi LOOK-IN. Pengguna perlu mengisi *username*, nama lengkap, *email*, dan kata sandi. Jika akun HR telah tersedia di dalam sistem, pengguna juga harus memasukkan kode undangan yang valid. Setelah data diisi dengan benar, pengguna dapat menekan tombol "Daftar sebagai HR" untuk membuat akun dan memperoleh akses ke sistem.

3. Halaman *Dashboard*



Gambar 4.4 Implementasi Halaman *Dashboard*

Gambar 4.4 merupakan halaman *dashboard* karyawan dirancang untuk memberikan ringkasan kehadiran personal karyawan yang mencakup jam digital, status kehadiran harian, serta grafik persentase rekapan absensi bulanan guna memantau kedisiplinan mandiri. Sementara itu, dashboard HR berfungsi sebagai pusat pemantauan manajerial yang menampilkan statistik akumulatif seluruh staf.

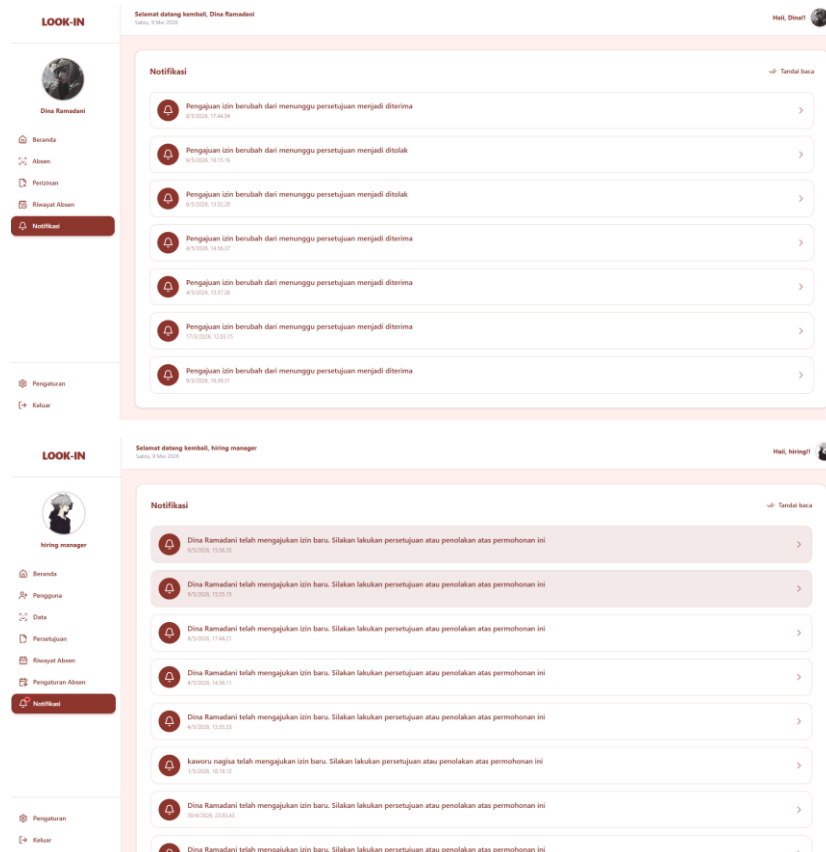
4. Halaman Pengaturan Akun

The image displays two screenshots of the 'LOOK-IN' application's account settings page. The top screenshot is for the user 'hiring manager' (Email: hiringmanager@gmail.com, Phone: 081277738731). The bottom screenshot is for the user 'Dina Ramadani' (Email: dina@gmail.com). Both pages feature a sidebar with navigation options: Beranda, Pengguna, Data, Penawaran, Riwayat Absen, Pengaturan Absen, and Notifikasi. The main content area includes a profile section with a photo, name, and email, followed by input fields for Email, Nama Lengkap, and Nomor Telepon. A 'Perbarui Profil' button is located below these fields. To the right, there is a password management section with fields for 'Kata Sandi Lama', 'Kata Sandi Baru', and 'Konfirmasi Kata Sandi', each with a 'Masukkan' or 'Konfirmasi' button. A 'Perbarui' button is at the bottom right. A list of password requirements is provided: minimum 8 characters, including uppercase, lowercase, and numbers; no spaces; and no repeated characters.

Gambar 4.5 Implementasi Halaman Pengaturan Akun

Gambar 4.5 merupakan halaman pengaturan akun bagi karyawan dan HR, halaman ini memungkinkan pengguna untuk memperbarui informasi profil mereka, mulai dari mengganti foto profil, mengubah nama lengkap, email, hingga nomor telepon. Selain data diri, halaman ini juga menyediakan fitur pembaruan kata sandi.

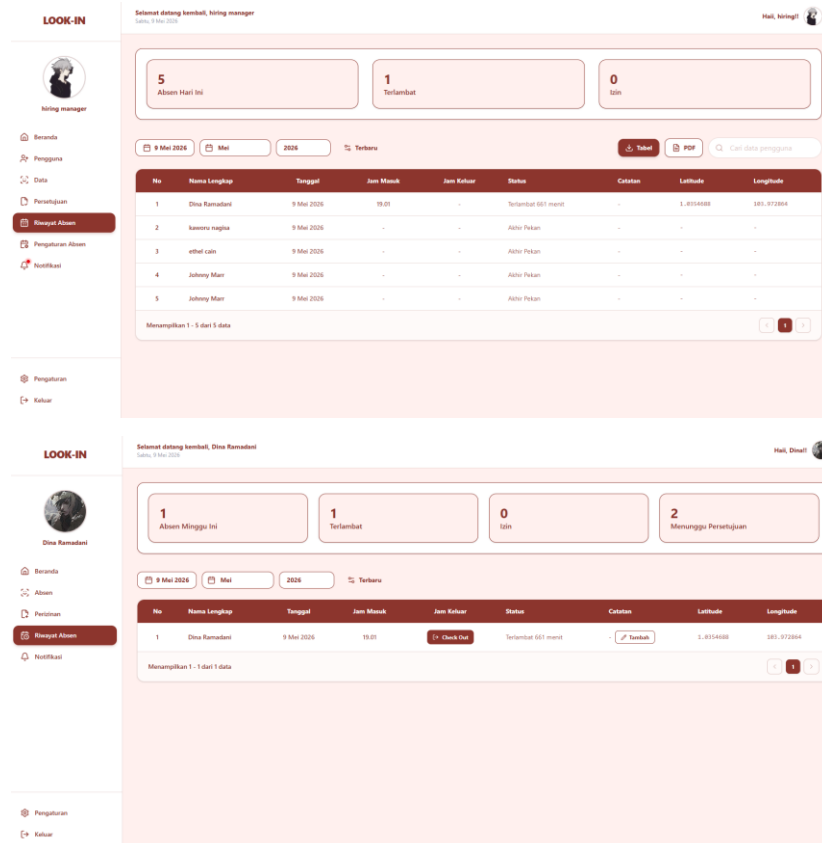
5. Halaman Notifikasi



Gambar 4.6 Implementasi Halaman Notifikasi

Gambar 4.6 merupakan halaman notifikasi bagi karyawan dan HR, berfungsi sebagai pusat informasi *real-time* mengenai status administrasi kepegawaian. Bagi karyawan, halaman ini menampilkan pembaruan status pengajuan izin apakah diterima atau ditolak oleh atasan sementara bagi HR, halaman ini memberikan pemberitahuan instan setiap kali ada karyawan yang mengajukan izin baru agar dapat segera ditindaklanjuti melalui proses persetujuan atau penolakan.

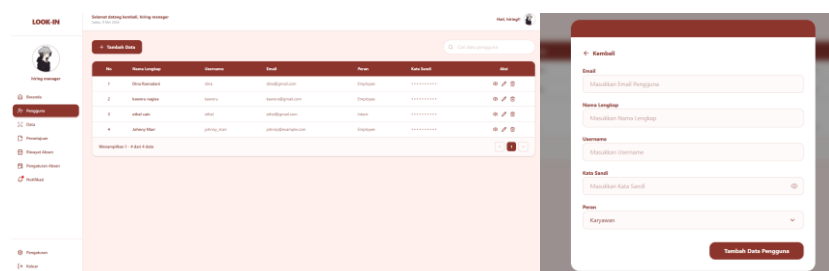
6. Halaman Riwayat Absensi



Gambar 4.7 Implementasi Halaman Riwayat Absensi

Gambar 4.7 merupakan riwayat absensi bagi karyawan dan HR, halaman ini mencatat detail kehadiran mulai dari jam masuk, jam keluar, hingga lokasi GPS saat melakukan absen. Bagi HR, fitur ini memudahkan pemantauan seluruh karyawan dan pengunduhan laporan dalam format PDF, sementara bagi pekerja, halaman ini digunakan untuk memantau kehadiran pribadi serta melakukan proses *check-out* kerja.

7. Halaman Mengelola Pengguna

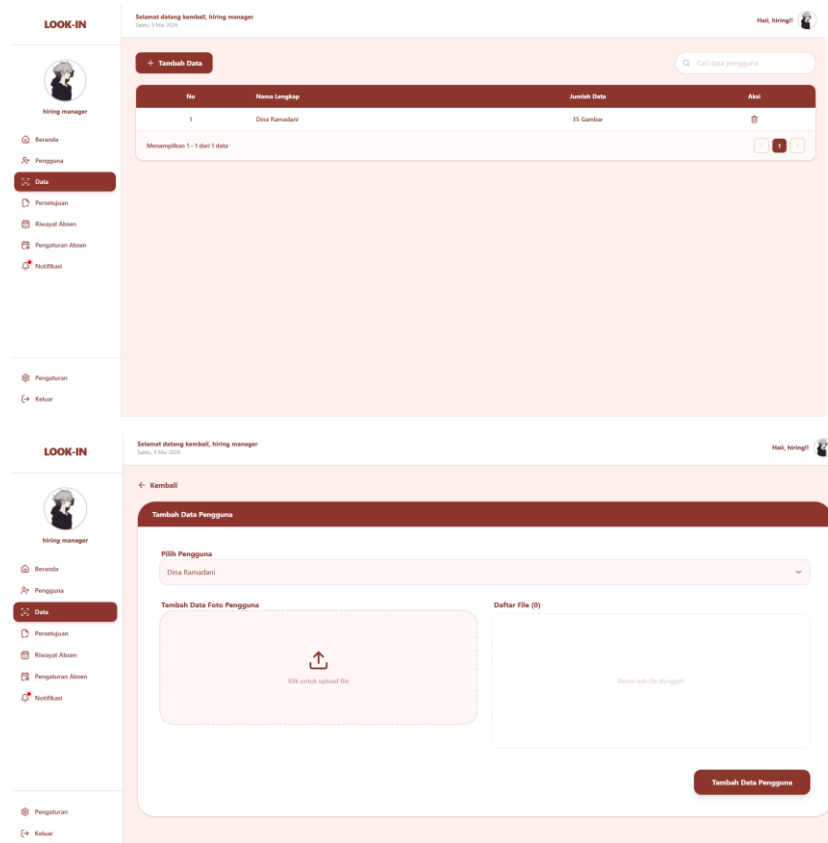


Gambar 4.8 Implementasi Halaman Mengelola Pengguna

Gambar 4.8 merupakan halaman kelola pengguna bagi HR, digunakan oleh HR untuk mengelola data karyawan, mulai dari

melihat daftar nama, *username*, hingga peran masing-masing. Melalui tombol "Tambah Data", HR bisa mendaftarkan akun baru bagi karyawan, serta memiliki kendali penuh untuk melihat detail, mengubah informasi, atau menghapus data pengguna yang ada melalui kolom aksi.

