

**DIGITALISASI SISTEM INFORMASI KARYAWAN DAN
PRESENSI MENGGUNAKAN *QR CODE* BERBASIS *ANDROID***

SKRIPSI



AJI MUKSEN

2022230033

**PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH
TAHUN 2026**

DIGITALISASI SISTEM INFORMASI KARYAWAN DAN PRESENSI MENGGUNAKAN *QR CODE* BERBASIS *ANDROID*

Untuk Memenuhi Salah Satu Syarat Menyusun Skripsi
Program Studi Informatika Universitas Prabumulih



AJI MUKSEN

2022230033

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI INFORMATIKA
UNIVERSITAS PRABUMULIH
2026**

HALAMAN PENGESAHAN

DIGITALISASI SISTEM INFORMASI KARYAWAN DAN PRESENSI MENGUNAKAN *QR CODE* BERBASIS *ANDROID*

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Mendapatkan Gelar Sarjana
Program Studi Informatika Universitas Prabumulih

Oleh :

Aji Muksen
2022230033

Prabumulih, 04 Juni 2026

Pembimbing I

Pembimbing II



Iwan Setiawan, S.Kom., M.Kom.
NUPTK. 1541769670130303



Hermizahadiwidastra, S.Kom., M.Kom.
NUPTK. 2863762663130192



Ketua Program Studi
Informatika

(Iwan Setiawan, S.Kom., M.Kom.)
NUPTK. 1541769670130303

HALAMAN PERSETUJUAN

Skripsi ini dengan judul “Digitalisasi Sistem Informasi Karyawan dan Presensi Menggunakan *QR Code* Berbasis *Android*” telah dipertahankan dan disetujui dihadapan Tim Penguji Skripsi Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih pada tanggal 04 Juni 2026.

Prabumulih, 04 Juni 2026

Tim Penguji Tugas Akhir

Ketua:

Iwan Setiawan, S.Kom., M.Kom.
NUPTK. 1541769670130303

()

Anggota:

1. Hermizahadiwidastra, S.Kom., M.Kom.
NUPTK. 2863762663130192

()

2. Suhartini, S.Kom., M.Kom.
NUPTK. 253475765823105

()

Mengetahui,
Dekan Fakultas Ilmu Komputer



(Khana Wijaya, S.Kom., M.Kom)
NUPTK. 7146768669130363

Ketua Program Studi Informatika



(Iwan Setiawan, S.Kom., M.Kom)
NUPTK. 1541769670130303

PERNYATAAN INTEGRITAS

Saya yang bertanda tangan di bawah ini

Nama : Aji Muksen

NIM : 2022230033

Program Studi : Informatika

Menyatakan dengan sebenarnya bahwa, skripsi ini benar-benar merupakan hasil karya saya dan tidak terdapat karya orang lain, apabila di kemudian hari terbukti atau dapat dibuktikan bahwa skripsi ini hasil plagiasi, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku atas perbuatan tersebut.



Prabumulih, 04 Juni 2026

Yang membuat pernyataan,



Aji Muksen

NIM. 2022230033

ABSTRAK

Digitalisasi Sistem Informasi Karyawan dan Presensi Menggunakan *QR Code* Berbasis *Android*

Aji Muksen, 2022230033, 2026

Digitalisasi sistem informasi karyawan dan presensi diperlukan untuk meningkatkan akurasi, efisiensi, dan transparansi pengelolaan kehadiran yang sebelumnya menggunakan mesin *fingerprint*. Sistem yang berjalan di PT Kendi Arindo masih memiliki kelemahan, seperti potensi titip absen, keterlambatan rekap data, serta keterbatasan akses informasi bagi pimpinan. Penelitian ini bertujuan merancang dan mengembangkan aplikasi *android* yang mengintegrasikan sistem informasi karyawan dengan presensi berbasis *QR code*. Penelitian menggunakan pendekatan kualitatif dengan teknik pengumpulan data melalui observasi dan wawancara kepada bagian *HR* dan karyawan untuk menggali kebutuhan pengguna dan permasalahan sistem lama. Metode pengembangan sistem yang digunakan adalah *design thinking* dengan tahapan *empathize*, *define*, *ideate*, *prototype*, dan *test*. Aplikasi yang dibangun menyediakan fitur pengelolaan data karyawan, pembuatan dan pemindaian *QR Code* untuk presensi, penyimpanan data kehadiran ke basis data, serta pembuatan laporan presensi yang dapat diakses pimpinan. Hasil pengujian menunjukkan bahwa aplikasi berjalan sesuai kebutuhan, mempermudah proses presensi, meningkatkan akurasi data kehadiran, serta mendukung transparansi dan akuntabilitas dalam pengelolaan sumber daya manusia di PT Kendi Arindo.

Kata Kunci: Digitalisasi, Sistem Informasi Karyawan, Presensi, *QR Code*, *Android*

ABSTRACT

Digitalization Employee Information and Attendance Systems Using Android-Based QR Codes

Aji Muksen, 2022230033, 2026

The digitalization of employee information and attendance systems is needed to improve the accuracy, efficiency, and transparency of attendance management, which previously relied on fingerprint devices. The existing system at PT Kendi Arindo still has several weaknesses, such as proxy attendance, delays in attendance recap, and limited access to information for management. This study aims to design and develop an Android-based application that integrates an employee information system with a QR Code-based attendance system. This research uses a qualitative approach, with data collected through observations and interviews with the HR department and employees to identify user needs and problems in the current system. The system is developed using the design thinking method, consisting of empathize, define, ideate, prototype, and test stages. The application provides employee data management, QR Code generation and scanning for attendance, automatic storage of attendance records, and attendance reports accessible to management. The test results show that the application operates according to the requirements, simplifies the attendance process, improves data accuracy, and supports transparency and accountability in human resource management at PT Kendi Arindo.

Keywords: *Digitalization, Employee Information System, Attendance, QR Code, Android*

KATA PENGANTAR

Puji dan syukur penulis ucapkan kehadirat Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Skripsi yang berjudul “Digitalisasi Sistem Informasi Karyawan dan Presensi Menggunakan *QR Code* Berbasis *Android*” Skripsi ini merupakan salah satu syarat untuk mendapatkan gelar Sarjana di Program Studi S1 Informatika Fakultas Ilmu Komputer Universitas Prabumulih. Penulis banyak menerima bantuan, arahan, dan bimbingan dari berbagai pihak. Penulis mengucapkan terima kasih kepada:

1. Ibu Dr. Yuniar Pratiwi, S.Si., M.Si selaku Rektor Universitas Prabumulih
2. Bapak Khana Wijaya, S.Kom., M.Kom selaku Dekan Fakultas Ilmu Komputer Universitas Prabumulih.
3. Ibu Suhartini, S.Kom., M.Kom selaku Wakil Dekan Keuangan Fakultas Ilmu Komputer Universitas Prabumulih.
4. Ibu Nur Aini Hutagalung, S.Kom., M.Kom selaku Wakil Dekan Kemahasiswaan Fakultas Ilmu Komputer Universitas Prabumulih.
5. Bapak Iwan Setiawan, S.Kom., M.Kom selaku Ketua Program Studi Informatika sekaligus sebagai pembimbing I.
6. Bapak Hermizahadiwidastra, S.Kom., M.Kom sebagai pembimbing II.
7. Bapak dan Ibu Dosen Program Studi Informatika. Fakultas Ilmu Komputer Universitas Prabumulih.
8. Bapak/Ibu staf administrasi Fakultas Ilmu Komputer Universitas Prabumulih.
9. Kepala Perusahaan, Pimpinan, dan seluruh Staf Perusahaan PT. Kendi Arindo yang telah memberikan izin dan membantu untuk melakukan penelitian.

Penulisan Skripsi ini masih terdapat kekurangan sehingga membutuhkan kritik dan saran yang bersifat membangun untuk kesempurnaan skripsi.

Prabumulih, 04 Juni 2026



Aji Muksen

2022230033

DAFTAR ISI

HALAMAN JUDUL DALAM	i
HALAMAN PENGESAHAN	ii
HALAMAN PERSETUJUAN	iii
PERNYATAAN INTEGRITAS.....	iv
ABSTRAK	v
<i>ABSTRACT</i>	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN	xv
DAFTAR SINGKATAN	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	5
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan.....	6
BAB II TINJAUAN PUSTAKA.....	8
2.1 Pengertian Digitalisasi	8
2.2 Pengertian Sistem Informasi Karyawan.....	8
2.3 Pengertian Presensi	9
2.4 Pengertian <i>QR Code</i>	10
2.5 Pengertian <i>Android</i>	11
2.6 Pengertian <i>Visual Studio Code</i>	12
2.7 Pengertian <i>Flutter</i>	13
2.8 Pengertian <i>Dart Programing Language</i>	14
2.9 Pengertian <i>Firebase</i>	15
2.10 Penelitian Terdahulu.....	16
2.11 Kerangka Berpikir Penelitian	18

BAB III METODE PENELITIAN	21
3.1 Objek, Subjek dan Jadwal Penelitian	21
3.2 Metode Penelitian dan Teori Penelitian.....	27
3.3 Sumber Data.....	29
3.4 Teknik Pengumpulan Data	32
3.5 Metode Pengembangan Sistem	34
3.6 Alat Bantu Perancangan	37
3.7 Pengujian Sistem.....	44
BAB IV ANALISA DAN PERANCANGAN	48
4.1 Analisa Sistem.....	48
4.2 Perancangan Sistem (Desain) yang diusulkan	53
BAB V HASIL DAN PEMBAHASAN	113
5.1 Spesifikasi Perangkat	113
5.2 Implementasi <i>Database</i>	114
5.3 Tampilan Antarmuka	121
5.4 Hasil Pengujian Sistem	150
BAB VI KESIMPULAN DAN SARAN	246
6.1 Kesimpulan	246
6.2 Saran.....	247
DAFTAR PUSTAKA.....	248
LAMPIRAN.....	254

DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu.....	16
Tabel 3. 1 <i>Use Case Diagram</i>	40
Tabel 3. 2 <i>Activity Diagram</i>	41
Tabel 4. 1 <i>Users</i>	79
Tabel 4. 2 <i>QR Code</i>	80
Tabel 4. 3 Presensi.....	80
Tabel 4. 4 Pengajuan	81
Tabel 4. 5 <i>Log Aktivitas</i>	82
Tabel 5. 1 Identifikasi <i>Statement</i> Fungsi Login	154
Tabel 5. 2 <i>Test Case</i> Fungsi <i>Login</i>	164
Tabel 5. 3 Identifikasi <i>Statement</i> Fungsi <i>Log Out</i>	166
Tabel 5. 4 <i>Test Case</i> Fungsi <i>Log Out</i>	170
Tabel 5. 5 Identifikasi <i>Statement</i> Fungsi Lupa <i>Password</i>	172
Tabel 5. 6 <i>Test Case</i> Fungsi Lupa <i>Password</i>	178
Tabel 5. 7 Identifikasi <i>Statement</i> Fungsi Presensi Masuk dan Pulang.....	183
Tabel 5. 8 <i>Test Case</i> Fungsi Presensi Masuk dan Pulang.....	192
Tabel 5. 9 Identifikasi <i>Statement</i> Fungsi Pengajuan Presensi	197
Tabel 5. 10 <i>Test Case</i> Fungsi Pengajuan Presensi.....	205
Tabel 5. 11 Identifikasi <i>Statement</i> Fungsi Persetujuan Presensi	210
Tabel 5. 12 <i>Test Case</i> Fungsi Persetujuan Presensi.....	216
Tabel 5. 13 Identifikasi <i>Statement</i> Fungsi Tambah Karyawan.....	220
Tabel 5. 14 <i>Test Case</i> Fungsi Tambah Karyawan	227
Tabel 5. 15 Identifikasi <i>Statement</i> Fungsi <i>Edit</i> Karyawan	231
Tabel 5. 16 <i>Test Case</i> Fungsi <i>Edit</i> Karyawan.....	237
Tabel 5. 17 Identifikasi <i>Statement</i> Fungsi Hapus Data Karyawan.....	239
Tabel 5. 18 <i>Test Case</i> Fungsi Hapus Data Karyawan	244