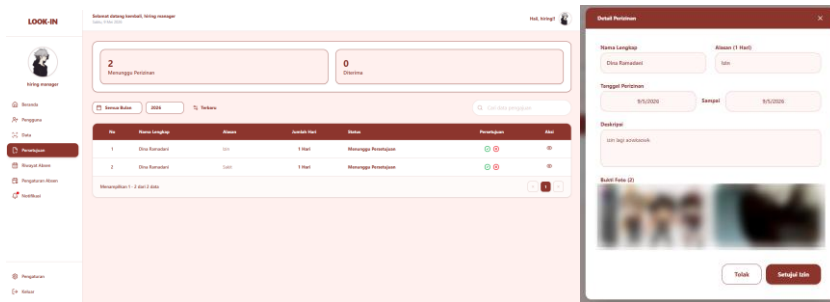
8. Halaman Mengelola *Dataset*



Gambar 4.9 Implementasi Halaman Mengelola *Dataset*

Gambar 4.9 merupakan halaman mengelola *dataset*, halaman ini tempat untuk mengolah *dataset* foto untuk kebutuhan fitur pengenalan wajah. Melalui formulir "Tambah Data Pengguna", HR dapat memilih nama karyawan dan mengunggah sejumlah foto mereka yang nantinya akan diproses melalui tahap *training* dataset AI agar sistem dapat mengenali wajah pekerja secara akurat saat melakukan absensi.

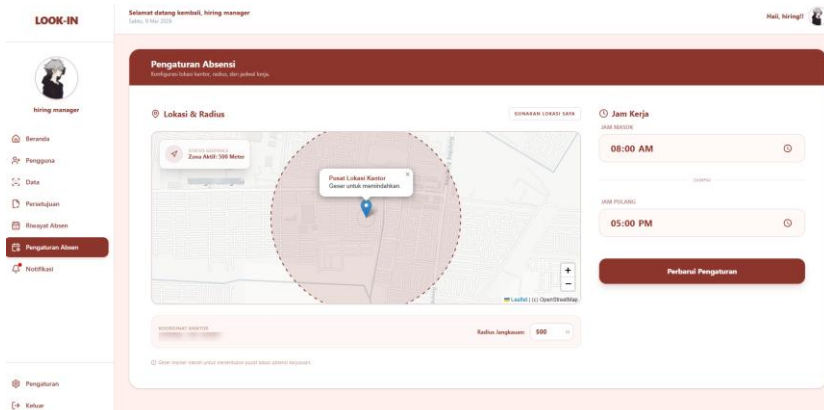
9. Halaman Menangani Persetujuan



Gambar 4.10 Implementasi Menangani Persetujuan

Gambar 4.10 merupakan halaman menangani persetujuan bagi HR, halaman ini berfungsi untuk membantu HR memproses pengajuan izin atau sakit dari karyawan dengan lebih terorganisir. Melalui tampilan ini, HR dapat melihat ringkasan jumlah permohonan yang masuk, memeriksa detail alasan serta bukti foto yang dilampirkan, lalu memberikan keputusan secara cepat dengan menekan tombol "Setujui Izin" atau "Tolak".

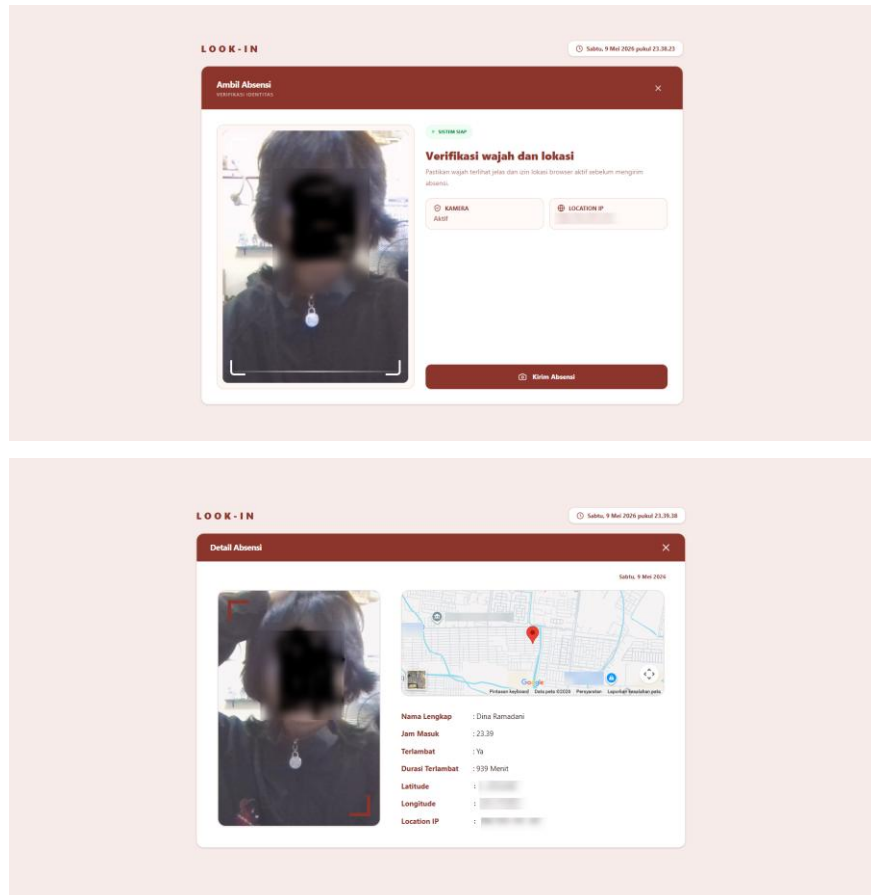
10. Halaman Mengelola Peraturan Absensi



Gambar 4.11 Implementasi Mengelola Peraturan Absensi

Gambar 4.11 merupakan halaman mengelola peraturan absensi bagi HR, halaman memungkinkan HR untuk menentukan lokasi kantor dan radius absensi melalui peta interaktif agar karyawan hanya bisa melakukan absen dalam zona yang sudah ditentukan. Selain mengatur area geografis, halaman ini juga digunakan untuk menetapkan jam kerja resmi, seperti jam masuk dan jam pulang, untuk memastikan pencatatan kehadiran berjalan sesuai jadwal perusahaan.

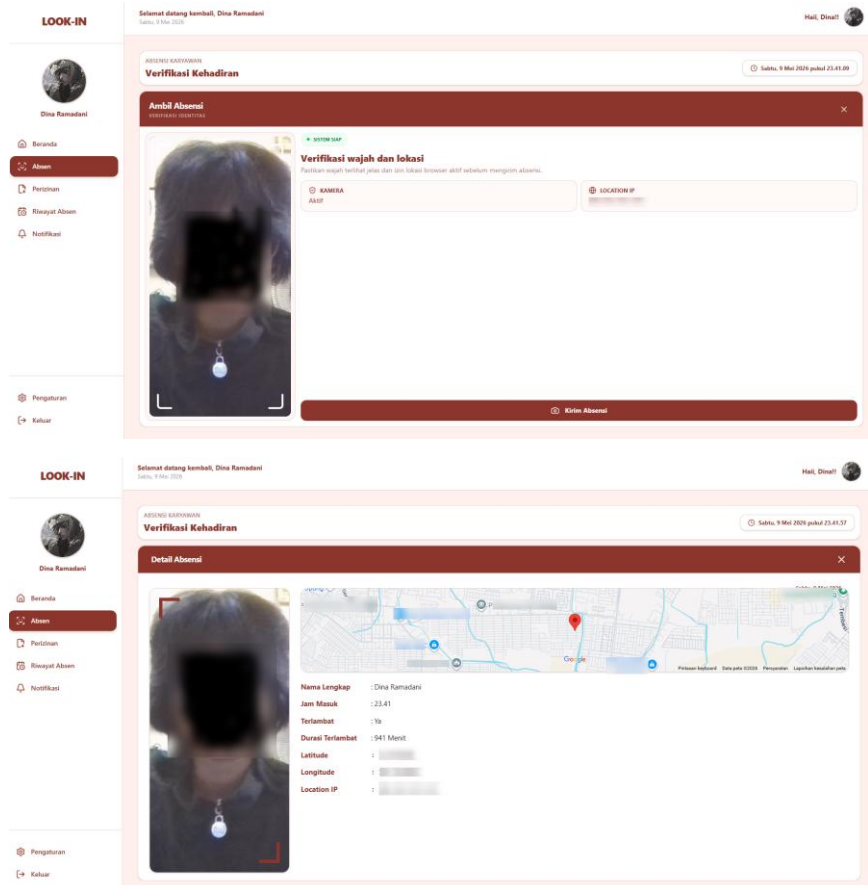
11. Halaman Mengambil Absensi Cepat



Gambar 4.12 Implementasi Mengambil Absensi Cepat

Gambar 4.12 merupakan halaman mengambil absensi cepat bagi karyawan, halaman ini memungkinkan karyawan untuk melakukan absen secara praktis tanpa harus melakukan *login* ke akun terlebih dahulu. Sistem ini bekerja dengan cara memverifikasi wajah melalui kamera dan mendeteksi lokasi perangkat secara otomatis untuk memastikan karyawan berada di area kantor yang sesuai. Setelah wajah terdeteksi dan lokasi tervalidasi, sistem akan menampilkan detail kehadiran seperti nama lengkap, jam masuk, serta titik koordinat GPS sebagai bukti absensi.

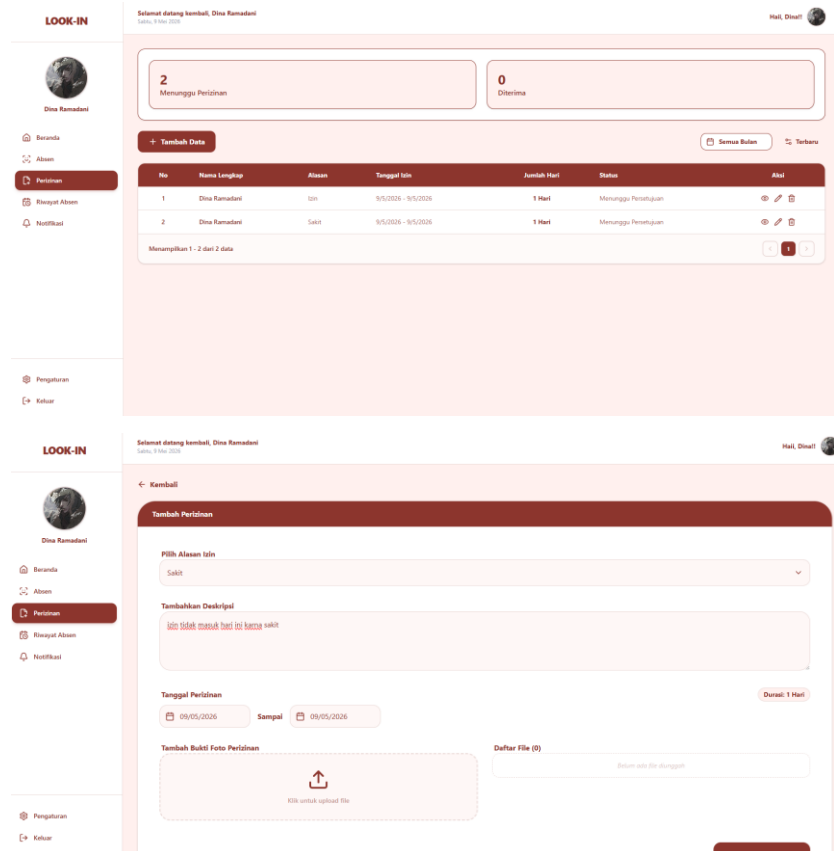
12. Halaman Mengambil Absensi



Gambar 4.13 Implementasi Mengambil Absensi

Gambar 4.13 merupakan halaman mengambil absensi bagi karyawan, halaman ini digunakan untuk melakukan absensi namun pada halaman karyawan harus *login* terlebih dahulu, karyawan dapat melakukan absensi dengan cara memindai wajah dan memvalidasi lokasi secara otomatis sebelum mengirim data kehadiran. Setelah proses verifikasi selesai, sistem akan menampilkan rincian absensi yang mencakup informasi waktu masuk, peta lokasi, serta koordinat GPS untuk memastikan akurasi data kehadiran karyawan tersebut.

13. Halaman Mengajukan Perizinan



Gambar 4.14 Mengajukan Perizinan

Gambar 4.14 merupakan halaman mengambil absensi cepat bagi karyawan, halaman ini memungkinkan karyawan untuk mengajukan izin atau sakit dengan mudah secara mandiri. Melalui formulir "Tambah Perizinan", pekerja dapat memilih alasan izin, menuliskan deskripsi singkat, menentukan rentang tanggal, serta mengunggah bukti foto pendukung agar pengajuan tersebut dapat segera diproses dan dipantau status persetujuannya melalui tabel riwayat perizinan.

4.1.4 Implementasi AI

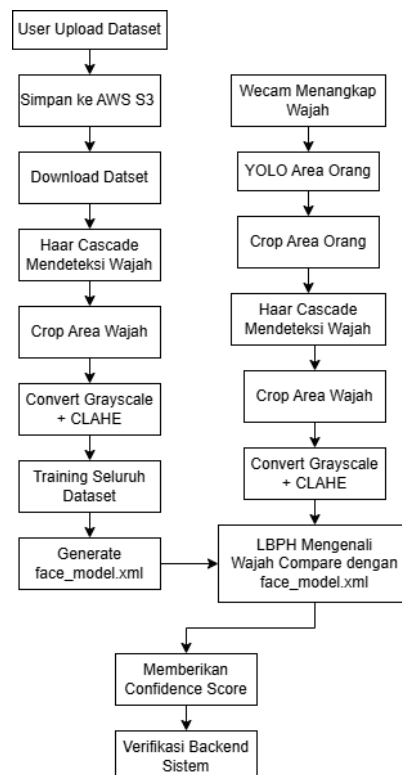
Implementasi AI pada sistem LOOK-IN digunakan untuk mendukung fitur face recognition dalam proses verifikasi kehadiran pengguna. Sistem ini dibangun dengan arsitektur terpisah, di mana backend menggunakan *NestJS* dan layanan AI menggunakan *FastAPI* yang diakses melalui *REST API*.

Pertukaran data antara *frontend*, *backend*, *database*, dan modul AI dilakukan secara *real-time* menggunakan format JSON. Proses yang terjadi

meliputi pengiriman data gambar dari *frontend* ke backend, kemudian diteruskan ke layanan AI untuk dilakukan proses deteksi dan pengenalan wajah. Hasil dari proses tersebut dikembalikan ke backend dalam bentuk respons yang berisi status dan data hasil verifikasi.

1. Arsitektur AI

Sistem AI pada LOOK-IN menggunakan kombinasi model YOLO untuk deteksi objek (*person detection*) dan LBPH *Face Recognizer* untuk proses pengenalan wajah. YOLO digunakan untuk mendeteksi keberadaan manusia dalam *frame*, kemudian area manusia yang terdeteksi diproses kembali menggunakan *Haar Cascade Classifier* untuk mendeteksi wajah sebelum dilakukan pengenalan identitas menggunakan metode *LBPH Face Recognizer*



Gambar 4.15 Arsitektur AI

Berdasarkan diagram pada Gambar 4.15 proses *face recognition* pada sistem dimulai dari tahap *webcam capture* yang berfungsi untuk mengambil gambar wajah pengguna secara *real-time* melalui kamera. Setelah gambar diperoleh, sistem menggunakan *algoritma* YOLO untuk mendeteksi objek manusia

pada *frame* kamera. Setelah area manusia berhasil dideteksi, sistem menggunakan *Haar Cascade Classifier* untuk mendeteksi posisi wajah sebelum dilakukan proses pengenalan wajah.

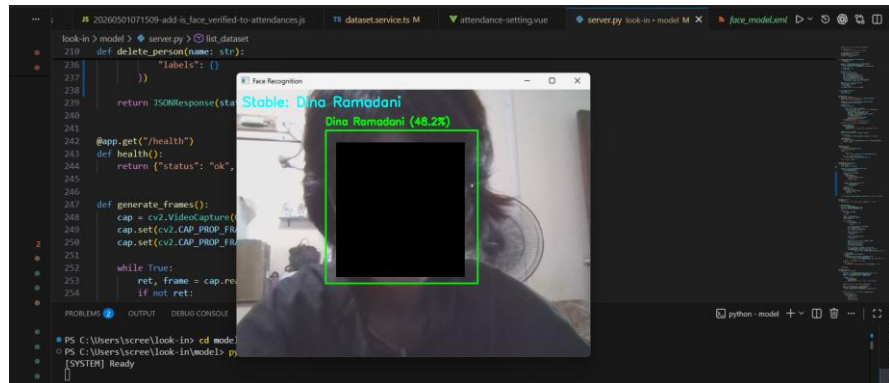
Selanjutnya, sistem melakukan proses *crop* pada area manusia yang terdeteksi. Setelah itu, *Haar Cascade* digunakan untuk mendeteksi area wajah sehingga proses pengenalan dapat difokuskan hanya pada bagian wajah pengguna. Hasil *crop* kemudian diubah menjadi image grayscale untuk menyederhanakan informasi citra dan mengurangi kompleksitas pemrosesan.

Setelah itu, citra wajah diproses menggunakan metode *Contrast Limited Adaptive Histogram Equalization* (CLAHE) untuk meningkatkan kualitas pencahayaan dan kontras gambar sehingga wajah lebih mudah dikenali meskipun berada pada kondisi pencahayaan yang kurang ideal.

Tahap berikutnya adalah proses pengenalan wajah menggunakan metode LBPH (*Local Binary Pattern Histogram*). Metode ini melakukan ekstraksi fitur wajah dan membandingkannya dengan data wajah yang telah tersimpan pada database sistem. Apabila hasil pencocokan memenuhi nilai threshold yang ditentukan, maka sistem akan melakukan verifikasi identitas pengguna dan mengirimkan hasilnya ke sistem *backend*.

2. Implementasi Deteksi Wajah

Implementasi deteksi wajah pada sistem LOOK-IN dilakukan secara real-time menggunakan webcam sebagai media input. Sistem memanfaatkan model YOLO untuk mendeteksi keberadaan objek manusia pada frame kamera. Selanjutnya, *Haar Cascade Classifier* digunakan untuk mendeteksi area wajah sebelum diproses menggunakan metode LBPH *Face Recognizer* untuk melakukan pengenalan identitas pengguna. Untuk melakukan pengenalan identitas pengguna. Hasil deteksi dan pengenalan ditampilkan dalam bentuk *bounding box*, nama pengguna, serta *confidence score* pada tampilan kamera secara langsung.



Gambar 4.16 Implementasi deteksi dan pengenalan wajah secara *real-time* menggunakan metode YOLO dan LBPH

Berdasarkan Gambar 4.16, sistem berhasil mendeteksi wajah pengguna secara *real-time* melalui *webcam*. Area wajah yang terdeteksi ditandai menggunakan *bounding box* berwarna hijau, kemudian sistem melakukan proses pengenalan identitas menggunakan metode LBPH. Hasil pengenalan ditampilkan dalam bentuk nama pengguna dan *confidence score* pada *frame* kamera.

3. Implementasi *Preprocessing* Wajah

Sebelum dilakukan proses pengenalan wajah menggunakan metode LBPH, sistem terlebih dahulu melakukan *preprocessing* citra wajah untuk meningkatkan kualitas gambar dan mempermudah proses ekstraksi fitur. Tahapan *preprocessing* meliputi proses *crop* wajah, konversi *grayscale*, dan peningkatan kontras menggunakan metode CLAHE.



Gambar 4.17 Tahapan *preprocessing* wajah pada sistem *face recognition*

Berdasarkan Gambar 4.17, sistem melakukan proses *crop* pada area wajah yang terdeteksi untuk memfokuskan pemrosesan hanya pada bagian wajah pengguna. Selanjutnya citra diubah menjadi *grayscale* untuk mengurangi kompleksitas data gambar.

Setelah itu, metode CLAHE digunakan untuk meningkatkan kontras dan kualitas pencahayaan citra sehingga proses pengenalan wajah dapat berjalan lebih optimal.

4. Implementasi *Face Recognition* LBPH

Setelah proses *preprocessing* selesai dilakukan, sistem menggunakan metode LBPH untuk melakukan pengenalan wajah pengguna. Metode ini bekerja dengan melakukan ekstraksi fitur tekstur wajah dan membandingkannya dengan dataset wajah yang telah tersimpan pada sistem.