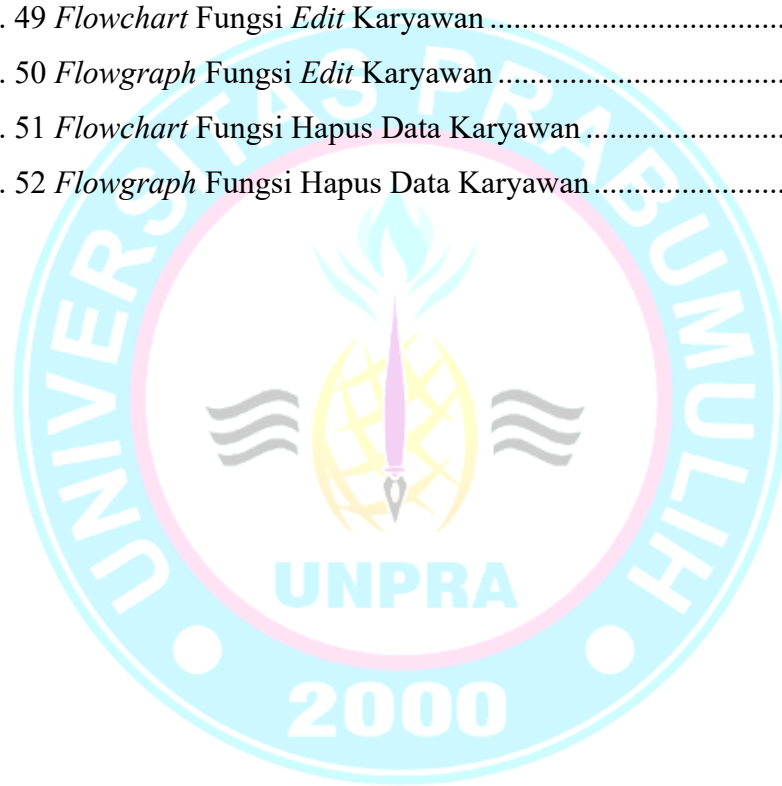
DAFTAR GAMBAR

Gambar 2. 1 Kerangka Berpikir Penelitian	19
Gambar 3. 1 Struktur Organisasi.....	24
Gambar 3. 2 Metode Pengembangan Sistem Design Thinking	36
Gambar 3. 3 <i>White Box Testing Process</i>	45
Gambar 4. 1 <i>Use Case Diagram</i> Utama.....	55
Gambar 4. 2 <i>Use Case Diagram</i> Karyawan.....	57
Gambar 4. 3 <i>Use Case Diagram HRD</i>	58
Gambar 4. 4 <i>Use Case Diagram</i> Pimpinan.....	60
Gambar 4. 5 <i>Activity Diagram Login</i> Karyawan yang diusulkan.	61
Gambar 4. 6 <i>Activity Diagram Reset Password</i> yang diusulkan.....	62
Gambar 4. 7 <i>Activity Diagram Presensi QR Code</i> yang diusulkan.....	63
Gambar 4. 8 <i>Activity Diagram Selfie Presensi</i> yang diusulkan.....	64
Gambar 4. 9 <i>Activity Diagram Pencatatan Lokasi (GPS)</i> yang diusulkan.....	65
Gambar 4. 10 <i>Activity Diagram Log Out</i> Karyawan yang diusulkan	66
Gambar 4. 11 <i>Activity Diagram Login HRD</i> yang diusulkan	67
Gambar 4. 12 <i>Activity Diagram Kelola Data Karyawan (Tambah)</i>	68
Gambar 4. 13 <i>Activity Diagram Kelola Data Karyawan (Edit)</i>	69
Gambar 4. 14 <i>Activity Diagram Kelola Data Karyawan(Hapus)</i>	70
Gambar 4. 15 <i>Activity Diagram Melihat Data Karyawan</i>	71
Gambar 4. 16 <i>Activity Diagram Generate QR Code</i> yang diusulkan	72
Gambar 4. 17 <i>Activity Diagram Kelola Presensi</i> yang diusulkan.....	73
Gambar 4. 18 <i>Activity Diagram Log Out HRD</i> yang diusulkan	74
Gambar 4. 19 <i>Activity Login Pimpinan</i> yang diusulkan.....	75
Gambar 4. 20 <i>Activity Diagram Laporan Presensi</i> yang diusulkan	76
Gambar 4. 21 <i>Activity Diagram Log Out Pimpinan</i> yang diusulkan	77
Gambar 4. 22 <i>Class Diagram</i> yang Diusulkan	78
Gambar 4. 23 Halaman <i>Splash Screen</i>	83
Gambar 4. 24 Halaman <i>Login Karyawan</i>	84
Gambar 4. 25 Halaman <i>Lupa Password</i>	85
Gambar 4. 26 Halaman <i>Dashboard Karyawan</i>	86

Gambar 4. 27 Halaman Presensi <i>Scan QR Code</i> Karyawan	87
Gambar 4. 28 Halaman Verifikasi Presensi.....	88
Gambar 4. 29 Halaman Verifikasi Presensi.....	89
Gambar 4. 30 Halaman Verifikasi Presensi (Cuti)	90
Gambar 4. 31 Halaman Presensi Berhasil (Masuk dan Pulang)	92
Gambar 4. 32 Halaman Riwayat Presensi.....	93
Gambar 4. 33 Halaman Profil dan <i>Logout</i> Karyawan.....	94
Gambar 4. 34 Halaman <i>Dashboard</i> HRD	95
Gambar 4. 35 Halaman Data Karyawan	96
Gambar 4. 36 Halaman Detail Karyawan	97
Gambar 4. 37 Halaman Tambah Karyawan	98
Gambar 4. 38 Halaman <i>Edit</i> Karyawan	99
Gambar 4. 39 Halaman Hapus Data Karyawan	100
Gambar 4. 40 Halaman Data Presensi.....	101
Gambar 4. 41 Halaman Persetujuan Presensi	102
Gambar 4. 42 Halaman <i>Generate QR Code</i>	103
Gambar 4. 43 Halaman Laporan Presensi.....	104
Gambar 4. 44 Halaman Profil dan <i>Logout HRD</i>	105
Gambar 4. 45 Halaman <i>Dashboard</i> Pimpinan	106
Gambar 4. 46 Halaman Data Karyawan	107
Gambar 4. 47 Halaman Detail Karyawan	108
Gambar 4. 48 Halaman Data Presensi.....	109
Gambar 4. 49 Halaman Persetujuan Presensi	110
Gambar 4. 50 Halaman Laporan Presensi.....	111
Gambar 4. 51 Halaman Profil dan <i>Logout</i> Pimpinan.....	112
Gambar 5. 1 Tabel Users	115
Gambar 5. 2 Tabel <i>QR Code</i>	116
Gambar 5. 3 Tabel Presensi.....	117
Gambar 5. 4 Tabel Pengajuan	118
Gambar 5. 5 Tabel <i>Log</i> Aktivitas	120
Gambar 5. 6 Halaman <i>Splash Screen</i>	121
Gambar 5. 7 Halaman <i>Login</i>	122

Gambar 5. 8 Halaman Lupa <i>Password</i>	123
Gambar 5. 9 Halaman <i>Dashboard</i> Karyawan	124
Gambar 5. 10 Halaman <i>Scan QR Code</i>	125
Gambar 5. 11 Halaman Verifikasi Presensi (Masuk dan Pulang)	126
Gambar 5. 12 Halaman Verifikasi Presensi (Sakit dan Izin).....	127
Gambar 5. 13 Halaman Verifikasi Presensi (Cuti)	128
Gambar 5. 14 Halaman Presensi Berhasil (Masuk dan Pulang)	129
Gambar 5. 15 Halaman Riwayat Presensi.....	130
Gambar 5. 16 Halaman Profil dan <i>Log Out</i> Karyawan.....	131
Gambar 5. 17 Halaman <i>Dashboard HRD</i>	132
Gambar 5. 18 Halaman Data Karyawan	133
Gambar 5. 19 Halaman Detail Karyawan	134
Gambar 5. 20 Halaman Tambah Karyawan	135
Gambar 5. 21 Halaman <i>Edit</i> Karyawan	136
Gambar 5. 22 Halaman Hapus Data Karyawan	137
Gambar 5. 23 Halaman Data Presensi.....	138
Gambar 5. 24 Halaman <i>Generate QR Code</i>	139
Gambar 5. 25 Halaman Persetujuan Presensi	140
Gambar 5. 26 Halaman Laporan Presensi.....	141
Gambar 5. 27 Halaman Profil dan <i>Log Out HRD</i>	142
Gambar 5. 28 Halaman <i>Dashboard</i> Pimpinan	143
Gambar 5. 29 Halaman Data Karyawan	144
Gambar 5. 30 Halaman Detail Karyawan	145
Gambar 5. 31 Halaman Data Presensi.....	146
Gambar 5. 32 Halaman Persetujuan Presensi	147
Gambar 5. 33 Halaman Laporan Presensi.....	148
Gambar 5. 34 Halaman Profil dan <i>Log Out</i> Pimpinan	149
Gambar 5. 35 <i>Flowchart</i> Fungsi <i>Login</i>	157
Gambar 5. 36 <i>Flowgraph</i> Fungsi <i>Login</i>	160
Gambar 5. 37 <i>Flowchart</i> Fungsi <i>Log Out</i>	167
Gambar 5. 38 <i>Flowgraph</i> Fungsi <i>Log Out</i>	168
Gambar 5. 39 <i>Flowchart</i> Fungsi Lupa <i>Password</i>	174

Gambar 5. 40 <i>Flowgraph</i> Fungsi Lupa <i>Password</i>	175
Gambar 5. 41 <i>Flowchart</i> Fungsi Presensi Masuk dan Pulang	186
Gambar 5. 42 <i>Flowgraph</i> Fungsi Presensi Masuk dan Pulang	188
Gambar 5. 43 <i>Flowchart</i> Fungsi Pengajuan Presensi	199
Gambar 5. 44 <i>Flowgraph</i> Fungsi Pengajuan Presensi	201
Gambar 5. 45 <i>Flowchart</i> Fungsi Persetujuan Presensi	211
Gambar 5. 46 <i>Flowgraph</i> Fungsi Persetujuan Presensi	213
Gambar 5. 47 <i>Flowchart</i> fungsi tambah, <i>edit</i> , dan hapus data karyawan	222
Gambar 5. 48 <i>Flowgraph</i> Fungsi Tambah Karyawan	224
Gambar 5. 49 <i>Flowchart</i> Fungsi <i>Edit</i> Karyawan	233
Gambar 5. 50 <i>Flowgraph</i> Fungsi <i>Edit</i> Karyawan	234
Gambar 5. 51 <i>Flowchart</i> Fungsi Hapus Data Karyawan	240
Gambar 5. 52 <i>Flowgraph</i> Fungsi Hapus Data Karyawan	242



DAFTAR LAMPIRAN

Lampiran 1 Surat Permohonan Penelitian.....	254
Lampiran 2 Surat Balasan Permohonan Penelitian	255
Lampiran 3 Dokumentasi Wawancara Dan Observasi	256
Lampiran 4 Daftar Pertanyaan Wawancara.....	257



DAFTAR SINGKATAN

Singkatan	Kepanjangan
PT	: Perseroan Terbatas
<i>HR</i>	: <i>Human Resources</i>
<i>HRD</i>	: <i>Human Resources Development</i>
<i>QR Code</i>	: <i>Quick Response Code</i>
PT. KA	: PT Kendi Arindo
PP	: Peraturan Pemerintah
KHLI	: Klasifikasi Baku Lapangan Usaha Indonesia
APL	: Area Peruntukan Lain
<i>ERP</i>	: <i>Enterprise Resource Planning</i>
<i>JSON</i>	: <i>Javascript Object Notation</i>
IDE	: <i>Integrated Development Environment</i>
AMDAL	: Analisis Mengenai Dampak Lingkungan Hidup
<i>TC</i>	: <i>Test case</i>



BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era teknologi informasi saat ini, penerapan teknologi informasi menjadi aspek penting dalam meningkatkan efektivitas dan efisiensi kegiatan operasional perusahaan. Sistem informasi berkontribusi secara signifikan dalam mengotomatisasi tugas yang berulang, mengurangi kesalahan manusia, dan meningkatkan kolaborasi antar departemen. Sistem informasi adalah suatu sistem yang dirancang untuk mengumpulkan, mengelola, menyimpan, dan mendistribusikan informasi guna mendukung pengambilan keputusan, koordinasi, kontrol, analisis, dan visualisasi dalam suatu organisasi (Permana et al., 2024). Salah satu komponen penting dalam manajemen sumber daya manusia adalah pengelolaan data karyawan dan sistem presensi, karena kedua aspek ini berperan langsung terhadap kedisiplinan, produktivitas, dan efisiensi kerja dalam sebuah organisasi.

PT Kendi Arindo merupakan perusahaan yang bergerak di bidang perkebunan karet yang memiliki sejumlah karyawan dengan jadwal kerja yang cukup dinamis. Dalam kegiatan operasionalnya, perusahaan saat ini masih menggunakan sistem presensi berbasis *fingerprint* untuk mencatat kehadiran karyawan. Metode sidik jari (*fingerprint*) adalah sistem absensi berbasis biometrik yang telah dikenal luas di dunia kerja dan memanfaatkan keunikan sidik jari setiap individu untuk mencatat kehadiran secara otomatis (Fitriyana & Clara Dewanti, 2025). Meskipun telah menggunakan teknologi digital, sistem *fingerprint* yang diterapkan masih memiliki keterbatasan.

Secara faktual, sistem yang berjalan belum mampu menyajikan laporan kehadiran secara detail dan *real-time*. Informasi seperti jumlah kehadiran bulanan, data keterlambatan, cuti, izin, maupun hari libur (*off*) belum tersedia secara otomatis dalam satu laporan terintegrasi. Selain itu, sistem *fingerprint* masih memiliki potensi terjadinya kecurangan, seperti praktik titip jari, serta belum terintegrasi dengan data profil karyawan secara menyeluruh seperti jabatan, divisi, dan status kepegawaian. Kondisi ini menyebabkan manajemen mengalami

kesulitan dalam memperoleh informasi presensi yang komprehensif untuk mendukung proses evaluasi kinerja dan pengambilan keputusan.

Selain itu, ketergantungan pada sistem presensi berbasis perangkat fisik juga menimbulkan keterbatasan fleksibilitas dalam pemantauan kehadiran karyawan. Sistem yang hanya dapat diakses pada lokasi tertentu menyulitkan manajemen dalam melakukan pengawasan secara cepat dan menyeluruh, terutama ketika pimpinan membutuhkan informasi presensi secara mendadak untuk keperluan evaluasi maupun pengambilan keputusan. Kondisi ini menunjukkan bahwa sistem presensi *modern* tidak hanya dituntut mampu mencatat kehadiran, tetapi juga menyediakan akses informasi yang cepat, fleksibel, dan dapat diakses dari berbagai lokasi.

Berdasarkan kondisi tersebut, dapat diidentifikasi adanya kesenjangan antara sistem presensi yang berjalan dengan kebutuhan perusahaan akan sistem yang lebih *modern*, terintegrasi, dan mampu menyajikan data secara *real-time*. Permasalahan utama yang dihadapi adalah belum tersedianya sistem informasi karyawan dan presensi yang terintegrasi dalam satu *platform* digital yang mampu memberikan laporan kehadiran secara lengkap, akurat, dan mudah diakses oleh manajemen.

Beberapa penelitian sebelumnya telah mengembangkan sistem presensi berbasis *QR Code* maupun sistem informasi kepegawaian secara terpisah. Namun, sebagian besar penelitian tersebut masih berfokus pada pencatatan kehadiran saja dan belum mengintegrasikan data presensi dengan sistem informasi karyawan secara komprehensif dalam satu aplikasi berbasis *android* yang mampu menyajikan laporan *real-time* dan terstruktur. Hal ini menunjukkan adanya gap riset dalam pengembangan sistem presensi yang tidak hanya berfungsi sebagai alat pencatat kehadiran, tetapi juga sebagai sistem pendukung keputusan dalam manajemen sumber daya manusia.

Perkembangan teknologi *mobile* dan komputasi berbasis *cloud* telah membuka peluang bagi organisasi untuk mengembangkan sistem presensi yang lebih fleksibel dan terintegrasi. Pemanfaatan perangkat *smartphone* yang telah digunakan secara luas oleh karyawan memungkinkan proses pencatatan kehadiran dilakukan secara lebih praktis dan efisien tanpa bergantung pada perangkat presensi statis. Integrasi teknologi *mobile* dengan basis data *real-time* memungkinkan

penyimpanan, pengolahan, dan penyajian informasi kehadiran dilakukan secara otomatis dan terpusat, sehingga meningkatkan akurasi data serta mempercepat proses pelaporan.

Sebagai solusi atas permasalahan tersebut, penelitian ini mengusulkan digitalisasi sistem informasi karyawan dan presensi menggunakan teknologi *QR Code* berbasis aplikasi *android*. *QR Code* adalah *image* berupa matriks dua dimensi yang memiliki kemampuan untuk menyimpan data di dalamnya (Gunawan et al., 2023). Melalui sistem ini, setiap karyawan akan memiliki kode unik yang digunakan untuk melakukan presensi melalui pemindaian menggunakan perangkat *smartphone*. Data hasil pemindaian akan tersimpan secara otomatis ke dalam basis data dan dapat diakses secara *real-time* oleh pihak *HR (Human Resource)* maupun pimpinan perusahaan.

Kebaruan (*novelty*) dalam penelitian ini terletak pada pengembangan sistem presensi berbasis *QR Code* yang terintegrasi langsung dengan sistem informasi karyawan dalam satu aplikasi *android*, sehingga tidak hanya berfungsi sebagai alat pencatat kehadiran, tetapi juga sebagai *platform* pengelolaan data karyawan yang lengkap dan terstruktur. Sistem ini mampu menyajikan laporan presensi secara komprehensif meliputi jumlah kehadiran, keterlambatan, cuti, izin, dan hari libur (*off*) dalam periode tertentu, serta terhubung langsung dengan data profil karyawan.

Penerapan sistem ini menjadi urgensi bagi PT Kendi Arindo dalam mendukung transformasi digital perusahaan, meningkatkan efisiensi administrasi kepegawaian, mempercepat penyusunan laporan presensi, serta meningkatkan transparansi dan akuntabilitas dalam pengelolaan sumber daya manusia. Dengan sistem yang terintegrasi dan *real-time*, proses pemantauan dan evaluasi kehadiran karyawan dapat dilakukan secara lebih efektif dan akurat, sehingga mendukung peningkatan kinerja organisasi secara keseluruhan.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

- 1 Bagaimana merancang sistem informasi karyawan berbasis aplikasi *android* pada PT Kendi Arindo yang mampu mengelola serta menampilkan informasi karyawan secara detail, terintegrasi, dan mudah diakses oleh pihak perusahaan?
- 2 Bagaimana menggantikan sistem presensi *fingerprint* yang masih memiliki banyak kelemahan dengan sistem presensi berbasis pemindaian *QR Code* pada aplikasi *android* agar proses kehadiran menjadi lebih akurat, aman, dan efisien?
- 3 Bagaimana menyajikan laporan presensi dan informasi detail karyawan secara *real-time* melalui aplikasi *android* sehingga dapat diakses dengan mudah oleh pimpinan perusahaan untuk keperluan *monitoring* dan pengambilan keputusan?

1.3 Batasan Masalah

Agar penelitian lebih terarah, maka batasan penelitian yang ditetapkan adalah sebagai berikut:

1. Hasil akhir penelitian berupa sebuah aplikasi *android* sistem informasi karyawan dan presensi berbasis *QR Code* yang dikembangkan khusus untuk kebutuhan internal PT Kendi Arindo Empat Lawang.
2. Aplikasi dikembangkan berbasis *android* dengan bahasa pemrograman *dart* dan *framework flutter*.
3. Fitur yang dihasilkan dalam aplikasi meliputi:
 1. Pengelolaan data dan informasi detail karyawan.
 2. Presensi karyawan menggunakan pemindaian *QR Code*.
 3. *Validasi* presensi melalui foto/*selfie* karyawan.
 4. Pencatatan lokasi (*GPS*) pada saat pengambilan foto presensi.
 5. Penyimpanan data presensi secara *real-time*.
 6. Penyajian laporan presensi dan informasi karyawan yang dapat diakses pimpinan secara *real-time*.

4. Penelitian ini difokuskan pada lingkungan internal PT Kendi Arindo Empat Lawang dan tidak mencakup akses eksternal pihak ketiga.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Untuk merancang sistem informasi karyawan berbasis aplikasi *android* pada PT Kendi Arindo Empat Lawang yang mampu mengelola dan menampilkan data karyawan secara detail, terintegrasi, dan terpusat, sehingga mempermudah proses administrasi kepegawaian secara digital.
2. Untuk menerapkan sistem presensi berbasis pemindaian *QR Code* pada aplikasi *android* sebagai pengganti sistem *fingerprint*, guna meningkatkan akurasi dan keamanan pencatatan kehadiran serta mengurangi potensi kecurangan seperti titip absen.
3. Untuk menyediakan laporan presensi dan informasi karyawan secara *real-time* melalui aplikasi *android* yang dapat diakses oleh pimpinan perusahaan, sehingga mendukung proses *monitoring*, evaluasi kehadiran, dan pengambilan keputusan manajemen.

1.5 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat sebagai berikut:

1. Manfaat Bagi Perusahaan
 - a. Mempermudah proses presensi dan pengelolaan data karyawan secara digital berbasis *QR Code*.
 - b. Meningkatkan efisiensi dan akurasi dalam pencatatan kehadiran dibandingkan sistem *fingerprint* sebelumnya.
 - c. Menyediakan laporan kehadiran yang lebih detail, *real-time*, dan mudah dipahami oleh pimpinan.
 - d. Meningkatkan transparansi dan akuntabilitas dalam sistem manajemen sumber daya manusia.
2. Manfaat Teoritis
 - a. Menambah referensi ilmiah dalam bidang sistem informasi, khususnya digitalisasi presensi dan pengelolaan data karyawan.

- b. Mengembangkan konsep penerapan teknologi *QR Code* dan aplikasi *mobile* dalam manajemen sumber daya manusia.
 - c. Menjadi landasan akademik bagi penelitian selanjutnya yang berkaitan dengan sistem presensi digital.
3. Manfaat Praktis
- a. Memberikan solusi sistem presensi digital yang dapat diterapkan pada lingkungan kerja.
 - b. Membantu organisasi dalam meningkatkan efisiensi administrasi dan akurasi data kehadiran.
 - c. Mendukung pengambilan keputusan berbasis data melalui laporan presensi yang *real-time* dan terintegrasi.
4. Manfaat Bagi Akademisi/Peneliti Selanjutnya
- a. Dapat menjadi referensi bagi penelitian sejenis yang berfokus pada pengembangan sistem informasi presensi digital.
 - b. Memberikan dasar untuk penelitian lanjutan mengenai implementasi teknologi *android* dan *QR Code* dalam bidang manajemen sumber daya manusia.

1.6 Sistematika Penulisan

Penulisan skripsi ini disusun agar memberikan gambaran yang jelas mengenai tahapan penelitian, sebagai berikut:

BAB I PENDAHULUAN

Menjelaskan latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Berisi teori-teori pendukung yang relevan dengan penelitian, seperti konsep sistem informasi karyawan, sistem presensi, teknologi *QR Code*, sistem operasi *android*, flutter dan dart, *firebase* sebagai basis data, serta penelitian terdahulu yang berkaitan dengan pengembangan sistem informasi karyawan dan presensi berbasis aplikasi *android*.

BAB III METODE PENELITIAN

Berisi metode penelitian yang digunakan, pendekatan pengembangan sistem (*design thinking*), tahapan penelitian, teknik pengumpulan data, serta alat dan bahan yang diperlukan untuk membangun aplikasi sistem informasi karyawan dan presensi berbasis *android* menggunakan *QR Code*.

BAB IV ANALISIS DAN PERANCANGAN

Bab ini membahas proses analisis dan perancangan aplikasi sistem informasi karyawan dan presensi berbasis *android* menggunakan *QR Code*. Tahapan analisis dilakukan untuk mengidentifikasi permasalahan serta kebutuhan sistem yang berkaitan dengan pengelolaan data karyawan dan pencatatan presensi secara digital. Hasil analisis tersebut digunakan sebagai dasar dalam perancangan sistem, meliputi perancangan alur proses, serta perancangan antarmuka aplikasi yang dikembangkan untuk mendukung pengelolaan data karyawan dan presensi secara efektif, efisien, dan *real-time*.

BAB V HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil implementasi dan pengujian dari aplikasi sistem informasi karyawan dan presensi berbasis *android* menggunakan *QR Code* yang telah dirancang. Implementasi meliputi pembangunan aplikasi menggunakan flutter dan dart serta integrasi dengan *firebase* sebagai basis data. Pengujian dilakukan untuk memastikan bahwa setiap fitur, seperti pengelolaan data karyawan, pembuatan dan pemindaian *QR Code*, serta penyajian laporan presensi, berjalan sesuai dengan kebutuhan yang telah ditetapkan. Hasil pengujian dianalisis untuk mengetahui tingkat keberhasilan sistem dalam meningkatkan efektivitas dan efisiensi pengelolaan presensi karyawan.

BAB VI KESIMPULAN DAN SARAN

Bab ini memuat kesimpulan berdasarkan hasil implementasi dan pengujian aplikasi sistem informasi karyawan dan presensi berbasis *android* menggunakan *QR Code*. Selain itu, bab ini juga menyajikan saran yang dapat digunakan sebagai bahan evaluasi dan pengembangan sistem di masa yang akan datang agar menjadi lebih optimal dan sesuai dengan kebutuhan perusahaan.

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Digitalisasi

Digitalisasi adalah proses mengubah data analog menjadi digital dengan menggunakan teknologi dan mengoperasikannya secara otomatis melalui sistem komputerisasi (Bainar, 2024). Digitalisasi dilakukan untuk membuat arsip dokumen berbentuk digital. Digitalisasi memerlukan peralatan seperti komputer, *scanner*, operator media sumber dan *software* pendukung. Digitalisasi merupakan proses alih media dari cetak atau analog ke dalam media digital atau elektronik melalui proses *scanning*, digital *photograph* atau teknik lainnya (Hidayatullah, 2023).

Digitalisasi melibatkan perubahan media dalam bentuk cetak, audio, atau video menjadi *format* digital (Hutagaol et al., 2024). Proses ini memerlukan perangkat seperti komputer, pemindai (*scanner*), operator media sumber, dan perangkat lunak pendukung. Digitalisasi merujuk pada peningkatan ketersediaan data digital yang memungkinkan oleh kemajuan dalam penciptaan, pengiriman, penyimpanan, dan analisis data digital, serta memiliki potensi untuk menyusun, membentuk, dan memengaruhi dunia masa kini.

Istilah digitalisasi mengacu pada penggunaan teknologi dan data digital untuk meningkatkan bisnis, pendapatan, dan menciptakan budaya digital (Yunaningsih et al., 2021). Berdasarkan pendapat para ahli tersebut, digitalisasi dapat dipahami sebagai suatu proses alih media dari bentuk analog ke bentuk digital yang didukung oleh perangkat keras dan perangkat lunak, sekaligus sebagai pemanfaatan teknologi digital untuk mengotomatisasi dan meningkatkan kualitas pengelolaan data serta proses kerja sehingga menjadi lebih efisien, akurat, mudah diakses, dan memberikan nilai tambah bagi organisasi maupun penggunanya.

2.2 Pengertian Sistem Informasi Karyawan

Sistem informasi karyawan merupakan sistem yang digunakan untuk mengelola dan menyajikan data sumber daya manusia dalam suatu organisasi agar informasi terkait karyawan dapat diperoleh secara cepat dan akurat. sistem

informasi dapat diartikan sebagai sebuah sistem dalam organisasi yang menggabungkan elemen-elemen seperti orang, fasilitas, teknologi, media, prosedur, dan pengendalian, yang dirancang untuk memastikan saluran komunikasi yang efektif, memproses transaksi rutin tertentu, memberikan informasi penting kepada manajemen terkait peristiwa internal dan eksternal, serta menyediakan dasar informasi yang diperlukan untuk pengambilan keputusan yang tepat (Nuryana et al., 2024).

Sistem informasi adalah kerangka kerja yang mengkoordinir sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (*output*) berupa informasi guna mencapai sasaran (Aidah et al., 2024). Pada dasarnya pengertian karyawan dipersamakan dengan pengertian buruh, tenaga kerja/pekerja atau diistilahkan juga dengan sebutan sumber daya manusia (SDM) (Rizaldhi & Fadli, 2023). Dalam artian secara makro, SDM (Sumber Daya Manusia) meliputi semua manusia sebagai penduduk atau warga negara suatu negara atau dalam batas wilayah tertentu yang sudah memasuki usia angkatan kerja, baik yang sudah memasuki usia angkatan kerja, maupun yang sudah mampu memperoleh pekerjaan.

Berdasarkan pendapat para ahli tersebut, sistem informasi karyawan dapat dipahami sebagai suatu sistem terintegrasi yang digunakan untuk mengelola data personal, jabatan, divisi, serta status kerja karyawan secara digital guna mendukung kegiatan administrasi dan pengambilan keputusan dalam perusahaan. Sistem ini tidak hanya berfungsi sebagai media penyimpanan data, tetapi juga sebagai alat pendukung analisis dan pengawasan kinerja sumber daya manusia, sehingga pengelolaan data menjadi lebih tertata, akurat, dan mudah diakses, serta membantu meningkatkan efisiensi operasional dan transparansi informasi dalam organisasi.

2.3 Pengertian Presensi

Sistem presensi adalah metode atau alat yang digunakan untuk mencatat, memantau, dan mengelola kehadiran individu, baik di lingkungan kerja, sekolah, maupun organisasi lainnya. Fungsi utama dari sistem presensi adalah memastikan akurasi dan efisiensi dalam pelacakan kehadiran, yang bisa berpengaruh pada pengelolaan absensi, penggajian, hingga evaluasi kinerja. Dengan teknologi

modern, sistem presensi juga dapat memberikan data analitik untuk membantu pengambilan keputusan (Manoppo et al., 2025).

Secara arti kamus presensi adalah kehadiran, yang berarti presensi adalah kehadiran seseorang pada suatu tempat yang menyatakan bahwa kehadiran seseorang tersebut diharapkan oleh orang-orang tertentu untuk suatu tujuan (Erzed et al., 2022). Sistem presensi adalah sebuah sistem pendataan kehadiran, bagian pelaporan aktivitas sebuah institusi, atau komponen institusi yang berisi data kehadiran yang disusun sedemikian rupa sehingga memudahkan pencarian data ketika data tersebut akan dipergunakan oleh institusi yang bersangkutan (Wijaya et al., 2023).