Berikut merupakan proses pengumpulan dan pelatihan data wajah yang digunakan dalam pembangunan model *face recognition* pada sistem LOOK-IN, mulai dari proses pengunduhan dataset hingga proses training model LBPH.

```
PS C:\Users\screel\look-in\model> uvicorn server:app --host 0.0.0.0 --port 8000 --reload
[AI] Saved to: database\Dina Ramadani\28.webp
[AI] Download status for https://look-in-app.s3.ap-southeast-2.amazonaws.com/dataset/images-20260511-23e90c9e-bb66b399684eda1092b54c97b4c1dcec.webp: 200
[AI] Saved to: database\Dina Ramadani\29.webp
[AI] Download status for https://look-in-app.s3.ap-southeast-2.amazonaws.com/dataset/images-20260511-48ded542-d9cd21392e16e69acc12a35cce224cab.webp: 200
[AI] Saved to: database\Dina Ramadani\30.webp
[AI] Download status for https://look-in-app.s3.ap-southeast-2.amazonaws.com/dataset/images-20260511-0c12982c-167275e928e70cffff142226bc2b78ac.webp: 200
[AI] Saved to: database\Dina Ramadani\31.webp
[AI] Download status for https://look-in-app.s3.ap-southeast-2.amazonaws.com/dataset/images-20260511-63edb82c-e0f0430ac83bab9274418b18eaa33e8c.webp: 200
[AI] Saved to: database\Dina Ramadani\32.webp
[AI] Download status for https://look-in-app.s3.ap-southeast-2.amazonaws.com/dataset/images-20260511-0c141b28-6c58103f60f844bfcae194656fd9902e.webp: 200
[AI] Saved to: database\Dina Ramadani\33.webp
[AI] Download status for https://look-in-app.s3.ap-southeast-2.amazonaws.com/dataset/images-20260511-9a0adade-8b2cd1c8fa1d26ba974222bc6669e1a2.webp: 200
```

Gambar 4.18 Proses Pengunduhan Dataset

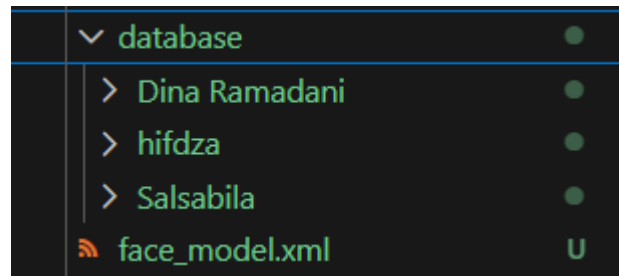
Proses training dilakukan melalui *endpoint /train* dengan menerima data gambar dari *cloud storage* (S3). Gambar yang diterima kemudian disimpan ke dalam direktori berdasarkan nama pengguna sebagai label identitas wajah.

```
PS C:\Users\screel\look-in\model> uvicorn server:app --host 0.0.0.0 --port 8000 --reload
[DB] Processing: database\Dina Ramadani\4.webp
[DB] Faces detected in 4.webp: 1
[DB] Processing: database\Dina Ramadani\5.webp
[DB] Faces detected in 5.webp: 1
[DB] Processing: database\Dina Ramadani\6.webp
[DB] Faces detected in 6.webp: 1
[DB] Processing: database\Dina Ramadani\7.webp
[DB] Faces detected in 7.webp: 1
[DB] Processing: database\Dina Ramadani\8.webp
[DB] Faces detected in 8.webp: 1
[DB] Processing: database\Dina Ramadani\9.webp
[DB] Faces detected in 9.webp: 0
[DB] Total faces collected: 35
[AI] build_db result: True, {'Dina Ramadani': 0}
[AI] Recognizer loaded
INFO: 127.0.0.1:52680 - "POST /train HTTP/1.1" 200 OK
```

Gambar 4.19 Proses Training Dataset

Setelah gambar berhasil diunduh, fungsi *build_db()* akan membaca seluruh *file* citra pada *dataset*. Setiap gambar dikonversi

menjadi *grayscale* dan diproses menggunakan *Haar Cascade Classifier* untuk mendeteksi area wajah. Jika wajah berhasil terdeteksi, sistem akan melakukan *crop wajah*, *resize* menjadi 200×200 piksel, dan *preprocessing* menggunakan metode *CLAHE* untuk meningkatkan kualitas citra sebelum dilakukan proses training model LBPH.



Gambar 4.20 Struktur Folder *Dataset Wajah*

Struktur folder *database* digunakan untuk menyimpan *dataset* wajah pengguna yang akan digunakan dalam proses *training* model *face recognition*. Setiap *folder* merepresentasikan identitas pengguna, sedangkan hasil pelatihan model disimpan dalam file *face_model.xml* dan *labels.npy* yang digunakan pada proses pengenalan wajah secara *real-time*.

Setelah model berhasil dilatih, sistem melakukan proses pengenalan wajah menggunakan fungsi *predict()* pada metode LBPH *Face Recognizer*. Sistem akan membandingkan wajah pengguna dengan dataset yang telah tersimpan dan menghasilkan nilai *confidence* untuk menentukan tingkat kecocokan wajah.

```
[AI] Verify - frame shape: (409, 374, 3)
[AI] Verify - faces detected: 1
[AI] Verify - name: Dina Ramadani, conf: 58.43608507442049
INFO: 127.0.0.1:59720 - "POST /verify HTTP/1.1" 200 OK
```

Gambar 4.21 Hasil Pengenalan Wajah Menggunakan LBPH

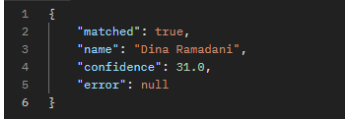
Berdasarkan Gambar 4.22, sistem berhasil melakukan proses pengenalan wajah menggunakan metode LBPH *Face Recognizer*. Sistem membandingkan citra wajah pengguna dengan *dataset* yang telah dilatih sebelumnya dan menghasilkan nilai *confidence* sebagai indikator tingkat kecocokan wajah. Semakin kecil nilai *confidence*

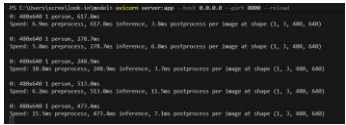
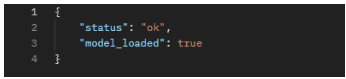
yang dihasilkan, maka tingkat kemiripan wajah dengan dataset semakin tinggi. Pada implementasi sistem LOOK-IN, wajah dianggap cocok apabila nilai confidence berada di bawah threshold yang telah ditentukan.

5. Implementasi API AI Service

pada sistem LOOK-IN. API digunakan untuk menghubungkan sistem utama dengan modul *Face Recognition* sehingga proses training, verifikasi wajah, dan streaming kamera dapat dilakukan secara terpisah dan *real-time*.

Tabel 4.2 Implementasi API AI

Endpoint	Deskripsi	Screenshot
/train	<i>Endpoint /train</i> digunakan untuk melakukan training <i>dataset</i> wajah. Sistem menerima nama pengguna dan daftar <i>URL</i> gambar dari <i>cloud storage</i> (S3), kemudian mengunduh gambar tersebut dan menyimpannya ke dalam <i>folder dataset</i> sesuai nama pengguna. Setelah proses pengunduhan selesai, sistem menjalankan fungsi <i>build_db()</i> untuk melakukan <i>retraining model</i> LBPH <i>Face Recognizer</i> .	
/verify	<i>Endpoint /verify</i> digunakan untuk melakukan verifikasi wajah berdasarkan gambar yang dikirim dalam format <i>Base64</i> . Sistem akan mendeteksi wajah pada gambar, melakukan <i>preprocessing</i> , kemudian	

	mencocokkan wajah dengan model LBPH yang telah dilatih sebelumnya.	
/stream	<i>Endpoint /stream</i> digunakan untuk menampilkan proses <i>Face Detection</i> dan <i>Recognition</i> secara <i>real-time</i> melalui <i>streaming</i> kamera. Setiap <i>frame</i> diproses menggunakan YOLO untuk mendeteksi manusia, kemudian dilakukan pengenalan wajah menggunakan LBPH <i>Face Recognizer</i> .	 <pre> 1 C:\Users\user\Downloads\server>python server.py --port 8080 --host 0.0.0.0 2 3 4/4/2024 1 person, 402.0ms 4 total: 4.0ms preprocess, 402.0ms inference, 3.0ms postprocess per image at shape (1, 1, 480, 640) 5 6 4/4/2024 1 person, 278.0ms 7 total: 3.0ms preprocess, 278.0ms inference, 6.0ms postprocess per image at shape (1, 1, 480, 640) 8 9 4/4/2024 1 person, 388.0ms 10 total: 18.0ms preprocess, 388.0ms inference, 1.7ms postprocess per image at shape (1, 1, 480, 640) 11 12 4/4/2024 1 person, 511.0ms 13 total: 3.0ms preprocess, 511.0ms inference, 11.0ms postprocess per image at shape (1, 1, 480, 640) 14 15 4/4/2024 1 person, 473.0ms 16 total: 17.0ms preprocess, 473.0ms inference, 7.0ms postprocess per image at shape (1, 1, 480, 640) </pre>
/dataset	<i>Endpoint /dataset</i> digunakan untuk menampilkan daftar pengguna yang tersimpan dalam <i>database</i> beserta jumlah foto <i>dataset</i> yang dimiliki masing-masing pengguna.	 <pre> 1 { 2 "people": [3 { 4 "name": "Dina Ramadani", 5 "photos": 35 6 }, 7 { 8 "name": "hidra", 9 "photos": 10 10 }, 11 { 12 "name": "Salsabila", 13 "photos": 20 14 } 15] 16 } </pre>
/health	<i>Endpoint /health</i> digunakan untuk memeriksa status layanan AI serta memastikan model <i>Face Recognition</i> berhasil dimuat ke dalam sistem.	 <pre> 1 { 2 "status": "ok", 3 "model_loaded": true 4 } </pre>

4.2 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan seluruh fitur pada aplikasi absensi berbasis *face recognition* dapat berjalan sesuai kebutuhan pengguna dan fungsi yang telah dirancang. Pengujian ini bertujuan untuk mengetahui apakah sistem mampu menjalankan proses autentikasi, absensi, pengelolaan pengguna, pengajuan izin, notifikasi, hingga pengaturan absensi secara stabil dan sesuai dengan kebutuhan operasional perusahaan.

Pada penelitian ini, pengujian sistem dilakukan menggunakan metode *User Acceptance Testing* (UAT). Pengujian dilakukan dengan cara menguji langsung setiap fitur utama sistem berdasarkan skenario penggunaan oleh pengguna akhir, yaitu HR dan karyawan. Setiap fitur diuji menggunakan beberapa skenario untuk memastikan sistem dapat menangani proses validasi data, penyimpanan data, pengolahan informasi, serta menampilkan hasil sesuai yang diharapkan

4.2.1 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) merupakan metode pengujian yang dilakukan untuk memastikan bahwa sistem telah memenuhi kebutuhan pengguna dan dapat digunakan sesuai proses bisnis yang berjalan. Pengujian ini berfokus pada validasi fungsi sistem berdasarkan interaksi langsung pengguna terhadap fitur-fitur yang tersedia.

Pada penelitian ini, UAT dilakukan oleh HR selaku pengguna utama sistem pada tanggal 29 Mei 2026 di Perusahaan Jasa Teknologi Informasi. Pengujian dilakukan terhadap seluruh fitur utama sistem absensi berbasis *face recognition*, meliputi *login*, *logout*, absensi cepat, absensi menggunakan GPS dan *webcam*, pengelolaan pengguna, pengelolaan *dataset* wajah, pengajuan izin, notifikasi, riwayat absensi, ekspor laporan, pengaturan absensi, serta fitur catatan keterlambatan. Pengujian dilakukan menggunakan berbagai skenario normal dan skenario validasi untuk memastikan setiap fitur berfungsi sesuai kebutuhan pengguna serta mampu menangani berbagai kondisi penggunaan yang mungkin terjadi.

Parameter yang diamati pada pengujian ini meliputi keberhasilan proses input data, validasi sistem, penyimpanan data, perubahan status, notifikasi sistem, serta kesesuaian output dengan hasil yang diharapkan pengguna. Hasil pengujian menunjukkan bahwa seluruh fitur utama sistem dapat berjalan dengan baik sesuai kebutuhan fungsional yang telah ditentukan.