Berdasarkan pendapat para ahli tersebut, sistem presensi dapat dipahami sebagai seperangkat prosedur dan teknologi yang digunakan untuk mencatat dan mengelola kehadiran individu secara sistematis, akurat, dan terdokumentasi. Sistem ini tidak hanya berfungsi sebagai alat pencatat hadir atau tidak hadir, tetapi juga sebagai bagian dari sistem informasi yang menyediakan data kehadiran untuk mendukung proses administrasi, seperti penggajian, penilaian kinerja, serta pengambilan keputusan manajerial secara lebih objektif dan efisien.

2.4 Pengertian *QR Code*

QR Code (Quick Response Code) adalah perkembangan dari *barCode* atau kode batang yang hanya mampu menyimpan informasi secara horizontal, sedangkan *QR Code* mampu menyimpan informasi lebih banyak baik secara horizontal maupun vertikal (Syam & Erdisna, 2022). *QR Code* telah banyak digunakan karena fitur-fiturnya yang baik seperti data yang berkapasitas besar, memindai kecepatan tinggi, dan mencetak ukuran kecil. Kenaikan jumlah *smartphone* adalah alasan di balik popularitas *QR Code*. *Smartphone* yang mampu *decoding* dan mengakses sumber daya *online* serta memiliki penyimpanan yang berkapasitas tinggi dan kecepatan tinggi *decoding*.

QR Code atau *Quick Response Code* adalah kode batang dua dimensi yang dapat dipindai menggunakan ponsel cerdas atau perangkat seluler lainnya yang dilengkapi kamera dan perangkat lunak pembaca *QR Code*. *QR Code* dapat menyimpan berbagai informasi, seperti *URL (Uniform Resource Locator)* situs

web, informasi kontak, dan detail produk, serta dapat digunakan dalam berbagai aplikasi (Kadek Merita et al., 2024). *Code QR* adalah simbol bergaya matriks yang terdiri dari titik-titik hitam putih yang disusun dalam sebuah kotak, yang setiap elemennya memiliki arti tersendiri. *Code QR* dapat menyimpan kurang lebih 7.000 karakter dan memindai 10 kali lebih cepat dibandingkan jenis *barCode* lainnya (Fitriani & Puspitasari, 2024).

Berdasarkan pendapat para ahli tersebut, *QR Code* dapat dipahami sebagai media pengkodean data dua dimensi yang mampu menyimpan informasi dalam jumlah besar, dapat dipindai dengan cepat menggunakan perangkat digital, serta memiliki kemampuan koreksi kesalahan. Teknologi ini sangat sesuai digunakan dalam berbagai sistem informasi *modern* yang menuntut kecepatan akses, keakuratan data, dan kemudahan integrasi dengan perangkat berbasis digital.

2.5 Pengertian *Android*

Android adalah sebuah sistem operasi untuk *smartphone* dan tablet. Sistem operasi dapat diilustrasikan sebagai jembatan antara peranti (*device*) dan penggunanya, sehingga pengguna dapat berinteraksi dengan *device* nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device* (Edriati et al., 2021). *Android* adalah sistem operasi dengan basis linux telepon seluler dan menyediakan *platform* terbuka untuk para pengembang dalam menciptakan aplikasi sendiri sehingga dapat digunakan oleh berbagai macam piranti bergerak (Dalimunthe & Simanjuntak, 2023).

Android merupakan sebuah sistem operasi telepon seluler dan komputer tablet layar sentuh yang berbasis linux, dengan kelebihan sistem operasi *android* akan membantu dan mengembangkan pada keragaman menu aplikasi (Rohmat & Gusti, 2023). *Android* merupakan salah satu sistem operasi berbasis *mobile* yang sangat banyak di gunakan sekarang ini. Pengertian *android* adalah sistem operasi yang dirancang oleh Google dengan basis kernel Linux untuk mendukung kinerja perangkat elektronik layar sentuh, seperti tablet atau *smartphone* (Saputra, 221 C.E.). Jadi, *android* digunakan dengan sentuhan, gesekan ataupun ketukan pada layar *gadget* anda.

Android bersifat *open source* atau bebas digunakan, dimodifikasi, diperbaiki dan didistribusikan oleh para pembuat ataupun pengembang perangkat lunak. Dengan sifat *open source* perusahaan teknologi bebas menggunakan *OS (Operating System)* ini diperangkatnya tanpa lisensi alias gratis. Berdasarkan pendapat para ahli tersebut, *android* dapat dipahami sebagai sistem operasi berbasis Linux yang bersifat terbuka dan dioptimalkan untuk perangkat layar sentuh, menyediakan *platform* yang fleksibel bagi pengembang, serta memberikan kemudahan, kecepatan, dan kenyamanan bagi pengguna dalam mengakses berbagai aplikasi dan layanan digital.

2.6 Pengertian *Visual Studio Code*

Visual Studio Code (VS Code) ini adalah sebuah teks *editor* ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows (Aldi & Aji, 2025). *Visual studio Code* merupakan salah satu *integrated development environment (IDE)* ringan yang digunakan untuk membantu proses penulisan, pengujian, dan pengelolaan kode program. *IDE* ini dikembangkan oleh Microsoft dan mendukung berbagai bahasa pemrograman serta *framework modern*. *Visual Studio Code* menyediakan beragam fitur seperti *syntax highlighting*, *Code completion*, *debugger* terintegrasi, *terminal* bawaan, serta dukungan ekstensi yang sangat luas.

Dalam konteks pengembangan aplikasi *mobile* berbasis *android* dengan flutter, *visual studio Code* berperan sebagai lingkungan kerja utama bagi pengembang untuk menulis dan mengelola kode dart, menjalankan *debugging*, serta mengintegrasikan proyek dengan *emulator* maupun perangkat fisik. Tampilan antarmukanya yang sederhana, kinerja yang ringan, dan kemudahan instalasi berbagai ekstensi (Seperti Flutter dan Dart). *IDE* ini memiliki fitur-fitur yang sangat lengkap dan memudahkan pengembang dalam membuat aplikasi *mobile* (Gunawan, 2024). Menjadikan *visual studio Code* sebagai salah satu *IDE* yang banyak digunakan dalam pengembangan aplikasi berbasis *mobile*.

Salah satu keunggulan lain dari *visual studio Code* adalah kemampuannya dalam mendukung penambahan ekstensi sesuai kebutuhan pengembang. Fitur lainnya adalah kemampuan untuk menambah ekstensi dimana para pengembang

dapat menambah ekstensi untuk menambah fitur yang tidak ada di *visual studio Code* (Firnando et al., 2023). Fitur ini memungkinkan pengembang menambahkan berbagai ekstensi untuk menambah fungsi yang belum tersedia secara bawaan di *visual studio Code*, seperti integrasi *framework* tertentu, alat bantu analisis kode, manajemen basis data, maupun *tools* pendukung pengembangan aplikasi *mobile*.

Berdasarkan pendapat para ahli tersebut, *visual studio Code* dapat dipahami sebagai sebuah *integrated development environment (IDE)* yang ringan dan *multiplatform*, dikembangkan oleh Microsoft dengan dukungan berbagai bahasa pemrograman dan *framework modern*. *Visual studio Code* tidak hanya menyediakan fitur dasar seperti penulisan dan pengelolaan kode, tetapi juga dilengkapi kemampuan *debugging*, terminal terintegrasi, serta pengaya (ekstensi) yang dapat disesuaikan dengan kebutuhan pengembang.

2.7 Pengertian *Flutter*

Flutter adalah sebuah *framework open-source* yang dikembangkan oleh Google untuk membangun antar muka (*user interface*) aplikasi *android* dan *iOS*. Jika kita berbicara bagaimana cara membuat aplikasi *android* maupun *iOS*, biasanya akan dihadapi dengan banyak pilihan kenapa aplikasi tersebut dapat dibangun. Selain itu yang membedakan *flutter* dengan *framework* lainnya adalah *flutter* merupakan *framework multiplatform* yang dikembangkan oleh tim dari Google (Irfankha, 2025). Banyak perusahaan besar di dunia telah mengadopsi *flutter* untuk mengembangkan aplikasi. Sebut saja Google, Alibaba Group, dan Grab.

Flutter adalah kerangka pengembangan aplikasi *mobile open-source* yang dikembangkan oleh Google. Dengan menggunakan *Flutter*, pengembang dapat membangun aplikasi yang memiliki antarmuka pengguna menarik, *responsive*, dan dapat berjalan di berbagai *platform* (Putra & Kurniawan, 2023). *Flutter* adalah *SDK (Software Development Kit)* untuk pengembangan aplikasi *mobile* yang dikembangkan oleh Google untuk membangun antarmuka (*user interface*) aplikasi *android* dan *iOS*. Rilis pada juni 2018 sama seperti *react native*, *framework* ini dapat digunakan untuk membuat atau mengembangkan aplikasi *mobile* yang dapat berjalan pada *device iOS* dan *android* (Sudradjat, 2021).

Berdasarkan pendapat para ahli tersebut, Flutter dapat dipahami sebagai sebuah *framework* sekaligus *SDK* (*Software Development Kit*) berbasis *open-source* yang dikembangkan oleh Google untuk membangun antarmuka aplikasi *mobile* yang menarik, responsif, dan *modern* pada berbagai *platform*, khususnya *Android* dan *iOS*. Flutter mendukung pengembangan aplikasi *cross-platform* dengan satu basis kode, sehingga proses pengembangan menjadi lebih efisien, hemat waktu, dan mudah dikelola. Selain itu, Flutter menawarkan kemampuan untuk menghasilkan tampilan antarmuka yang kaya dan dinamis, performa tinggi, serta pengalaman pengguna yang konsisten di berbagai perangkat.

2.8 Pengertian Dart Programming Language

Dart adalah sebuah bahasa pemrograman yang dikembangkan oleh Google, dirancang oleh Lars Bak dan Kasper Lund (Sahputra et al., 2023). Bahasa Dart merupakan *programming language* lintas *platform* atau *platform* independen yang artinya dapat dijalankan pada sistem operasi yang berbeda seperti Windows, Linux, Unix dan MacOS, dll yang awalnya dikembangkan oleh Google dan kemudian disetujui sebagai standar oleh Ecma, yang saat ini digunakan untuk membangun aplikasi web, server, *desktop*, dan seluler (Raisa et al., 2024).

Dart adalah bahasa pemrograman terstruktur *open source* untuk membuat aplikasi *web* berbasis *browser* yang kompleks. Pengguna dapat menjalankan aplikasi yang dibuat di dart baik dengan menggunakan *browser* yang secara langsung mendukung kode dart atau dengan mengkompilasi kode dart pengguna ke javascript. Salah satu tujuan utama dart adalah agar bahasa itu tampak akrab (Nelly Sofi & Riza Dharmawan, 2022).

Berdasarkan pendapat para ahli tersebut, Dart dapat dipahami sebagai bahasa pemrograman terstruktur dan *open source* yang bersifat lintas *platform*, sehingga dapat dijalankan pada berbagai sistem operasi seperti Windows, Linux, Unix, dan MacOS. Dart digunakan untuk membangun beragam jenis aplikasi, mulai dari aplikasi *mobile*, web, desktop, *IoT* (*Internet of Things*), *backend* (*CLI*), hingga *game*. Selain itu, Dart dirancang sebagai bahasa *modern* yang dapat dikompilasi ke javascript dan mendukung pengembangan antarmuka yang responsif, sehingga

banyak dimanfaatkan dalam *framework* seperti Flutter untuk menghasilkan aplikasi dengan performa tinggi dan mudah dikelola.

2.9 Pengertian *Firebase*

Firebase sendiri merupakan suatu layanan yang disediakan oleh google untuk dapat memberikan kemudahan bahkan dapat mempermudah para pengembang sebuah aplikasi dalam mengembangkan aplikasinya (Khaledi et al., 2022). Dengan menggunakan *firebase*, pihak pengembang dapat fokus dalam hal mengembangkan aplikasi saja tanpa harus memberikan *effort* yang besar untuk urusan *backend* (sesuatu yang berkaitan dengan sebuah penyimpanan secara *cloud*) yang merupakan solusi yang ditawarkan oleh Google untuk mempercepat pekerjaan seseorang *developer*.

Firebase menyediakan layanan seperti *realtime database* dan *cloud firestore* yang memungkinkan data disimpan, diubah, dan diambil secara *online* melalui koneksi internet tanpa perlu membangun server dan *database* secara mandiri. Sebagai basis data berbasis *cloud*, *firebase* memungkinkan proses pengelolaan data dilakukan secara *real-time*. *firebase realtime database* merupakan solusi yang ditawarkan oleh Google untuk mempermudah pengembangan aplikasi *mobile* dan web, serta memiliki kemampuan untuk mengakses *database* secara *realtime* (Riswanto & Fahrudin, 2024).

Didalam *firebase* terdapat sebuah sarana yang bernama *firebase realtime database* yang berfungsi sebagai sarana untuk menyimpan data dan pembacaan data yang dilakukan lebih cepat dan lebih aman (M. I. Z. Hasibuan & Triase, 2022). *Database* diberikan secara *realtime* dan disimpan didalam *format JSON (Javascript Object Notation)* yang akan dilakukan penyinkronan secara terus menerus. *Firebase realtime database* merupakan *database* yang dapat memperbaharui secara otomatis data yang telah dibaca ketika terjadi perubahan data yang terjadi pada *database*, sehingga tidak perlu lagi melakukan pemanggilan berulang agar mendapatkan data terbaru yang ada didalam *database*.

Berdasarkan pendapat para ahli tersebut, *Firebase* dapat dipahami sebagai layanan basis data berbasis *cloud* yang disediakan oleh Google untuk mempermudah pengembang dalam menangani kebutuhan *backend* aplikasi.

Firebase mendukung proses penyimpanan, pengelolaan, sinkronisasi, dan pembaruan data secara *real-time* dengan *format* terstruktur dan aman, sehingga sangat sesuai digunakan pada aplikasi *mobile* maupun web yang membutuhkan akses data cepat, dinamis, dan terintegrasi.

2.10 Penelitian Terdahulu

Sebagai bahan acuan bagi peneliti dalam melakukan penelitian, penulisan merujuk pada penelitian-penelitian yang pernah ada dalam bentuk jurnal-jurnal yang akan dijelaskan pada tabel 2.1.

Tabel 2. 1 Penelitian Terdahulu

No	Peneliti & Tahun	Judul Penelitian	Rangkuman
1	(Qois & Jumaryadi, 2021)	Penerapan <i>Database</i> MySQL dalam Sistem Kepegawaian	Penelitian ini mengembangkan sistem kehadiran pegawai dengan memanfaatkan <i>Location Based Service (LBS)</i> dan <i>GPS</i> . Aplikasi memungkinkan absensi dari rumah (<i>WFH</i>), tugas luar kota, atau lapangan. <i>Validasi</i> absensi dilakukan melalui foto <i>selfie</i> dan pelacakan lokasi otomatis. Metode pengembangan yang digunakan adalah <i>Waterfall</i> , dan pengujian dilakukan dengan <i>black box</i> serta <i>usability testing</i> . Sistem berhasil memberikan laporan kehadiran <i>real-time</i> lengkap dengan koordinat lokasi.
2	(Pratama et al., 2023)	Pendekatan Metode <i>Prototype</i> Pada Aplikasi Presensi Berbasis <i>Mobile</i> (Studi Kasus: Kantor Desa Mekar Jaya)	Penelitian ini membuat aplikasi presensi pegawai berbasis <i>mobile</i> menggunakan metode <i>Prototype</i> . Masalah awal yaitu pengelolaan absensi manual yang berpotensi terjadi kesalahan dan lambat dalam rekap. Sistem dikembangkan dengan berbagai diagram <i>UML</i> (<i>use case</i> , <i>activity</i> , <i>sequence</i> , <i>class</i>) sebagai rancangan. Aplikasi yang dihasilkan memiliki fitur absensi masuk, izin, riwayat absensi, dan keamanan akses <i>login</i> pegawai. Metode <i>prototype</i> mempermudah perbaikan cepat sesuai kebutuhan pengguna
3	(Solihin & Adli, 2023)	Sistem Absensi Mahasiswa Menggunakan <i>QR Code</i> dengan Metode <i>ZXing</i> Berbasis <i>Android</i>	Penelitian ini membuat sistem absensi mahasiswa berbasis <i>Android</i> menggunakan <i>QR Code</i> dan library <i>ZXing</i> . Sistem terhubung ke <i>server</i> melalui intranet untuk mencegah kecurangan titip absen. Hasilnya, proses absensi menjadi lebih cepat, akurat, dan mampu mengurangi manipulasi kehadiran.

4	(Muhammad Ibnu Rosikhin & Farhanna Mar'i, 2024)	Pengembangan Sistem Presensi Guru UPT SMPN 34 Gresik Berbasis Web	Penelitian ini mengembangkan sistem presensi guru berbasis web untuk menggantikan pencatatan manual. Sistem dibuat dengan metode <i>Waterfall</i> dan dilengkapi fitur pengelolaan kehadiran serta perhitungan gaji. Hasilnya, sistem membantu meningkatkan efisiensi administrasi dan memastikan keakuratan data presensi.
5	(Lee et al., 2024)	Aplikasi Absensi <i>Cleaning Service</i> dengan Metode <i>MADLC</i>	Penelitian ini mengembangkan aplikasi absensi bagi staf <i>cleaning service</i> pada perusahaan distributor mesin diesel. Masalah utama adalah pencatatan manual yang tidak efisien. Peneliti menggunakan metode <i>Mobile Application Development Life Cycle (MADLC)</i> dengan tahapan analisis, desain, implementasi, prototipe, dan pengujian. Aplikasi yang dihasilkan memiliki fitur <i>scan QR</i> , <i>selfie</i> absensi, dan <i>monitoring</i> kehadiran secara <i>real-time</i> . Hasil pengujian menunjukkan aplikasi berjalan baik dan mempermudah pengelolaan absensi.
6	(Irmaliantie et al., 2024)	Sistem Informasi Presensi Berbasis <i>Android</i> Menggunakan <i>QR Code</i> Pada Kantor Desa Pandawan	Penelitian ini menghasilkan sistem informasi presensi berbasis <i>Android</i> yang memanfaatkan <i>QR Code</i> untuk mencatat kehadiran perangkat desa secara otomatis. Sistem yang dibangun mampu menyimpan data kehadiran ke dalam basis data, menampilkan riwayat presensi, serta membantu meminimalkan manipulasi data dan ketidaktepatan pencatatan presensi yang sebelumnya dilakukan secara manual.
7	(Shobur & Satrio, 2025)	Perancangan Sistem Informasi Absensi pada Sogogi Shabu & Grill Jambi Berbasis <i>Mobile</i>	Penelitian ini merancang aplikasi absensi <i>mobile</i> untuk restoran Sogogi Shabu & Grill Jambi yang masih menggunakan absensi foto manual via <i>WhatsApp</i> . Masalah utamanya ialah kesalahan rekap absensi yang berdampak pada perhitungan gaji. Peneliti menggunakan metode <i>Waterfall</i> serta teknologi pengenalan wajah <i>Eigenfaces</i> untuk memvalidasi absensi. Sistem dirancang menggunakan <i>UML (use case, activity, class diagram)</i> . Tujuan utama adalah mempermudah <i>admin</i> dalam rekap absensi dan mencegah kecurangan.
8	(Hafizhah & Hidayat, 2025)	Optimalisasi Pengembangan Sistem Presensi Karyawan Menggunakan <i>Extreme Programming</i> dan Teknologi <i>QR Code</i>	Penelitian ini mengembangkan sistem presensi karyawan berbasis <i>QR Code</i> dengan metode <i>Extreme Programming</i> sebagai pendekatan pengembangan perangkat lunak. Sistem yang dibangun mampu melakukan pencatatan kehadiran secara cepat dan terkomputerisasi, mengurangi potensi kecurangan, serta mempermudah proses rekapitulasi data presensi. Metode <i>Extreme Programming</i>

membantu pengembangan sistem menjadi lebih terstruktur, adaptif terhadap perubahan kebutuhan, dan menghasilkan aplikasi presensi yang sesuai dengan kebutuhan pengguna.

Sumber : data diolah oleh penulis, 2025

Berdasarkan Tabel 2.1 penelitian terdahulu, dapat diketahui bahwa penerapan sistem presensi digital telah banyak dikembangkan dengan berbagai pendekatan teknologi dan metode pengembangan sistem. Beberapa penelitian menunjukkan bahwa penggunaan teknologi *mobile*, *QR Code*, *GPS (Global Positioning System)*, maupun biometrik mampu meningkatkan kecepatan pencatatan kehadiran, mengurangi potensi kecurangan, serta menyediakan laporan kehadiran secara *real-time* dan akurat. Selain itu, pemanfaatan metode pengembangan seperti *waterfall*, *prototype*, *extreme programming* serta pendekatan *design thinking* menunjukkan bahwa proses perancangan sistem yang berpusat pada kebutuhan pengguna mampu menghasilkan aplikasi yang lebih efektif, mudah digunakan, dan sesuai dengan kondisi lapangan.

Hasil penelitian terdahulu juga menegaskan bahwa digitalisasi sistem presensi dan pengelolaan data kepegawaian mampu meningkatkan efisiensi administrasi, mempermudah proses rekapitulasi data, serta mendukung pengambilan keputusan berbasis informasi yang lebih akurat. Integrasi sistem berbasis *mobile* memungkinkan akses data yang lebih fleksibel dan transparan bagi pihak manajemen.

Dengan demikian, penelitian mengenai digitalisasi sistem informasi karyawan dan presensi menggunakan *QR Code* berbasis *android* memiliki relevansi yang kuat dengan penelitian sebelumnya, sekaligus memberikan kontribusi dalam pengembangan sistem yang lebih terintegrasi, *real-time*, dan mudah diakses guna mendukung efektivitas manajemen sumber daya manusia di PT Kendi Arindo.

2.11 Kerangka Berpikir Penelitian

Kerangka berpikir atau kerangka pemikiran adalah dasar pemikiran dari penelitian yang disintesis dari fakta-fakta, observasi dan kajian kepustakaan. Oleh karena itu, kerangka berpikir memuat teori, dalil atau konsep-konsep yang akan dijadikan dasar dalam penelitian. Di dalam kerangka pemikiran variabel-

variabel penelitian dijelaskan secara mendalam dan relevan dengan permasalahan yang teliti, sehingga dapat dijadikan dasar untuk menjawab permasalahan penelitian (Syahputri Addini Zahra, 2023).

Berikut Adalah kerangka berpikir pada penelitian ini dapat dilihat pada gambar 2.1.



Sumber : data diolah oleh penulis, 2026

Gambar 2. 1 Kerangka Berpikir Penelitian

Dari gambar 2.1 diatas dapat dijelaskan dan disimpulkan bahwa :

1. Identifikasi Masalah: PT Kendi Arindo masih menggunakan sistem absensi *fingerprint* yang memiliki beberapa kelemahan, seperti potensi kecurangan (*titip presensi*), kesalahan pencatatan, serta keterbatasan dalam menampilkan laporan kehadiran yang lengkap. Sistem tersebut juga belum mampu menyediakan data kehadiran secara *real-time* bagi pimpinan.
2. Analisis Kebutuhan Sistem: Diperlukan sistem informasi karyawan dan *presensi* berbasis aplikasi *android* yang mampu mencatat kehadiran

menggunakan *QR Code* secara otomatis, menyimpan data ke dalam basis data *real-time*, serta menyediakan laporan kehadiran lengkap yang dapat diakses oleh pimpinan kapan saja dan di mana saja.

3. Perancangan Sistem: Sistem dirancang menggunakan *framework* flutter dengan bahasa pemrograman dart dan basis data. Fitur utama meliputi manajemen data karyawan, proses presensi dengan *QR Code*, dan tampilan laporan kehadiran bulanan yang lebih lengkap seperti data hadir, izin, cuti, dan hari *off*.
4. Implementasi Sistem: Aplikasi dikembangkan menggunakan flutter dan diintegrasikan dengan basis data. Proses pemindaian *QR Code* dilakukan melalui kamera *smartphone*, dan hasil presensi langsung tersimpan serta dapat diakses secara *real-time* oleh bagian *HR* dan pimpinan perusahaan. Pengujian Sistem: Dilakukan uji fungsi dan kepuasan pengguna untuk memastikan aplikasi berjalan sesuai kebutuhan dan meningkatkan efisiensi kerja.
5. Pengujian dan Evaluasi: Sistem diuji menggunakan metode *White-Box Testing* untuk memastikan bahwa setiap alur logika program berjalan sesuai rancangan dan menghasilkan keluaran yang benar. Pengujian mencakup fungsi utama seperti proses pemindaian *QR Code*, penyimpanan data presensi ke dalam basis data, serta penampilan laporan kehadiran. Metode ini juga digunakan untuk menilai efisiensi algoritma, akurasi hasil, serta kestabilan sistem dalam berbagai kondisi penggunaan.
6. Hasil dan Kesimpulan: Aplikasi yang dikembangkan mampu meningkatkan efisiensi pengelolaan data karyawan dan presensi, mengurangi potensi kecurangan pada sistem *fingerprint*, serta menampilkan laporan kehadiran karyawan secara lengkap dan *real-time*. Sistem ini juga membantu pimpinan dalam pengawasan kehadiran.

BAB III

METODE PENELITIAN

3.1 Objek, Subjek dan Jadwal Penelitian

3.1.1 Objek Penelitian

Objek penelitian ini adalah PT Kendi Arindo, sebuah perusahaan yang bergerak di bidang perkebunan karet, yang berlokasi di Jalan Lingkar Talang Gunung, Kelurahan Jaya Loka, Kecamatan Tebing Tinggi, Kabupaten Empat Lawang, Provinsi Sumatera Selatan. Penelitian ini berfokus pada digitalisasi sistem informasi karyawan berbasis aplikasi *android* yang mencakup pengelolaan data karyawan dan sistem presensi menggunakan teknologi *QR Code*. Fokus penelitian diarahkan pada perancangan dan pengembangan aplikasi yang mampu mempermudah proses pencatatan kehadiran, penyimpanan data karyawan, serta penyajian laporan presensi secara *real-time* dan terintegrasi.

Pemilihan objek ini didasarkan pada kondisi di lapangan yang masih menggunakan sistem *fingerprint* dalam pencatatan kehadiran, yang dinilai kurang efisien karena berisiko terhadap kecurangan (titip absen), keterlambatan pelaporan, dan keterbatasan dalam menampilkan informasi kehadiran secara lengkap dan *real-time*. Melalui penerapan digitalisasi sistem informasi karyawan berbasis aplikasi *android*, diharapkan PT Kendi Arindo dapat meningkatkan efisiensi administrasi kepegawaian, akurasi data presensi, serta memberikan kemudahan bagi pimpinan perusahaan dalam memantau kehadiran dan aktivitas karyawan dengan laporan yang lebih komprehensif dan transparan, *real-time*, dan mudah diakses kapan pun.

3.1.2 Subjek Penelitian

Subjek dalam penelitian ini adalah pihak-pihak yang terlibat dalam penggunaan dan pengelolaan sistem informasi karyawan dan presensi di PT Kendi Arindo, yaitu *HRD*, kepala bagian umum dan administrasi, staf administrasi, serta karyawan sebagai pengguna akhir sistem. Kepala bagian umum dan administrasi berperan dalam pengelolaan data kepegawaian dan penyusunan laporan presensi, staf administrasi bertanggung jawab dalam *penginputan* serta pengelolaan data, sedangkan karyawan merupakan pengguna utama sistem presensi melalui

pemindaian *QR Code*. Subjek penelitian ini dipilih karena mereka berinteraksi langsung dengan sistem yang akan dikembangkan dan menjadi sumber informasi utama dalam analisis kebutuhan serta evaluasi sistem.

3.1.3 Sejarah

PT. Kendi Arindo (PT.KA) awalnya merupakan perusahaan yang bergerak dibidang produksi dan ekspor arang kayu dimana hingga saat ini arang diproduksi berasal dari bahan baku limbah kayu gergajian *sawmill* berupa kayu babiran atau pemanfaatan limbah kayu yang terdapat pada tapak lahan perkebunan. Dengan memperkirakan sumber bahan baku akan semakin sedikit maka perusahaan berinisiatif untuk membangun kebun bahan baku berupa perkebunan bahan baku kayu. Dengan target produksi arang awal sebanyak 10.000 ton per tahun PT.KA membutuhkan bahan baku kayu kurang lebih 70.000 m³ /tahun atau ekuivalen produksi tebangan pohon sekitar 467 hektar / tahun tanaman mudah tumbuh dengan daur 6 tahun; berarti dibutuhkan luas efektif tanaman sekitar 3.250 hektar.