1. UATFR-001

Pengujian UAT FR-001 dilakukan untuk memastikan fitur login pada aplikasi LOOK-IN berjalan dengan baik untuk HR dan karyawan, termasuk validasi akun, password, dan field input. Hasil

pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-001 berikut ini.

Tabel 4.3 *User Acceptance Testing* FR-001

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-001	Pengujian fungsionalitas masuk ke sistem <i>Login</i> untuk <i>role</i> HR dan Karyawan, termasuk validasi akun yang didaftarkan oleh HR.	<i>Login</i> berhasil sebagai HR	HR berhasil masuk ke sistem	Sesuai	✓	
		<i>Login</i> Berhasil sebagai Karyawan	Karyawan berhasil masuk ke sistem	Sesuai	✓	
		<i>Login</i> Gagal Karena Password Salah	Sistem menolak login	Sesuai	✓	
		<i>Login</i> Gagal Karena akun Belum Dibuat HR	Sistem menolak akun tidak terdaftar	Sesuai	✓	
		<i>Login</i> Dengan <i>Field</i> Kosong	Sistem menampilkan validasi input	Sesuai	✓	

Persentase test case yang lolos = $5/5 = 100\%$

Persentase test case yang gagal = $0/5 = 0\%$

Persentase test case yang telah diuji = $5/5 = 100\%$

Persentase test case yang belum diuji = $0/5 = 0\%$

Jumlah total test case dalam pengujian = 5

2. UAT FR-002

Pengujian UAT FR-002 dilakukan untuk memastikan fitur logout pada aplikasi LOOK-IN dapat mengakhiri sesi penggunaan sistem dengan aman untuk HR maupun karyawan. Hasil pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-002 berikut ini.

Tabel 4.4 *User Acceptance Testing* FR-002

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-002	Pengujian fungsionalitas keluar dari sistem <i>Logout</i> untuk memastikan pengguna dapat mengakhiri sesi penggunaan aplikasi dengan aman.	<i>Logout</i> berhasil sebagai HR	HR berhasil keluar dari sistem	Sesuai	✓	
		<i>Logout</i> berhasil sebagai Karyawan	HR berhasil keluar dari sistem	Sesuai	✓	

Persentase test case yang lolos = $2/2 = 100\%$

Persentase test case yang gagal = $0/2 = 0\%$

Persentase test case yang telah diuji = $2/2 = 100\%$

Persentase test case yang belum diuji = $0/2 = 0\%$

Jumlah total test case dalam pengujian = 2

3. UAT FR-003

Pengujian UAT FR-003 dilakukan untuk memastikan fitur dashboard laporan pada aplikasi LOOK-IN dapat menampilkan ringkasan data absensi, statistik kehadiran, grafik laporan, dan informasi absensi sesuai hak akses HR maupun karyawan. Hasil pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-003 berikut ini.

Tabel 4.5 *User Acceptance Testing* FR-003

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-003	Pengujian fungsionalitas dashboard laporan untuk memastikan HR dan karyawan dapat melihat ringkasan data absensi, statistik kehadiran, grafik laporan, serta informasi rekap absensi sesuai hak akses masing-masing.	HR melihat <i>dashboard</i> laporan absen	<i>Dashboard</i> laporan HR tampil	Sesuai	✓	
		Karyawan melihat <i>dashboard</i> absensi pribadi	<i>Dashboard</i> absensi karyawan tampil	Sesuai	✓	
		Perubahan filter laporan <i>dashboard</i>	Data <i>dashboard</i> berhasil difilter	Sesuai	✓	

Persentase test case yang lolos = $3/3 = 100\%$

Persentase test case yang gagal = $0/3 = 0\%$

Persentase test case yang telah diuji = $3/3 = 100\%$

Persentase test case yang belum diuji = $0/3 = 0\%$

Jumlah total test case dalam pengujian = 3

4. UAT FR-004

Pengujian UAT FR-004 dilakukan untuk memastikan fitur melihat laporan pada aplikasi LOOK-IN dapat menampilkan data laporan absensi sesuai hak akses HR dan karyawan, termasuk fitur filter dan pencarian data laporan. Hasil pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-004 berikut ini.

Tabel 4.6 *User Acceptance Testing* FR-001

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-004	Pengujian fungsionalitas melihat laporan untuk memastikan karyawan dan HR dapat mengakses laporan yang tersedia sesuai hak akses masing-masing.	HR melihat laporan absensi	Laporan absensi HR tampil	Sesuai	✓	
		Karyawan melihat laporan pribadi	Laporan pribadi tampil	Sesuai	✓	
		Filter laporan berdasarkan tanggal	Data laporan berhasil difilter	Sesuai	✓	
		Pencarian data laporan oleh HR	Data laporan ditemukan	Sesuai	✓	
		Pencarian data laporan oleh HR yang tidak ada	Sistem menampilkan data tidak ditemukan	Sesuai	✓	

Persentase test case yang lolos = $5/5 = 100\%$

Persentase test case yang gagal = $0/5 = 0\%$

Persentase test case yang telah diuji = $5/5 = 100\%$

Persentase test case yang belum diuji = $0/5 = 0\%$

Jumlah total test case dalam pengujian = 5

5. UAT FR-005

Pengujian UAT FR-005 dilakukan untuk memastikan fitur mengelola profil pada aplikasi LOOK-IN dapat digunakan oleh HR dan karyawan untuk melihat, mengubah, dan melengkapi data pribadi, termasuk perubahan password dan foto profil. Hasil pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-005 berikut ini.

Tabel 4.7 *User Acceptance Testing* FR-005

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-005	Pengujian fungsionalitas mengelola profil untuk memastikan HR dan karyawan dapat melihat, mengubah, dan melengkapi data pribadi pada menu profil, termasuk perubahan	Mengubah data profil	Data profil berhasil diperbarui	Sesuai	✓	
		Mengubah password berhasil	Password berhasil diperbarui	Sesuai	✓	
		Gagal mengubah password karena kata sandi lama salah	Sistem menolak perubahan password	Sesuai	✓	

	password dan foto profil.	Gagal mengubah password karena konfirmasi password tidak sesuai	Sistem menampilkan validasi password	Sesuai	✓	
		Validasi ketentuan password baru	Sistem memvalidasi password baru	Sesuai	✓	
		Password baru tidak boleh sama dengan password lama	Sistem menolak password yang sama	Sesuai		
		Mengubah foto profil	Foto profil berhasil diperbarui	Sesuai		

Persentase test case yang lolos = $7/7 = 100\%$

Persentase test case yang gagal = $0/7 = 0\%$

Persentase test case yang telah diuji = $7/7 = 100\%$

Persentase test case yang belum diuji = $0/7 = 0\%$

Jumlah total test case dalam pengujian = 7

6. UAT FR-006

Pengujian UAT FR-006 dilakukan untuk memastikan fitur mengubah kata sandi pada aplikasi LOOK-IN dapat digunakan dengan aman oleh HR dan karyawan sesuai validasi sistem yang telah ditentukan. Hasil pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-006 berikut ini.

Tabel 4.8 *User Acceptance Testing* FR-006

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-006	Pengujian fungsionalitas mengubah kata sandi untuk memastikan HR dan karyawan dapat mengganti kata sandi akun dengan aman sesuai validasi yang telah ditentukan.	Mengubah password berhasil	Password berhasil diperbarui	Sesuai	✓	
		Gagal mengubah password karena kata sandi lama salah	Sistem menolak perubahan password	Sesuai	✓	
		Gagal mengubah password karena konfirmasi password tidak sesuai	Sistem menampilkan validasi password	Sesuai	✓	
		Validasi ketentuan password baru	Sistem memvalidasi password baru	Sesuai	✓	
		Password baru tidak boleh sama dengan password lama	Sistem menolak password yang sama	Sesuai	✓	

Persentase test case yang lolos = $5/5 = 100\%$

Persentase test case yang gagal = $0/5 = 0\%$

Persentase test case yang telah diuji = $5/5 = 100\%$

Persentase test case yang belum diuji = $0/5 = 0\%$

Jumlah total test case dalam pengujian = 5

7. UAT FR-007

Pengujian UAT FR-007 dilakukan untuk memastikan fitur notifikasi pada aplikasi LOOK-IN dapat memberikan informasi aktivitas sistem kepada HR dan karyawan, termasuk perubahan status perizinan, navigasi ke halaman terkait, serta pengelolaan status notifikasi dibaca dan belum dibaca. Hasil pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-007 berikut ini.

Tabel 4.9 *User Acceptance Testing* FR-007

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-007	Pengujian fungsionalitas notifikasi untuk memastikan HR dan karyawan mendapatkan notifikasi sesuai aktivitas sistem, dapat membuka notifikasi untuk menuju halaman terkait, serta mengelola status notifikasi telah dibaca dan belum dibaca.	HR menerima notifikasi pengajuan perizinan	Notifikasi pengajuan tampil	Sesuai	✓	
		HR membuka notifikasi perizinan	Halaman perizinan berhasil dibuka	Sesuai	✓	
		Karyawan menerima notifikasi perubahan status perizinan approved	Notifikasi approved tampil	Sesuai	✓	
		Karyawan menerima notifikasi perubahan status perizinan reject	Notifikasi reject tampil	Sesuai	✓	

		Tidak ada notifikasi baru	Sistem menampilkan notifikasi kosong	Sesuai	✓	
		Menandai semua notifikasi sebagai dibaca	Semua notifikasi berubah menjadi dibaca	Sesuai	✓	

Persentase test case yang lolos = $6/6 = 100\%$

Persentase test case yang gagal = $0/6 = 0\%$

Persentase test case yang telah diuji = $6/6 = 100\%$

Persentase test case yang belum diuji = $0/6 = 0\%$

Jumlah total test case dalam pengujian = 6

8. UAT FR-008

Pengujian UAT FR-008 dilakukan untuk memastikan fitur absensi cepat pada aplikasi LOOK-IN dapat digunakan oleh karyawan tanpa login melalui verifikasi face recognition, kamera, lokasi GPS, dan attendance setting yang telah ditentukan oleh HR. Hasil pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-008 berikut ini.

Tabel 4.10 *User Acceptance Testing* FR-008

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-008	Pengujian fungsionalitas absensi cepat menggunakan face recognition	Berhasil melakukan absensi cepat	Absensi berhasil disimpan	Sesuai	✓	
		Gagal	Sistem	Sesuai	✓	

	untuk memastikan karyawan dapat melakukan absensi tanpa login melalui halaman login dengan verifikasi wajah, kamera, lokasi GPS, dan pengaturan attendance setting yang telah ditentukan oleh HR.	absensi karena berada di luar radius lokasi	menolak absensi			
		Gagal absensi karena sudah check-in hari ini	Sistem menolak absensi duplikat	Sesuai	✓	
		Gagal absensi karena wajah tidak terdeteksi	Sistem menampilkan gagal verifikasi wajah	Sesuai	✓	

Persentase test case yang lolos = $4/4 = 100\%$

Persentase test case yang gagal = $0/4 = 0\%$

Persentase test case yang telah diuji = $4/4 = 100\%$

Persentase test case yang belum diuji = $0/4 = 0\%$

Jumlah total test case dalam pengujian = 4

9. UAT FR-009

Pengujian UAT FR-009 dilakukan untuk memastikan fitur absensi menggunakan webcam dan verifikasi lokasi GPS pada aplikasi LOOK-IN dapat digunakan oleh karyawan setelah login ke dalam sistem sesuai radius lokasi dan attendance setting yang telah ditentukan oleh HR. Hasil pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-009 berikut ini.

Tabel 4.11 *User Acceptance Testing* FR-008

Kode	Deskripsi	Skenario	Hasil	Hasil	Status
------	-----------	----------	-------	-------	--------

			Diharapkan	Aktual	B	G
FR-009	Pengujian fungsionalitas absensi menggunakan webcam dan verifikasi lokasi GPS untuk memastikan karyawan dapat melakukan check-in setelah login ke dalam sistem berdasarkan radius lokasi dan attendance setting yang telah ditentukan oleh HR	Berhasil melakukan absensi	Absensi berhasil disimpan	Sesuai	✓	
		Gagal absensi karena berada di luar radius lokasi	Sistem menolak absensi	Sesuai	✓	
		Gagal absensi karena sudah check-in hari ini	Sistem menolak absensi duplikat	Sesuai	✓	
		Gagal absensi karena wajah tidak terdeteksi	Sistem menampilkan gagal verifikasi wajah	Sesuai	✓	

Persentase test case yang lolos = $4/4 = 100\%$

Persentase test case yang gagal = $0/4 = 0\%$

Persentase test case yang telah diuji = $4/4 = 100\%$

Persentase test case yang belum diuji = $0/4 = 0\%$

Jumlah total test case dalam pengujian = 4

10. UAT FR-010

Pengujian UAT FR-010 dilakukan untuk memastikan fitur pengajuan persetujuan pada aplikasi LOOK-IN dapat digunakan oleh karyawan untuk membuat, melihat, mengubah, dan menghapus pengajuan izin kerja sesuai kebutuhan sistem. Hasil pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-010 berikut ini.

Tabel 4.12 *User Acceptance Testing* FR-008

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-010	Pengujian fungsionalitas pengajuan persetujuan untuk memastikan karyawan dapat membuat pengajuan izin kerja dengan menyertakan alasan, tanggal izin, dan durasi hari sesuai kebutuhan sistem.	Berhasil membuat pengajuan izin	Pengajuan izin berhasil disimpan	Sesuai	✓	
		Melihat detail pengajuan izin	Detail pengajuan berhasil ditampilkan	Sesuai	✓	
		Mengubah pengajuan izin yang masih menunggu persetujuan	Pengajuan izin berhasil diperbarui	Sesuai	✓	
		Menghapus pengajuan izin yang masih menunggu persetujuan	Pengajuan izin berhasil dihapus	Sesuai	✓	

Persentase test case yang lolos = $4/4 = 100\%$

Persentase test case yang gagal = $0/4 = 0\%$

Persentase test case yang telah diuji = $4/4 = 100\%$

Persentase test case yang belum diuji = $0/4 = 0\%$

Jumlah total test case dalam pengujian = 4

11. UAT FR-011

Pengujian UAT FR-010 dilakukan untuk memastikan fitur pengajuan persetujuan pada aplikasi LOOK-IN dapat digunakan oleh karyawan untuk membuat, melihat, mengubah, dan menghapus pengajuan izin kerja sesuai kebutuhan sistem. Hasil pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-010 berikut ini.

Tabel 4.13 *User Acceptance Testing* FR-008

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-011	Pengujian fungsionalitas penambahan catatan keterlambatan untuk memastikan karyawan yang terlambat melakukan absensi wajib menuliskan alasan keterlambatan sesuai durasi keterlambatan yang dihitung sistem	Tombol tambah catatan muncul ketika terlambat	Tombol tambah berhasil tampil	Sesuai	✓	
		Membuka form catatan keterlambatan	Form catatan berhasil ditampilkan	Sesuai	✓	
		Berhasil mengirim catatan keterlambatan	Catatan berhasil disimpan	Sesuai	✓	
		Gagal mengirim catatan karena jumlah kata kurang	Sistem menolak pengiriman catatan	Sesuai	✓	
		Perhitungan jumlah kata berdasarkan durasi keterlambatan	Jumlah kata dihitung sesuai keterlambatan	Sesuai	✓	

		Tombol tambah tidak muncul jika tidak terlambat	Tombol tambah tidak ditampilkan	Sesuai	✓	
--	--	---	---------------------------------	--------	---	--

Persentase test case yang lolos = $6/6 = 100\%$

Persentase test case yang gagal = $0/6 = 0\%$

Persentase test case yang telah diuji = $6/6 = 100\%$

Persentase test case yang belum diuji = $0/6 = 0\%$

Jumlah total test case dalam pengujian = 6

12. UAT FR-012

Pengujian UAT FR-012 dilakukan untuk memastikan fitur pendaftaran akun pada aplikasi LOOK-IN dapat digunakan oleh HR untuk membuat akun baru dengan mengisi data username, nama lengkap, *email*, dan *password* sesuai validasi sistem. Selain itu, pengujian juga dilakukan untuk memastikan mekanisme kode undangan berfungsi dengan baik ketika akun HR telah tersedia di dalam sistem. Hasil pengujian ditunjukkan pada Tabel *User Acceptance Testing* FR-012 berikut ini.

Tabel 4.14 *User Acceptance Testing* FR-008

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-012	Pengujian fungsionalitas pendaftaran akun untuk memastikan HR dapat membuat akun baru	Berhasil melakukan pendaftaran akun	Akun berhasil dibuat	Sesuai	✓	
		Gagal daftar karena	Sistem menolak username	Sesuai	✓	

	ke dalam sistem dengan mengisi data username, nama lengkap, email, dan password	username sudah digunakan	duplikat			
		Gagal daftar karena field kosong	Sistem menampilkan validasi input	Sesuai	✓	
		Validasi format email	Sistem memvalidasi format email	Sesuai	✓	
		Gagal daftar karena kode undangan tidak valid	Sistem menolak pendaftaran dan menampilkan pesan kesalahan	Sesuai	✓	
		Berhasil daftar menggunakan kode undangan yang valid	Sistem membuat akun dan mengarahkan pengguna ke halaman login	Sesuai	✓	

Persentase test case yang lolos = $6/6 = 100\%$

Persentase test case yang gagal = $0/6 = 0\%$

Persentase test case yang telah diuji = $6/6 = 100\%$

Persentase test case yang belum diuji = $0/6 = 0\%$

Jumlah total test case dalam pengujian = 6

13. UAT FR-013

Pengujian UAT FR-013 dilakukan untuk memastikan fitur pengelolaan pengguna pada aplikasi LOOK-IN dapat digunakan oleh HR untuk menambahkan, memperbarui, dan menghapus data

pengguna sesuai validasi sistem. Hasil pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-013 berikut ini.

Tabel 4.15 *User Acceptance Testing* FR-008

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-013	Pengujian fungsionalitas pengelolaan pengguna untuk memastikan HR dapat menambahkan, memperbarui, dan menghapus data pengguna dalam sistem	Berhasil menambahkan data pengguna	Data pengguna berhasil ditambahkan	Sesuai	✓	
		Berhasil memperbaiki data pengguna	Data pengguna berhasil diperbarui	Sesuai	✓	
		Berhasil menghapus data pengguna	Data pengguna berhasil dihapus	Sesuai	✓	
		Gagal menambahkan pengguna karena field kosong	Sistem menampilkan validasi input	Sesuai	✓	
		Validasi format email	Sistem memvalidasi format email	Sesuai	✓	

Persentase test case yang lolos = $5/5 = 100\%$

Persentase test case yang gagal = $0/5 = 0\%$

Persentase test case yang telah diuji = $5/5 = 100\%$

Persentase test case yang belum diuji = $0/5 = 0\%$

Jumlah total test case dalam pengujian = 5

14. UAT FR-014

Pengujian UAT FR-014 dilakukan untuk memastikan fitur pengelolaan dataset pengguna pada aplikasi LOOK-IN dapat digunakan oleh HR untuk menambahkan dan menghapus *dataset* wajah pengguna yang digunakan dalam proses training *face recognition*. Hasil pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-014 berikut ini.

Tabel 4.16 *User Acceptance Testing* FR-008

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-014	Pengujian fungsionalitas pengelolaan dataset pengguna untuk memastikan HR dapat menambahkan dan menghapus dataset wajah pengguna yang digunakan untuk proses training <i>face recognition</i>	Berhasil menambahkan dataset pengguna	Dataset berhasil ditambahkan	Sesuai	✓	
		Gagal menambahkan dataset karena pengguna belum dipilih	Sistem menampilkan validasi pengguna	Sesuai	✓	
		Gagal menambahkan dataset karena foto belum diupload	Sistem menampilkan validasi foto	Sesuai	✓	
		Berhasil menghapus dataset pengguna	Dataset berhasil dihapus	Sesuai	✓	

Persentase test case yang lolos = $4/4 = 100\%$

Persentase test case yang gagal = $0/4 = 0\%$

Persentase test case yang telah diuji = $4/4 = 100\%$

Persentase test case yang belum diuji = $0/4 = 0\%$

Jumlah total test case dalam pengujian = 4

15. UAT FR-015

Pengujian UAT FR-015 dilakukan untuk memastikan fitur penanganan persetujuan izin pada aplikasi LOOK-IN dapat digunakan oleh HR untuk menerima, menolak, dan melihat detail pengajuan izin yang diajukan oleh karyawan. Hasil pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-015 berikut ini.

Tabel 4.17 *User Acceptance Testing* FR-008

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-015	Pengujian fungsionalitas penanganan persetujuan izin untuk memastikan HR dapat menerima atau menolak pengajuan izin yang diajukan oleh karyawan	HR menerima pengajuan izin	Status pengajuan berhasil diterima	Sesuai	✓	
		HR menolak pengajuan izin	Status pengajuan berhasil ditolak	Sesuai	✓	
		Melihat detail pengajuan izin	Detail pengajuan berhasil ditampilkan	Sesuai	✓	
		Tidak ada pengajuan izin	Sistem menampilkan data kosong	Sesuai	✓	

Persentase test case yang lolos = $4/4 = 100\%$

Persentase test case yang gagal = $0/4 = 0\%$

Persentase test case yang telah diuji = $4/4 = 100\%$

Persentase test case yang belum diuji = $0/4 = 0\%$

Jumlah total test case dalam pengujian = 4

16. UAT FR-016

Pengujian UAT FR-016 dilakukan untuk memastikan fitur export riwayat absensi pada aplikasi LOOK-IN dapat digunakan oleh HR untuk mengunduh laporan absensi dalam format PDF dan Excel berdasarkan filter tanggal tertentu. Hasil pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-016 berikut ini.