Selanjutnya dalam perjalanannya terjadi perubahan peraturan pemerintah terutama penyelenggaraan perizinan yang semula perizinan manual saat ini sudah menggunakan *system* berupa *online single submission* (OSS) yang merupakan pelaksanaan undang undang No. 11 tahun 2022 tentang Cipta Kerja. OSS-RBA (*OSS Risk Based Approach*) wajib digunakan oleh pelaku usaha, kementerian/lembaga pemerintah daerah, administrasi kawasan ekonomi khusus dan badan pengusaha kawasan.

Semenjak terbit dan diimplementasikannya PP No.5 Tahun 2021 tersebut beberapa KBLI diiadakan termasuk KBLI 02117 (Pengelolaan Tanaman Akasia) dan KBLI 02118 (Pengelolaan Tanaman Eukaliptus) yang merupakan sandaran hukum berdirinya perkebunan PT.Kendi Arindo (PT.KA) sebagai perkebunan penyedia bahan baku arang harus berubah. Setelah melakukan koordinasi, konsultasi dengan berbagai pihak yang terkait hingga kementerian Lingkungan Hidup dan Kehutanan, karena status lahan PT.KA adalah Area Peruntukan Lain / APL (bukan lahan Kawasan hutan), maka bentuk izin usaha PT.KA harus berupa perkebunan dan harus mengelola tanaman perkebunan bukan jenis tanaman kehutanan (termasuk diantaranya adalah akasia dan eukaliptus).

Memohon perizinan baru dengan *format* sebagai usaha bidang perkebunan tanaman karet. Sehingga pada tanggal 24 Mei 2022 melalui surat direktur PT.KA mengajukan permohonan AMDAL perkebunan karet dan industri arang PT.KA perkebunan. Melalui surat No.660/80/PPLH/DLH2022 Kepala Dinas lingkungan hidup Kabupaten Empat Lawang memberikan arahan kepada PT.KA untuk menyusun dan menilai AMDAL baru kategori B.

3.1.4 Visi Dan Misi PT Kendi Arindo

Sebagai salah satu perusahaan yang bergerak di bidang perkebunan karet, PT Kendi Arindo memiliki komitmen kuat untuk berperan aktif dalam mendukung pembangunan nasional melalui pengelolaan sumber daya alam yang berkelanjutan. Perusahaan ini berfokus pada pengembangan usaha perkebunan yang tidak hanya berorientasi pada produktivitas, tetapi juga memperhatikan aspek lingkungan dan kesejahteraan masyarakat sekitar.

Dalam menjalankan kegiatan operasionalnya, PT Kendi Arindo menetapkan arah dan pedoman kerja yang tertuang dalam visi dan misi perusahaan sebagai landasan dalam mencapai keberhasilan dan keberlanjutan usaha.

1. Visi

Menjadi Perusahaan perkebunan karet terkemuka bersama masyarakat sekitar melalui peningkatan pertumbuhan dan hasil getah karet serta hasil hutan lainnya secara berkelanjutan dan meningkat dari waktu ke waktu

2. Misi

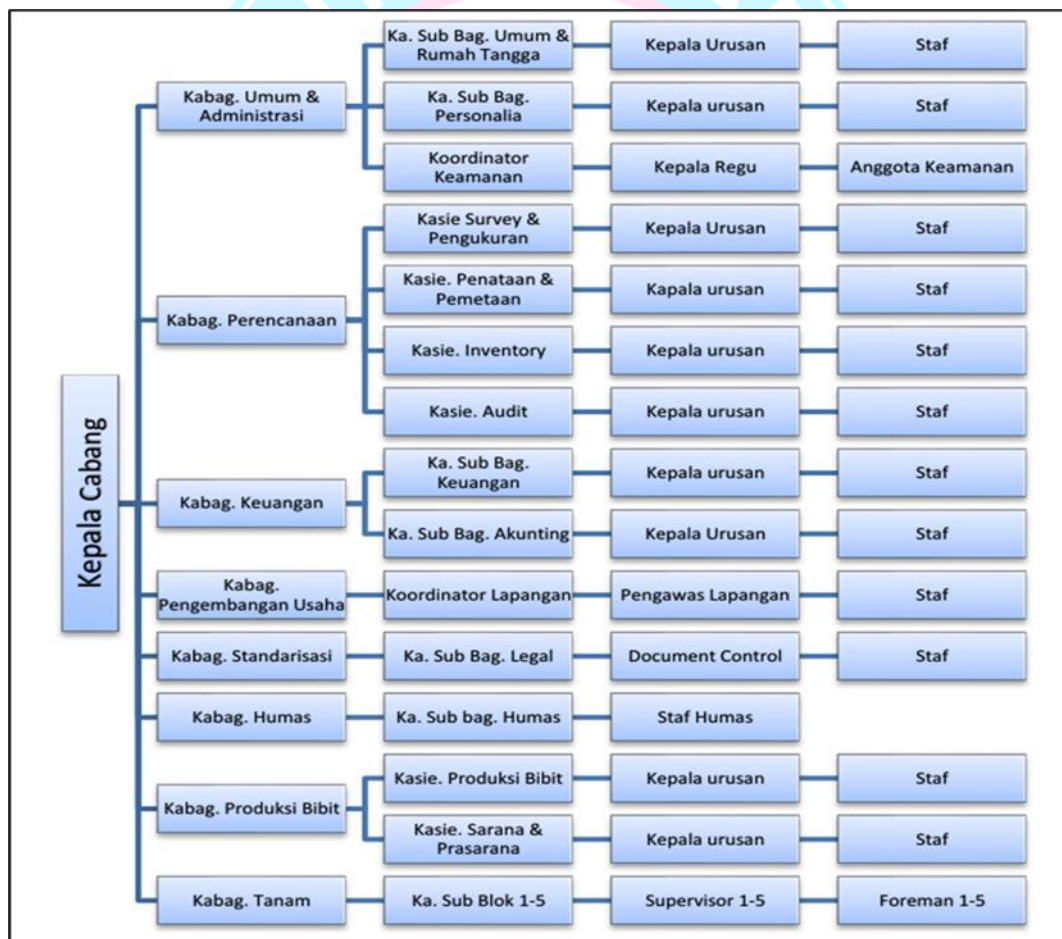
Misi Utama PT. KA adalah menunjang pembangunan nasional di bidang perkebunan khususnya penyediaan bahan baku getah karet untuk berbagai keperluan industri, peningkatan produktivitas lahan, peningkatan kualitas lingkungan hidup dan menjalankan usaha dalam bidang pengusahaan dan pengelolaan lahan, yang diimplementasikan dalam kegiatan operasional, meliputi:

- a. Kegiatan pengelolaan perkebunan secara lestari dan berwawasan lingkungan.
- b. Meningkatkan kuantitas dan kualitas tegakan pokok karet untuk memenuhi kebutuhan bahan baku industri karet.

- c. Meningkatkan kualitas lingkungan hidup melalui rehabilitasi lahan yang tidak produktif.
- d. Memperluas kesempatan kerja dan berusaha.
- e. Meningkatkan pendapatan dan kesejahteraan karyawan dan masyarakat sekitar.

3.1.5 Struktur Organisasi

Struktur organisasi merupakan hal penting dalam suatu perusahaan karena menggambarkan pembagian tugas, wewenang, dan tanggung jawab setiap bagian di dalamnya. Adapun struktur organisasi PT Kendi Arindo dapat dilihat pada gambar 3.1.



Sumber : PT.Kendi Arindo ,2024

Gambar 3. 1 Struktur Organisasi

Berdasarkan Gambar 3.1 struktur organisasi PT Kendi Arindo, dapat dijelaskan bahwa struktur ini menggambarkan susunan dan hubungan antar unsur yang berperan dalam penyelenggaraan kegiatan kerja di PT Kendi Arindo.

1. Kepala Cabang

Bertanggung jawab penuh terhadap seluruh kegiatan operasional di tingkat cabang. Kepala cabang mengoordinasikan seluruh bagian agar kegiatan perusahaan berjalan sesuai dengan kebijakan dan target yang telah ditetapkan oleh manajemen pusat.

2. Kabag. Umum & Administrasi

Mengelola kegiatan administrasi umum, personalia, dan keamanan perusahaan. Bagian ini memastikan seluruh kebutuhan operasional, kepegawaian, dan tata usaha berjalan dengan baik.

- a. Ka. Sub Bag. Umum & Rumah Tangga: Mengatur kebutuhan fasilitas kantor, perawatan aset, dan layanan rumah tangga perusahaan.
- b. Ka. Sub Bag. Personalia: Mengelola data karyawan, *absensi*, penggajian, serta urusan kepegawaian lainnya.
- c. Koordinator Keamanan: Bertanggung jawab atas keamanan lingkungan kerja dan pengawasan anggota keamanan.

3. Kabag. Perencanaan

Mengatur dan menyusun rencana kegiatan operasional, termasuk survei, pemetaan, dan audit internal.

4. Kabag. Keuangan

Bertanggung jawab dalam pengelolaan dan pelaporan keuangan perusahaan cabang.

- a. Ka. Sub Bag. Keuangan: Mengatur arus kas, pembayaran, dan administrasi keuangan.
- b. Ka. Sub Bag. Akunting: Menyusun laporan keuangan, pembukuan, dan pengawasan transaksi keuangan.

5. Kabag. Pengembangan Usaha

Bertugas mencari peluang pengembangan baru, memperluas jaringan usaha, dan memastikan perusahaan tetap kompetitif di bidang perkebunan karet.

6. Kabag. Standarisasi

Mengawasi penerapan standar mutu, efisiensi kerja, serta kepatuhan terhadap prosedur perusahaan agar seluruh kegiatan operasional memenuhi standar yang ditetapkan.

7. Kabag. Humas

Menjalin komunikasi dengan pihak eksternal, termasuk masyarakat dan pemerintah daerah.

- a. Ka. Sub Bag. Legal: Mengurus dokumen hukum, perizinan, dan kontrak kerja sama.
- b. Ka. Sub Bag. Humas: Mengelola hubungan publik dan penyampaian informasi dari perusahaan kepada pihak luar.

8. Kabag. Produksi Bibit

Bertanggung jawab terhadap seluruh kegiatan produksi bibit tanaman karet mulai dari pembibitan hingga distribusi ke lapangan.

- a. Kasie Produksi Bibit: Mengawasi kegiatan produksi bibit agar sesuai standar kualitas.
- b. Kasie Sarana & Prasarana: Menjamin ketersediaan dan pemeliharaan alat serta fasilitas produksi.

9. Kabag. Tanam

Mengatur kegiatan penanaman, pemeliharaan, dan pengawasan blok tanaman di lapangan.

- a. Ka. Sub Blok 1–5: Mengelola masing-masing area atau blok perkebunan.
- b. *Supervisor* 1–5: Mengawasi pelaksanaan kegiatan lapangan agar sesuai dengan rencana kerja.
- c. *Foreman* 1–5: Melaksanakan pengawasan langsung terhadap pekerja di lapangan dan memastikan kegiatan berjalan efektif.

Struktur organisasi ini menunjukkan bahwa PT Kendi Arindo memiliki sistem manajemen yang terstruktur dengan pembagian tanggung jawab yang jelas di setiap bagian, sehingga seluruh proses dari administrasi, keuangan, hingga produksi bibit dan penanaman dapat berjalan secara efisien dan terkoordinasi.

3.1.6 Jadwal Penelitian

Pelaksanaan penelitian ini direncanakan selama kurang lebih enam bulan, dimulai pada tanggal 21 Oktober 2025 dan berakhir pada 21 April 2026. Dalam kurun waktu tersebut, kegiatan penelitian akan dilaksanakan secara bertahap, meliputi penyusunan rencana penelitian, pengumpulan dan pengolahan data, analisis kebutuhan sistem, perancangan serta pengembangan sistem, hingga tahap evaluasi dan penyusunan laporan akhir. Dengan pembagian waktu yang terencana, diharapkan seluruh tahapan penelitian dapat berjalan efektif dan menghasilkan hasil penelitian yang optimal. Adapun rincian jadwal pelaksanaan penelitian dapat dilihat pada tabel 3.1 berikut:

Tabel 3.1 Jadwal Penelitian

No	Tahapan Kegiatan	Okt 2025	Nov 2025	Des 2025	Jan 2026	Feb 2026	Mar 2026	Apr 2026
1	Penyusunan rencana penelitian	✓	✓					
2	Pengeumpulan & pengolaan data		✓	✓				
3	Analisis kebutuhan sistem			✓	✓			
4	Perancangan dan pengembangan sistem					✓		
5	Eveluasi dan pengujian sistem						✓	
6	Penyusunan laporan akhir & revisi akhir							✓

Sumber: data diolah oleh penulis, 2026

3.2 Metode Penelitian dan Teori Penelitian

3.2.1 Metode Penelitian

Metode penelitian adalah cara atau langkah-langkah sistematis yang digunakan oleh peneliti untuk memperoleh data dan informasi yang dapat digunakan dalam menjawab pertanyaan penelitian atau memecahkan suatu masalah. Metode ini berfungsi sebagai pedoman dalam pelaksanaan penelitian agar prosesnya berjalan secara ilmiah, terarah, dan dapat dipertanggung jawabkan. Metode penelitian merupakan serangkaian kegiatan dalam mencari kebenaran suatu

studi penelitian, yang diawali dengan suatu pemikiran yang membentuk rumusan masalah sehingga menimbulkan hipotesis awal, dengan dibantu dan persepsi penelitian terdahulu, sehingga penelitian bisa diolah dan dianalisis yang akhirnya membentuk suatu kesimpulan (Sahir, 2022).

Sejalan dengan fokus penelitian yang menekankan digitalisasi sistem informasi karyawan berbasis aplikasi *android*, pendekatan kualitatif digunakan untuk memberikan gambaran menyeluruh mengenai proses perancangan, pengembangan, dan implementasi sistem. Pendekatan ini memungkinkan peneliti untuk memperoleh pemahaman mendalam terhadap kebutuhan pengguna di lapangan, efektivitas penerapan teknologi *QR Code* dalam presensi karyawan, serta faktor-faktor yang memengaruhi keberhasilan penerapan digitalisasi sistem informasi kepegawaian di PT Kendi Arindo.

3.2.2 Teori Penelitian

Teori penelitian merupakan landasan konseptual yang digunakan untuk menjelaskan, memahami, dan menafsirkan fenomena yang dikaji dalam suatu penelitian sehingga proses penelitian memiliki arah yang jelas, sistematis, dan dapat dipertanggungjawabkan secara ilmiah. Teori berfungsi sebagai kerangka berpikir yang membantu peneliti dalam menyusun rumusan masalah, menentukan variabel yang diteliti, memilih metode yang tepat, serta menjadi dasar dalam menganalisis dan menarik kesimpulan dari hasil penelitian. Dengan adanya teori penelitian, suatu studi tidak hanya bersifat deskriptif, tetapi juga memiliki dasar ilmiah yang kuat karena didukung oleh konsep, prinsip, dan temuan penelitian sebelumnya.