Tabel 4.18 *User Acceptance Testing* FR-008

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-016	Pengujian fungsionalitas export riwayat absensi untuk memastikan HR dapat mengunduh laporan riwayat absensi dalam format PDF dan Excel berdasarkan filter tanggal tertentu	Export riwayat absensi dalam format PDF	File PDF berhasil diunduh	Sesuai	✓	
		Export riwayat absensi dalam format Excel	File Excel berhasil diunduh	Sesuai	✓	
		Filter laporan berdasarkan tanggal	Data laporan berhasil difilter	Sesuai	✓	

Persentase test case yang lolos = $3/3 = 100\%$

Persentase test case yang gagal = $0/3 = 0\%$

Persentase test case yang telah diuji = $3/3 = 100\%$

Persentase test case yang belum diuji = $0/3 = 0\%$

Jumlah total test case dalam pengujian = 3

17. UAT FR-017

Pengujian UAT FR-017 dilakukan untuk memastikan fitur pengelolaan peraturan absensi pada aplikasi LOOK-IN dapat digunakan oleh HR untuk mengatur lokasi GPS absensi, radius absensi, jam shift kerja, jam masuk, dan jam keluar sesuai kebutuhan sistem. Hasil pengujian ditunjukkan pada tabel *User Acceptance Testing* FR-017 berikut ini.

Tabel 4.19 *User Acceptance Testing* FR-008

Kode	Deskripsi	Skenario	Hasil Diharapkan	Hasil Aktual	Status	
					B	G
FR-017	Pengujian fungsionalitas pengelolaan peraturan absensi untuk memastikan HR dapat mengatur lokasi GPS absensi, radius absensi, jam shift kerja, jam masuk, dan jam keluar pada sistem	Berhasil menambahkan pengaturan absensi	Pengaturan absensi berhasil disimpan	Sesuai	✓	
		Berhasil memperbaiki pengaturan absensi	Pengaturan absensi berhasil diperbarui	Sesuai	✓	

Persentase test case yang lolos = $2/2 = 100\%$

Persentase test case yang gagal = $0/2 = 0\%$

Persentase test case yang telah diuji = $2/2 = 100\%$

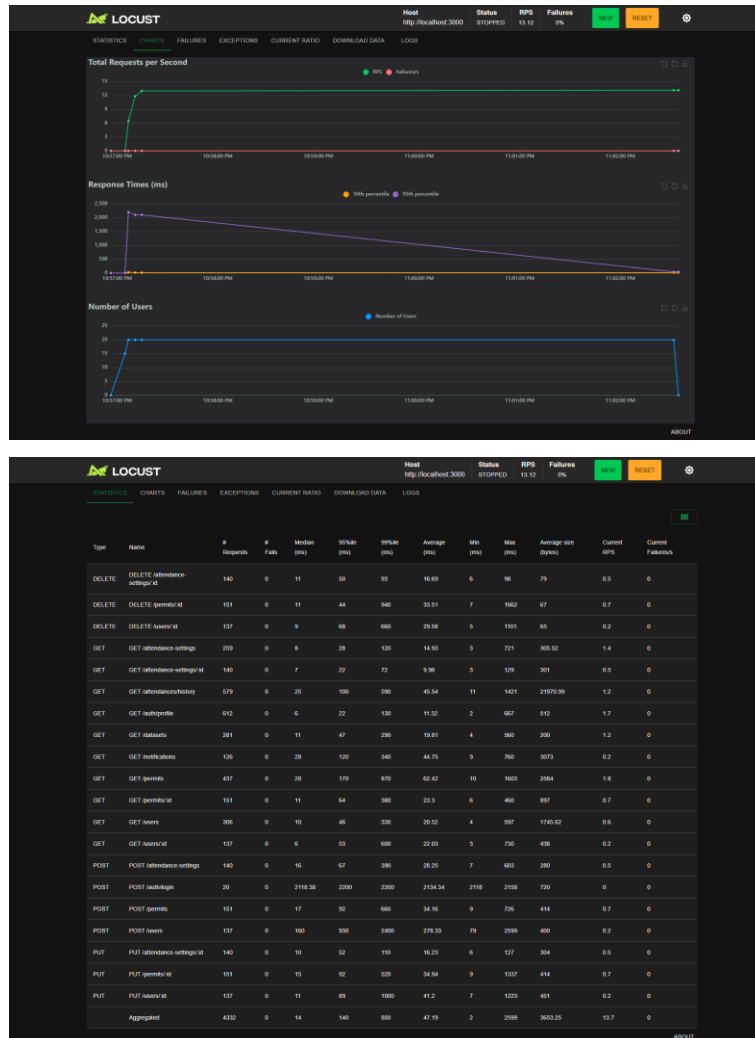
Persentase test case yang belum diuji = $0/2 = 0\%$

Jumlah total test case dalam pengujian = 2

4.2.2 Load Testing

Load testing dilakukan untuk menguji performa dan kestabilan sistem ketika diakses oleh banyak pengguna secara bersamaan. Pengujian ini bertujuan untuk mengetahui kemampuan sistem dalam menangani *request* secara *real-time*, mengukur response time, serta memastikan tidak terjadi kegagalan layanan ketika sistem menerima beban akses tertentu.

Pada penelitian ini, *load testing* dilakukan menggunakan *Locust* dengan melakukan simulasi beberapa pengguna virtual yang mengakses *endpoint* API sistem secara bersamaan. Pengujian dilakukan pada beberapa fitur utama seperti autentikasi, pengelolaan pengguna, riwayat absensi, notifikasi, pengajuan izin, dan pengaturan absensi. Parameter yang diamati pada pengujian ini meliputi jumlah *request*, *response time*, *request per second* (RPS), serta tingkat kegagalan request (*failure rate*).



Gambar 4.23 Hasil Pengujian *Load Testing* Menggunakan *Locust*

Gambar 4.23 menampilkan hasil pengujian *load testing* menggunakan *Locust* dengan simulasi 20 pengguna virtual dan *spawn rate* sebesar 5 pengguna per detik. Berdasarkan hasil pengujian, sistem mampu menangani request secara stabil dengan tingkat kegagalan request sebesar 0% serta response time yang masih berada pada batas yang dapat diterima selama proses pengujian berlangsung. Berdasarkan hasil *load testing* menggunakan *Locust*, sistem berhasil menangani total 4.332 request dengan tingkat kegagalan sebesar 0%. Rata-rata response time keseluruhan sistem berada pada 47.19 ms yang menunjukkan bahwa sistem mampu berjalan dengan stabil saat diakses secara bersamaan oleh beberapa pengguna. Endpoint login memiliki response time

tertinggi karena proses autentikasi dan validasi data pengguna, sedangkan endpoint lainnya menunjukkan performa yang cukup baik dan stabil selama proses pengujian berlangsung.

Tabel 4.20 Hasil Pengujian *Load Testing* Menggunakan *Locust*

Endpoint	Method	Total Request	Avg Response Time	Max Response Time	Failure
/attendance-settings/:id	DEL	140	16.69	98	0%
/permit/:id	DEL	151	33.51	1662	0%
/users/:id	DEL	137	29.58	1101	0%
/attendance-settings	GET	259	14.93	721	0%
/attendance-settings/:id	GET	140	9.98	129	0%
/attendance-history	GET	579	45.54	1421	0%
/auth/profile	GET	612	11.32	667	0%
/datasets	GET	281	19.81	560	0%
/notifications	GET	126	44.75	760	0%
/permits	GET	437	62.42	1603	0%
/permits/:id	GET	151	23.3	460	0%
/users	GET	306	20.52	597	0%
/users/:id	GET	137	22.03	730	0%

/attendance-settings	POST	140	28.25	603	0%
/auth/login	POST	20	2134.34	2158	0%
/permits	POST	151	34.16	726	0%
/users	POST	137	278.33	2599	0%
/attendance-settings/:id	PUT	140	16.23	127	0%
/permit/:id	PUT	151	34.94	1337	0%
/users/:id	PUT	137	41.2	1223	0%
Aggregated		4332	47.19	2599	0%

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, pengembangan, dan pengujian aplikasi LOOK-IN di Perusahaan Jasa Teknologi Informasi, maka dapat disimpulkan bahwa tujuan penelitian ini telah berhasil dicapai, yaitu membangun sistem absensi berbasis website menggunakan teknologi *face recognition* untuk membantu proses absensi karyawan secara lebih efektif dan otomatis. Kesimpulan utama yang diperoleh adalah sebagai berikut:

1. Sistem absensi berbasis website berhasil dirancang dan dikembangkan untuk membantu proses pencatatan kehadiran karyawan secara digital dan *real-time* sehingga proses absensi menjadi lebih efisien dibandingkan metode manual sebelumnya.
2. Implementasi teknologi face recognition menggunakan metode LBPH dan YOLO berhasil diterapkan untuk melakukan proses verifikasi identitas karyawan melalui webcam secara otomatis sehingga membantu meningkatkan keamanan dan meminimalkan potensi kecurangan dalam absensi.
3. Sistem berhasil mengintegrasikan fitur GPS untuk memvalidasi lokasi pengguna sehingga absensi hanya dapat dilakukan pada area yang telah ditentukan oleh pihak HR.
4. Fitur-fitur utama sistem seperti autentikasi pengguna, absensi cepat, riwayat absensi, pengajuan izin, notifikasi, pengelolaan pengguna, pengelolaan dataset wajah, export laporan, dan pengaturan absensi berhasil diimplementasikan sesuai kebutuhan fungsional yang telah dirancang.
5. Berdasarkan hasil pengujian sistem dan *User Acceptance Test* (UAT), aplikasi LOOK-IN mampu menjalankan seluruh fungsi utama sesuai kebutuhan pengguna dan dapat digunakan sebagai solusi absensi digital di Perusahaan Jasa Teknologi Informasi.

Dengan tercapainya tujuan tersebut, sistem LOOK-IN diharapkan dapat membantu perusahaan dalam meningkatkan efisiensi, keamanan, dan akurasi proses absensi karyawan berbasis teknologi *face recognition*.

5.2 Saran

Berdasarkan hasil pengembangan dan pengujian aplikasi LOOK-IN yang telah dilakukan, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan sistem di masa mendatang, yaitu sebagai berikut:

1. Sistem *face recognition* dapat dikembangkan menggunakan model *deep learning* yang lebih kompleks seperti *FaceNet* atau *ArcFace* agar tingkat akurasi pengenalan wajah menjadi lebih baik terutama pada kondisi pencahayaan rendah, perubahan posisi wajah, maupun penggunaan aksesoris seperti masker dan kacamata.
2. Pengujian sistem disarankan dilakukan dalam skala yang lebih besar dengan jumlah pengguna dan *dataset* wajah yang lebih banyak agar dapat diketahui tingkat performa, akurasi, serta kestabilan sistem dalam penggunaan jangka panjang secara nyata di lingkungan perusahaan.
3. Sistem dapat dikembangkan dengan fitur notifikasi otomatis untuk mengingatkan pengguna apabila belum melakukan absensi masuk atau absensi pulang sesuai jam kerja yang telah ditentukan oleh pihak HR.
4. Pengembangan selanjutnya dapat menambahkan dukungan aplikasi *mobile* berbasis *Android* atau *iOS* agar proses absensi dan pengelolaan data dapat dilakukan secara lebih fleksibel melalui perangkat *smartphone*.

Dengan adanya pengembangan lebih lanjut terhadap fitur, keamanan, dan performa sistem, aplikasi LOOK-IN diharapkan dapat menjadi solusi absensi digital yang lebih optimal, akurat, dan adaptif terhadap kebutuhan perusahaan di masa mendatang.