Dalam konteks penelitian skripsi ini yang membahas digitalisasi sistem informasi karyawan dan presensi berbasis aplikasi *android*, teori penelitian digunakan sebagai acuan untuk memahami bagaimana penerapan teknologi informasi dapat meningkatkan efektivitas pengelolaan data kepegawaian. Teori membantu menjelaskan hubungan antara penggunaan sistem digital dengan peningkatan efisiensi kerja, akurasi pencatatan presensi, serta kemudahan akses informasi bagi pihak manajemen. Selain itu, teori juga memberikan dasar pemahaman mengenai pentingnya transformasi digital dalam organisasi, khususnya

dalam menggantikan sistem manual menjadi sistem terkomputerisasi yang lebih cepat, transparan, dan terintegrasi.

Dengan menggunakan landasan teori yang relevan, penelitian ini dapat menganalisis kebutuhan sistem, merancang solusi yang sesuai dengan kondisi lapangan, serta mengevaluasi sejauh mana implementasi sistem presensi berbasis *QR Code* mampu meningkatkan kinerja administrasi kepegawaian. Oleh karena itu, teori penelitian tidak hanya berfungsi sebagai pendukung konseptual, tetapi juga menjadi pedoman dalam merancang, mengembangkan, dan menilai keberhasilan sistem yang dibangun sehingga hasil penelitian memiliki nilai akademik dan manfaat praktis bagi organisasi.

3.3 Sumber Data

Dalam penelitian ini, sumber data merupakan komponen penting yang digunakan untuk memperoleh informasi yang relevan dengan permasalahan penelitian. Sumber data merupakan dari mana data tersebut diperoleh. Sumber data dapat berasal dari manusia maupun non manusia. Sumber data dalam penelitian kualitatif dinamakan sebagai narasumber atau partisipan dalam penelitian. Sumber data yang digunakan dalam penelitian ini adalah data primer dan data sekunder (Zulkipli, 2022).

Pemilihan sumber data yang tepat bertujuan untuk memastikan bahwa informasi yang diperoleh benar-benar menggambarkan kondisi nyata di lapangan serta mendukung tercapainya tujuan penelitian. Data primer dalam penelitian ini diperoleh langsung dari pihak-pihak yang terlibat dalam pengelolaan kepegawaian dan presensi di PT Kendi Arindo, sedangkan data sekunder diperoleh dari berbagai dokumen, arsip, dan literatur yang relevan. Melalui kombinasi kedua jenis sumber data tersebut, peneliti dapat melakukan analisis yang lebih komprehensif dan mendalam terhadap proses digitalisasi sistem informasi kepegawaian dan presensi berbasis *QR Code* yang dikembangkan.

3.3.1 Data Primer

Data primer adalah data yang dikumpulkan langsung oleh peneliti untuk tujuan penelitian tertentu (Daruhadi & Sopiati, 2024). Dalam penelitian ini, data

primer diperoleh melalui wawancara dan observasi langsung terhadap kegiatan operasional di PT Kendi Arindo, khususnya pada bagian kepegawaian dan sistem presensi karyawan. Sumber data primer dalam penelitian ini adalah:

1. Kepala Bagian Umum dan Administrasi memberikan informasi terkait kebijakan umum perusahaan yang berkaitan dengan kepegawaian dan sistem presensi. Dari narasumber ini, peneliti memperoleh gambaran mengenai prosedur absensi yang berlaku, alur pengelolaan data kehadiran, serta kendala yang sering muncul dalam pembuatan laporan presensi dan kepegawaian. Informasi ini penting untuk memahami bagaimana sistem kehadiran *fingerprint* diterapkan secara *formal* di lingkungan perusahaan.
2. Kepala Bagian Perencanaan memberikan data terkait kebutuhan perusahaan dalam perencanaan sumber daya manusia, terutama yang berkaitan dengan keakuratan data presensi sebagai dasar penyusunan laporan kepegawaian, evaluasi kinerja, dan perencanaan jadwal kerja. Dari narasumber ini, peneliti memperoleh informasi mengenai pentingnya ketersediaan data kehadiran yang cepat dan akurat, serta harapan terhadap sistem presensi berbasis *QR Code* yang dapat mendukung proses perencanaan tersebut.
3. Staf *HRD (Human Resources Development)* memberikan informasi teknis dan operasional mengenai proses pencatatan kehadiran, *penginputan* data, hingga pembuatan rekap presensi dan laporan kepegawaian. Dari narasumber ini, peneliti dapat mengetahui secara langsung kendala yang dihadapi dalam penggunaan sistem *fingerprint*, seperti keterlambatan rekap, risiko kesalahan *input*, serta kesulitan dalam mengakses data ketika dibutuhkan. Selain itu, staff *HRD (Human Resources Development)* juga menyampaikan kebutuhan fitur-fitur yang diharapkan pada sistem presensi berbasis *QR Code* dan aplikasi *android* agar dapat mempermudah pekerjaan sehari-hari.

Data primer dari ketiga sumber ini digunakan untuk mengetahui kondisi sistem kehadiran yang saat ini masih menggunakan *fingerprint*, kebutuhan pengguna terhadap sistem presensi berbasis *QR Code*, serta kendala yang dihadapi dalam proses pengelolaan data kehadiran dan pembuatan laporan kepegawaian sebelum diterapkannya sistem digital berbasis aplikasi *android*.

3.3.2 Data Sekunder

Data sekunder adalah data yang sudah ada, yang dikumpulkan untuk tujuan lain tetapi digunakan oleh peneliti untuk analisis lanjutan (Daruhadi & Sopiati, 2024). Data ini dapat berupa buku, jurnal ilmiah, artikel, laporan penelitian terdahulu, serta dokumen resmi yang relevan dengan topik penelitian. Sumber data sekunder dalam penelitian ini meliputi:

1. Arsip dokumen kepegawaian, seperti data biodata karyawan, riwayat kerja, dan status kepegawaian, digunakan untuk mengetahui struktur organisasi, jumlah karyawan, serta pola pengelolaan administrasi kepegawaian di perusahaan. Dokumen ini membantu peneliti memahami konteks kebutuhan sistem informasi kepegawaian yang lebih terstruktur dan terdigitalisasi.
2. Rekap presensi dan laporan kehadiran digunakan untuk melihat bagaimana data kehadiran selama ini dicatat, disimpan, serta disajikan dalam bentuk laporan. Dari data ini, peneliti dapat menganalisis pola kehadiran karyawan, frekuensi keterlambatan, serta potensi kendala yang muncul dalam proses rekap manual atau semi-manual yang berbasis *fingerprint*. Dokumen ini juga menjadi acuan untuk membandingkan efektivitas sistem presensi sebelum dan sesudah digitalisasi berbasis *QR Code*.
3. Dokumen ini meliputi peraturan perusahaan tentang jam kerja, aturan absensi, sanksi keterlambatan, serta prosedur pelaporan kehadiran. Data ini digunakan untuk memastikan bahwa sistem presensi berbasis *QR Code* dan aplikasi *android* yang dikembangkan tetap sesuai dengan kebijakan yang berlaku di perusahaan, serta dapat mendukung penerapan disiplin kehadiran karyawan secara lebih terukur.
4. Sumber-sumber ini digunakan sebagai landasan teoritis dalam menyusun kerangka konsep penelitian. Literatur yang digunakan berkaitan dengan digitalisasi sistem informasi kepegawaian, teknologi *QR Code* dalam sistem presensi, serta pengembangan aplikasi *android* untuk manajemen data. Data sekunder ini membantu peneliti memperkuat argumentasi teoritis, membandingkan hasil penelitian terdahulu, serta menempatkan penelitian ini dalam konteks perkembangan teknologi informasi di bidang manajemen kepegawaian.

Data sekunder tersebut berfungsi untuk mendukung analisis dan pembahasan hasil penelitian yang diperoleh dari data primer, serta menjadi dasar dalam merancang sistem informasi kepegawaian dan presensi berbasis *QR Code* yang sesuai dengan kebutuhan PT Kendi Arindo.

3.4 Teknik Pengumpulan Data

Pengumpulan data merupakan tahapan penting dalam pelaksanaan penelitian ini karena kualitas data yang diperoleh sangat berpengaruh terhadap ketepatan dan keberhasilan perancangan sistem yang dikembangkan. Teknik pengumpulan data berfungsi sebagai sarana untuk memperoleh informasi dari lapangan yang nantinya diolah untuk memverifikasi hipotesis atau menjawab pertanyaan penelitian (Romdona et al., 2025). Untuk memperoleh data yang akurat dan relevan, penelitian ini menggunakan beberapa metode pengumpulan data, yaitu observasi, wawancara, dokumentasi, dan studi literatur.

Melalui kombinasi keempat teknik tersebut, peneliti dapat memperoleh gambaran yang lebih komprehensif mengenai kondisi nyata sistem presensi dan pengelolaan data kepegawaian di PT Kendi Arindo. Observasi dan wawancara digunakan untuk menggali informasi langsung dari pengguna dan pelaksana sistem, sehingga kebutuhan, kendala, serta harapan terhadap sistem baru dapat teridentifikasi dengan jelas. Sementara itu, dokumentasi dan studi literatur dimanfaatkan untuk melengkapi data lapangan dengan bukti tertulis dan dasar teoritis, sehingga perancangan sistem presensi berbasis *QR Code* dan aplikasi *android* yang dikembangkan tidak hanya sesuai dengan kondisi operasional di lapangan, tetapi juga selaras dengan konsep-konsep ilmiah yang relevan.

3.4.1 Observasi

Observasi merupakan merupakan teknik pengumpulan data yang dilakukan melalui sesuatu pengamatan, dengan disertai pencatatan-pencatatan terhadap keadaan atau perilaku objek sasaran. Observasi adalah pengamatan dan pencatatan yang sistematis terhadap gejala-gejala yang diteliti (M. P. Hasibuan et al., 2023).

Dalam penelitian ini, observasi dilakukan dengan mengamati secara langsung proses pengelolaan data karyawan dan sistem presensi di PT Kendi Arindo. Tujuan

dari observasi ini adalah untuk mengetahui kondisi nyata sistem kepegawaian yang sedang berjalan, termasuk mekanisme pencatatan kehadiran menggunakan *fingerprint*, efisiensi kerja bagian *HR (Human Resources)*, serta kendala yang dihadapi dalam rekapitulasi laporan presensi.

Dari hasil observasi ditemukan bahwa sistem presensi yang ada masih memiliki kekurangan, seperti potensi kecurangan, keterlambatan pelaporan, dan kurangnya transparansi data kehadiran. Temuan ini menjadi dasar pengembangan digitalisasi sistem informasi karyawan berbasis aplikasi *android* agar proses pencatatan dan pelaporan kehadiran menjadi lebih cepat, akurat, serta terintegrasi secara *real-time*.

3.4.2 Wawancara

Wawancara adalah proses tanya jawab lisan antara dua orang atau lebih untuk mengetahui tanggapan dan pendapat seseorang terhadap suatu objek (Sutikno et al., 2021). Wawancara merupakan cara yang paling tepat untuk mengungkapkan keadaan pribadi dari orang yang diwawancarai. Dalam penelitian ini, wawancara dilakukan dengan bagian *HR (Human Resources)* dan beberapa karyawan PT Kendi Arindo sebagai informan utama.

Tujuannya adalah untuk memperoleh informasi mendalam mengenai proses pengelolaan data karyawan, mekanisme presensi menggunakan *fingerprint*, kendala yang sering muncul, serta kebutuhan akan sistem yang lebih efisien dan terintegrasi. Hasil wawancara ini menjadi dasar dalam merancang fitur dan fungsi pada digitalisasi sistem informasi karyawan berbasis aplikasi *android*, agar aplikasi yang dikembangkan benar-benar sesuai dengan kebutuhan pengguna di lapangan dan mampu meningkatkan efektivitas proses administrasi kepegawaian serta akurasi data presensi.

3.4.3 Dokumentasi

Dokumentasi adalah metode mencari data mengenai hal-hal yang berupa catatan, buku, transkrip, surat kabar, prasasti, majalah, notulen rapat, agenda serta foto-foto kegiatan (Saádi Ahmad, 2025). Sumber dokumentasi dapat berupa catatan, laporan, buku, arsip, transkrip, agenda, maupun foto kegiatan yang relevan

dengan topik penelitian. Dalam penelitian ini, dokumentasi dilakukan dengan mengumpulkan berbagai dokumen dan data pendukung yang berkaitan dengan pengelolaan data karyawan dan sistem presensi di PT Kendi Arindo.

Dokumen yang dikumpulkan meliputi data kepegawaian, laporan kehadiran, rekap absensi *fingerprint*, serta arsip internal perusahaan yang berhubungan dengan sistem administrasi kehadiran. Data hasil dokumentasi ini dianalisis untuk memahami alur kerja sistem kepegawaian yang berjalan saat ini, kebutuhan pencatatan presensi, serta kendala yang sering muncul dalam proses administrasi. Teknik dokumentasi ini membantu peneliti dalam menyesuaikan rancangan sistem digitalisasi sistem informasi karyawan dan presensi menggunakan *QR Code* berbasis aplikasi *android* agar sesuai dengan kebutuhan perusahaan dan mampu meningkatkan efisiensi serta akurasi pengelolaan data kepegawaian.

3.4.4 Studi Literatur

Studi literatur atau studi kepustakaan adalah metode pengumpulan data yang melibatkan analisis dan penelaahan terhadap nuku buku, literatur-literatur, catatan-catatan, dan laporan-laporan yang relevan dengan masalah yang sedang di investigasikan (Habsy et al., 2023). Studi literatur dilakukan dengan menelaah berbagai sumber seperti jurnal ilmiah, buku, artikel, dan penelitian terdahulu yang berkaitan dengan sistem informasi kepegawaian, digitalisasi data, serta penerapan teknologi berbasis aplikasi *android*.

Tujuan dari studi literatur ini adalah untuk memperoleh landasan teori yang kuat dalam pengembangan digitalisasi sistem informasi karyawan dan presensi menggunakan *QR Code* berbasis aplikasi *android*, serta memahami metode, konsep, dan teknologi yang tepat untuk meningkatkan efisiensi, akurasi, dan transparansi dalam pengelolaan data kehadiran dan administrasi kepegawaian di PT Kendi Arindo.