DAFTAR PUSTAKA

- Abdussalaam, F., & Ramdani, A. (2023). PERANCANGAN SISTEM INFORMASI MANAJEMEN PRAKTEK KERJA LAPANGAN BERBASIS WEB MENGGUNAKAN METODE AGILE. *INFOKOM (Informatika & Komputer)*, 10(2). <https://doi.org/10.56689/infokom.v10i2.950>
- Adusumalli, H., Kalyani, D., Sri, R. K., Pratapteja, M., & Rao, P. V. R. D. P. (2021). Face Mask Detection Using OpenCV. *Proceedings of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, ICICV 2021*. <https://doi.org/10.1109/ICICV50876.2021.9388375>
- Agustini, A. A. A. W., & Dewi, I. A. U. (2022). PERANCANGAN SISTEM INFORMASI BERBASIS WEB MENGGUNAKAN WATERFALL MODEL (STUDI KASUS : LPD DESA ADAT SUMERTA). *RESI : Jurnal Riset Sistem Informasi*, 1(1). <https://doi.org/10.32795/resi.v1i1.2940>
- Alfan Rosid, M., & Sumarno, S. (2021). RANCANGAN SISTEM PRESENSI MAHASISWA MENGGUNAKAN QR CODE DENGAN FITUR GEOLOCATION DAN ENKRIPSI AES. *JIKA (Jurnal Informatika)*, 5(2). <https://doi.org/10.31000/jika.v5i2.4052>
- Andre, & Suciadi, M. F. (2022). The online attendance system models for educational institutions. *AIP Conference Proceedings*, 2470. <https://doi.org/10.1063/5.0080180>
- Aryputra, E. R., Hanggara, B. T., & Wicaksono, S. A. (2020). Pengembangan Sistem Informasi Akademik Sekolah Sepak Bola berbasis Website menggunakan Metode Waterfall (Studi Kasus: SSB Summersari FC). *Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 4(5).
- Boulila, W., Alzahem, A., Almoudi, A., Afifi, M., Alturki, I., & Driss, M. (2021). A Deep Learning-based Approach for Real-time Facemask Detection. *Proceedings - 20th IEEE International Conference on Machine Learning and Applications, ICMLA 2021*. <https://doi.org/10.1109/ICMLA52953.2021.00238>
- Das, A., Wasif Ansari, M., & Basak, R. (2020). Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV. *2020 IEEE 17th India Council International Conference, INDICON 2020*. <https://doi.org/10.1109/INDICON49873.2020.9342585>
- Evelyn, Adipranata, R., & Gunadi, K. (2022). Sistem Presensi Mahasiswa Menggunakan Face Recognition Dengan Metode Facenet Pada Android. *Jurnal Infra, Vol 10, No.*
- Golec, M., & Plechawska-Wójcik, M. (2022). Comparative analysis of frameworks using TypeScript to build server applications. *Journal of Computer Sciences Institute*, 23. <https://doi.org/10.35784/jcsi.2910>
- Gupta, N., Sharma, P., Deep, V., & Shukla, V. K. (2020). Automated Attendance System Using OpenCV. *ICRITO 2020 - IEEE 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*. <https://doi.org/10.1109/ICRITO48877.2020.9197936>
- Handayani, H., Ayulya, A. M., Faizah, K. U., Wulan, D., & Rozan, M. F. (2023). Perancangan Sistem Informasi Inventory Barang Berbasis Web Menggunakan Metode Agile Software Development. *Jurnal Testing Dan Implementasi Sistem Informasi*, 1(1). <https://doi.org/10.55583/jtisi.v1i1.324>
- Hasan, R. T. H., & Sallow, A. B. (2021). Face Detection and Recognition Using OpenCV. *Journal of Soft Computing and Data Mining*, 2(2). <https://doi.org/10.30880/jscdm.2021.02.02.008>

- Hendrawan, J. J. (2023). DIGITALISASI USAHA MIKRO KECIL DAN MENENGAH DI DESA MELALUI APLIKASI KEDE DESA BERBASIS WEB. *JSR : Jaringan Sistem Informasi Robotik*, 7(1). <https://doi.org/10.58486/jsr.v7i1.226>
- Hidayanto, K., & Fauzi, R. (2025). IMPLEMENTASI APLIKASI E-KEHADIRAN BERBASIS FACE RECOGNITION DAN GPS MENGGUNAKAN METODE AGILE DEVELOPMENT. *JURNAL COMASIE*, 12(02).
- Hidayat, A. (2022). Perancangan Sistem Informasi Warning Stock Sparepart Mesin Produksi Berbasis Web Pada PT Kobayashi Eglin. *Scientia Sacra: Jurnal Sains, Teknologi Dan Masyarakat*, 2(1).
- Larasati, I., Yusril, A. N., & Zukri, P. Al. (2021). Systematic Literature Review Analisis Metode Agile Dalam Pengembangan Aplikasi Mobile. *SISTEMASI*, 10(2). <https://doi.org/10.32520/stmsi.v10i2.1237>
- Nova, S. H., Widodo, A. P., & Warsito, B. (2022). Analisis Metode Agile pada Pengembangan Sistem Informasi Berbasis Website: Systematic Literature Review. *Techno.Com*, 21(1). <https://doi.org/10.33633/tc.v21i1.5659>
- Oloyede, M. O., Hancke, G. P., & Myburgh, H. C. (2020). A review on face recognition systems: recent approaches and challenges. *Multimedia Tools and Applications*, 79(37–38). <https://doi.org/10.1007/s11042-020-09261-2>
- Pricillia, T., & Zulfachmi. (2021). Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD). *Jurnal Bangkit Indonesia*, 10(1). <https://doi.org/10.52771/bangkitindonesia.v10i1.153>
- Prof. Dr. Paul Mccullagh. (2023). Face detection by using Haar Cascade Classifier. *Wasit Journal of Computer and Mathematics Science*, 2(1). <https://doi.org/10.31185/wjcm.109>
- Putra, S. P., Fitri, I., & Ningsih, S. (2021). Absensi Pengenalan Wajah Menggunakan Menggunakan Algoritma Eigenface Berbasis Web. *Journal of Applied Informatics and Computing*, 5(1). <https://doi.org/10.30871/jaic.v5i1.2711>
- Ramadhan, E. R., Prihandani, K., Voutama, A., Singaperbangsa, U., & Abstract, K. (2023). Penerapan Metode Agile Pada Development Aplikasi Pengelolaan Data Magang Berbasis Web Menggunakan Framework Laravel. *Jurnal Ilmiah Wahana Pendidikan*, 9(7).
- Santoso, B., & Kristianto, R. P. (2020). IMPLEMENTASI PENGGUNAAN OPENCV PADA FACE RECOGNITION UNTUK SISTEM PRESENSI PERKULIAHAN MAHASISWA. *SISTEMASI*, 9(2). <https://doi.org/10.32520/stmsi.v9i2.822>
- Saputra, T., Utari, A., Teisnajaya, U., & Twenty Augustine, G. (2023). Sistem Absensi Karyawan Menggunakan Self Potrait dan Geolocation Pada PT Sucofindo Palembang. *Klik - Jurnal Ilmu Komputer*, 4(2). <https://doi.org/10.56869/klik.v4i2.553>
- Septyanto, K., Hamid, M. A., & Aribowo, D. (2020). Pengembangan E-Learning Berbasis Website menggunakan Metode Waterfall. *Elinvo (Electronics, Informatics, and Vocational Education)*, 5(1). <https://doi.org/10.21831/elinvo.v5i1.31054>
- Sonny, S., & Rizki, S. N. (2021). PENGEMBANGAN SISTEM PRESENSI KARYAWAN DENGAN TEKNOLOGI GPS BERBASIS WEB PADA PT BPR DANA MAKMUR BATAM. *Jurnal Comasie*, 6(2).
- Sonny Sonny, S. N. R. (2021). pengembangan sistem presensi karyawan dengan teknologi GPS berbasis web. *Jurnal Comasie*, 6(2).

- Sugeng, S., & Mulyana, A. (2022). Sistem Absensi Menggunakan Pengenalan Wajah (Face Recognition) Berbasis Web LAN. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 11(1). <https://doi.org/10.32736/sisfokom.v11i1.1371>
- Suhari, S., Faqih, A., & Basysyar, F. M. (2022). Sistem Informasi Kepegawaian Menggunakan Metode Agile Development di CV. Angkasa Raya. *Jurnal Teknologi Dan Informasi*, 12(1). <https://doi.org/10.34010/jati.v12i1.6622>
- Sunardi, S., Yudhana, A., & Talib, M. A. (2022). Perancangan Sistem Pengenalan Wajah untuk Keamanan Ruangan Menggunakan Metode Local Binary Pattern Histogram. *Jurnal Teknologi Elektro*, 13(2). <https://doi.org/10.22441/jte.2022.v13i2.010>
- Susim, T., & Darujati, C. (2021). Pengolahan Citra untuk Pengenalan Wajah (Face Recognition) Menggunakan OpenCV. *Jurnal Syntax Admiration*, 2(3). <https://doi.org/10.46799/jsa.v2i3.202>
- Suwarno, S., & Kevin, K. (2020). Analysis of Face Recognition Algorithm: Dlib and OpenCV. *JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING*, 4(1). <https://doi.org/10.31289/jite.v4i1.3865>
- Ukamaka Betrand, C., Juliet Onyema, C., Eberechi Benson-Emenike, M., & Allswell Kelechi, D. (2023). Authentication System Using Biometric Data for Face Recognition. *International Journal of Sustainable Development Research*. <https://doi.org/10.11648/j.ijdsr.20230904.12>
- Wahyuni, S., & Sulaeman, M. (2022). Penerapan Algoritma Deep Learning Untuk Sistem Absensi Kehadiran Deteksi Wajah Di PT Karya Komponen Presisi. *Jurnal Informatika SIMANTIK*, 7(1).
- Widiarta, I. M., Mulyanto, Y., & Sutrianto, A. (2023). Rancang Bangun Sistem Informasi Inventory Menggunakan Metode Agile Software Development (Studi Kasus Toko Nada). *Digital Transformation Technology (Digitech)*, 3(Maret).
- Yulianti, D., Triastomoro, I., & Sa'idah, S. (2022). IDENTIFIKASI PENGENALAN WAJAH UNTUK SISTEM PRESENSI MENGGUNAKAN METODE KNN (K-NEAREST NEIGHBOR). *Jurnal Teknik Informasi Dan Komputer (Tekinkom)*, 5(1). <https://doi.org/10.37600/tekinkom.v5i1.477>
- Yusuf, M., Ginardi, R. V. H., & Ahmadiyah, A. S. (2016). Rancang Bangun Aplikasi Absensi Perkuliahan Mahasiswa dengan Pengenalan Wajah. *Jurnal Teknik ITS*, 5(2). <https://doi.org/10.12962/j23373539.v5i2.17518>

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan Requirement

<https://drive.google.com/drive/folders/1G04jCkXNi6unJIFu2sie3HpqEI2iddcC?usp=sharing>

Lampiran B. Link Produk

<https://github.com/vanhdeventr1/look-in>

Lampiran C. Dokumen Pengujian UAT

<https://drive.google.com/drive/folders/1tG7y9NiBtgdxfqsyHErsz9MvSCgPKcs?usp=sharing>