3.5 Metode Pengembangan Sistem

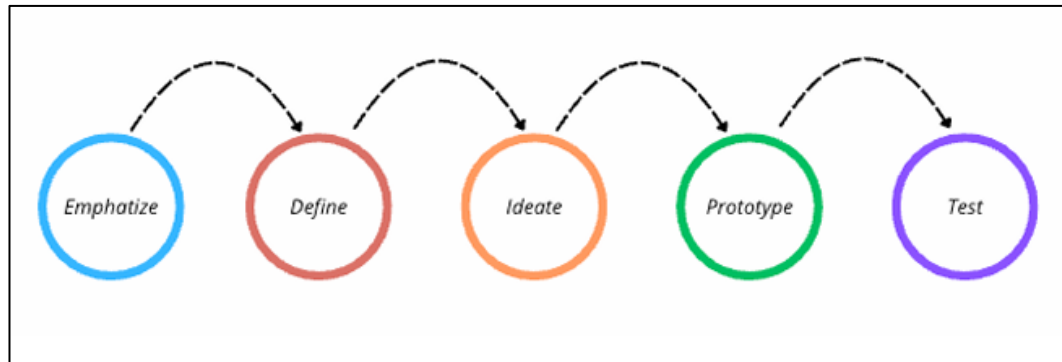
Metode pengembangan sistem secara umum diartikan sebagai urutan langkah-langkah yang terstruktur untuk mengembangkan sebuah sistem informasi berbasis *computer* (Akbar, 2022). Dalam penelitian ini digunakan metode

pengembangan sistem *design thinking*, yaitu pendekatan yang berpusat pada pengguna (*user-centered approach*) dengan tujuan menghasilkan solusi inovatif berdasarkan kebutuhan nyata di lapangan. Metode ini dipilih karena sesuai dengan fokus penelitian yang menekankan pengembangan aplikasi *android* digitalisasi sistem informasi karyawan dan presensi menggunakan *QR Code*, di mana pemahaman terhadap kebutuhan pengguna baik karyawan maupun pimpinan perusahaan menjadi faktor utama keberhasilan sistem.

Design Thinking adalah pendekatan yang melibatkan kolaborasi dengan pengguna untuk menemukan solusi bagi suatu masalah. Tujuan utamanya adalah menciptakan layanan yang inovatif, sesuai dengan kebutuhan pengguna, dan dapat mengatasi masalah yang ada (Ansori & Hendradi, 2023). Adapun alasan peneliti menggunakan metode *Design Thinking* adalah sebagai berikut:

1. Berorientasi pada kebutuhan pengguna. Setiap tahapan difokuskan pada pengalaman pengguna (*user experience*) sehingga sistem yang dikembangkan benar-benar sesuai dengan kebutuhan karyawan dan pimpinan.
2. Fleksibel dan adaptif. Metode ini memungkinkan perbaikan dan penyesuaian ide secara berulang selama proses pengembangan berlangsung.
3. Mendukung inovasi digital. *Design Thinking* mendorong pengembang untuk menciptakan solusi kreatif dalam digitalisasi sistem informasi kepegawaian.
4. Efisien dalam pengujian konsep. Proses *prototyping* dan *testing* dilakukan lebih awal, sehingga kesalahan dapat diminimalkan sebelum sistem dikembangkan sepenuhnya.

Sebelum dilakukan implementasi sistem, tahapan pengembangan *Design Thinking* digambarkan dalam Gambar 3.2 di bawah ini. Diagram ini menunjukkan lima langkah utama yang saling berhubungan secara iteratif, dimulai dari pemahaman terhadap pengguna hingga pengujian sistem. Setiap tahap menghasilkan keluaran yang menjadi dasar perbaikan atau pengembangan tahap berikutnya.



Sumber: Dhieka Avrilia Lantana Fauziah, 2025

Gambar 3. 2 Metode Pengembangan Sistem *Design Thinking*

Dari gambar 3. 2 tentang tahapan metode pengembangan *Design Thinking* dapat dijelaskan dan disimpulkan sebagai berikut:

1 *Empathize* (Memahami Pengguna)

Tahap pertama dalam proses *design thinking* adalah *empathize* atau empati, yaitu membangun pemahaman mendalam terhadap pengguna, termasuk kebutuhan, keinginan, motivasi, dan tantangan yang mereka hadapi (Lantana & Fauziah, 2025). Pada tahap ini penulis telah melakukan observasi dan wawancara terhadap bagian *HR (Human Resources)* dan karyawan PT Kendi Arindo untuk memahami permasalahan sistem kehadiran berbasis *fingerprint* yang masih digunakan. Data dikumpulkan untuk mengetahui kendala utama, seperti risiko titip absen, keterlambatan pelaporan, dan kurangnya transparansi data presensi.

2 *Define* (Menentukan Permasalahan)

Setelah memahami kebutuhan dan perilaku pengguna melalui tahap *empathize*, langkah berikutnya dalam proses *design thinking* adalah *define*. Tahapan ini berfungsi untuk merumuskan masalah secara fokus dan mendalam berdasarkan temuan yang telah dikumpulkan (Lantana & Fauziah, 2025). Pada tahap ini penulis telah melakukan analisis terhadap informasi yang diperoleh untuk merumuskan inti permasalahan. Didefinisikan bahwa perusahaan memerlukan sistem presensi dan manajemen data karyawan berbasis aplikasi *android* yang mampu mencatat kehadiran secara *real-time*, aman, dan mudah diakses oleh pimpinan.

3 *Ideate* (Menghasilkan Ide Solusi)

Setelah merumuskan pernyataan masalah yang jelas pada tahap *define*, proses selanjutnya adalah *ideate*. Tahapan ini bertujuan untuk menghasilkan sebanyak mungkin ide solusi yang inovatif dan relevan dengan permasalahan pengguna. Pada tahap ini penulis telah menyusun ide solusi berupa rancangan sistem digital berbasis aplikasi *android* dengan fitur utama: manajemen data karyawan, presensi menggunakan *QR Code*, dan laporan kehadiran yang komprehensif. Ide-ide ini dituangkan dalam bentuk sketsa awal (*wireframe*) dan alur sistem.

4 *Prototype* (Membuat Prototipe Sistem)

Tahap *Prototype* adalah momen ketika ide-ide yang telah dipilih dalam tahap *Ideate* mulai diwujudkan ke dalam bentuk nyata (Lantana & Fauziah, 2025). Pada tahap ini penulis akan melakukan pengembangan prototipe aplikasi *android* menggunakan flutter yang menggambarkan fungsi utama sistem, termasuk pemindaian *QR Code*, penyimpanan data presensi ke dalam *database*, dan tampilan laporan kehadiran bagi pimpinan.

5 *Test* (Uji Coba Sistem)

Tahap *Test* adalah fase akhir dari proses *design thinking*, dimana solusi yang telah diwujudkan dalam bentuk prototipe diuji langsung kepada pengguna sesungguhnya. Tujuannya Adalah untuk mendapatkan umpan balik yang jujur dan mendalam mengenai bagaimana pengguna merasakan, memahami, dan menggunakan solusi yang ditawarkan (Lantana & Fauziah, 2025). Pada tahap ini penulis akan melakukan pengujian sistem menggunakan metode *White-Box Testing* untuk memastikan logika program berjalan sesuai rancangan. Uji coba akan dilakukan pada beberapa pengguna internal PT Kendi Arindo untuk menilai keakuratan data presensi, efisiensi proses, serta kemudahan penggunaan aplikasi.

3.6 Alat Bantu Perancangan

Perancangan sistem adalah sebuah kegiatan merancang dan menentukan cara mengolah sistem informasi dari hasil analisa sistem sehingga dapat memenuhi kebutuhan dari pengguna termasuk diantaranya perancangan *user interface*, data

dan aktivitas proses (Putri & Pratiwi, 2021). Alat bantu perancangan sistem pada penelitian ini berperan penting dalam membantu peneliti merancang, menggambarkan, serta mengimplementasikan sistem. Dalam penelitian ini, peneliti menggunakan beberapa alat bantu, baik untuk perancangan sistem, desain antarmuka, maupun proses *debugging* dan pengembangan.

Penggunaan alat bantu tersebut diharapkan dapat mendukung proses pengembangan sistem presensi berbasis *QR Code* dan aplikasi *android* secara lebih terstruktur dan efisien. Dengan adanya *pemodelan* sistem, desain antarmuka yang terencana, serta dukungan *tools* pengujian dan *debugging*, peneliti dapat meminimalkan kesalahan, meningkatkan kualitas fungsional sistem, serta memastikan bahwa sistem yang dibangun sesuai dengan kebutuhan pengguna dan mudah diimplementasikan di lingkungan PT Kendi Arindo. Berikut merupakan uraian dari alat bantu yang digunakan:

3.6.1 Alat Bantu Utama

Visual studio Code merupakan *integrated development environment (IDE)* yang digunakan dalam proses pengembangan sistem informasi karyawan dan presensi menggunakan *QR Code* berbasis *android*. *Visual studio Code* adalah sebuah teks *editor* ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows (Surya Ningsih et al., 2022). *IDE* ini menyediakan berbagai fitur seperti *syntax highlighting*, *auto-completion*, dukungan ekstensi, serta integrasi dengan *version control system* seperti Git. Fitur-fitur tersebut memudahkan pengembang dalam menulis, menguji, dan mengelola kode program secara efisien dan terstruktur. Pemilihan *visual studio Code* didasarkan pada keunggulannya yang ringan, fleksibel, serta mendukung berbagai bahasa pemrograman dan *framework modern*, sehingga sangat sesuai untuk pengembangan aplikasi berbasis *android* sistem informasi karyawan di PT Kendi Arindo.

Flutter digunakan sebagai *framework* utama dalam pembuatan aplikasi. *Framework* ini berperan dalam membangun tampilan antarmuka pengguna (*User Interface*) sekaligus mengatur interaksi antara pengguna dan sistem. Dengan pendekatan *single Codebase*, flutter memungkinkan pengembang membuat aplikasi

yang responsif dan memiliki performa tinggi pada berbagai *platform*, khususnya *android*. Selain itu, fitur *hot reload* yang dimilikinya membantu mempercepat proses pengujian dan perbaikan, sehingga pengembangan sistem informasi karyawan dan presensi menggunakan *QR Code* berbasis *android* dapat dilakukan secara lebih efisien, interaktif, dan mudah digunakan oleh karyawan maupun pimpinan perusahaan.

3.6.2 Alat Bantu Pemodelan Visual

Bahasa pemodelan visual digunakan untuk membantu menggambarkan struktur dan alur kerja aplikasi secara konseptual sebelum tahap implementasi dilakukan. Melalui pemodelan visual, pengembang dapat memahami hubungan antar komponen sistem, aliran proses presensi dan pengelolaan data karyawan, serta interaksi antara pengguna (karyawan, *HR (Human Resources)*, dan pimpinan) dengan sistem secara lebih jelas dan terstruktur.

Selain itu, diagram visual juga berperan sebagai sarana komunikasi antara pengembang, pembimbing, dan pihak perusahaan agar rancangan aplikasi dapat dipahami dengan baik sebelum proses pengkodean dimulai. Dengan adanya pemodelan visual ini, pengembangan sistem menjadi lebih terarah, efisien, serta meminimalkan risiko kesalahan pada tahap implementasi. Adapun alat bantu pemodelan visual yang digunakan dalam penelitian ini meliputi:

1. *Use Case Diagram*

Use case diagram digunakan untuk menggambarkan hubungan antara aktor (pengguna) dengan sistem pada aplikasi digitalisasi sistem informasi karyawan dan presensi. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara aktor dengan *system* (Arianti et al., 2022). Berikut adalah simbol-simbol *use case diagram* dapat dilihat pada tabel 3.1 dibawah ini:

Tabel 3. 1 *Use Case Diagram*

No	Nama	Gambar	Keterangan
1	<i>Actor</i>		Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2	<i>Dependency</i>		Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang tidak mandiri (<i>dependent</i>).
3	<i>Generalization</i>		Hubungan dimana objek anak (<i>descendent</i>) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4	<i>Include</i>		Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5	<i>Extend</i>		Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6	<i>Association</i>		Apa yang menghubungkan antara objek satu dengan objek lainnya.
7	<i>System</i>		Menspesifikasikan data paket yang menampilkan <i>system</i> secara terbatas.

Sumber: (Rifaldy et al., 2022)

2. Activity Diagram

Activity diagram atau diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak, yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan yang dilakukan aktor, jadi aktivitas dilakukan oleh sistem (Rifaldy et al., 2022). Selain itu, *activity diagram* juga digunakan untuk memperlihatkan hubungan antar aktivitas, keputusan yang

mungkin terjadi dalam suatu proses, serta kemungkinan percabangan alur kerja. Dengan adanya diagram ini, pengembang sistem dapat memahami proses yang berjalan secara lebih terstruktur dan jelas, sehingga memudahkan dalam tahap analisis, perancangan, maupun pengembangan sistem perangkat lunak. Berikut adalah simbol-simbol *activity diagram* dapat dilihat pada tabel 3.2 dibawah ini:

Tabel 3. 2 *Activity Diagram*

No	Simbol	Keterangan	Uraian
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

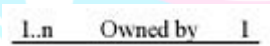
Sumber: (Rifaldy et al., 2022)

3. Class Diagram

Class diagram adalah *class* yang menggambarkan struktur dan penjelasan dari *class*, *package*, dan *object* serta hubungan satu sama lain seperti *containment*, *inheritance*, *associations*, dan lain-lain (Wayahdi & Ruziq, 2023). *Class* adalah abstraksi yang merepresentasikan struktur dan perilaku umum dari sekumpulan *object*. *Object* adalah turunan dari *class* yang dibuat (*created*), dimodifikasi

(*modified*), dan dihancurkan (*destroyed*) selama eksekusi sistem. Sebuah *object* memiliki status yang menyertakan nilai *attribute*-nya dan hubungannya dengan *object* lain. *Class diagram* berfungsi sebagai cara untuk merencanakan dan berkomunikasi antar *developers*. *Class diagram* merepresentasikan struktur statis dari sistem yang mewakili *object class* dalam sistem dan *relationship* pada setiap *class*. Selain itu, *class diagram* digunakan untuk memodelkan tampilan struktur sistem yang mencakup kemampuan untuk membawa data dan menjalankan tindakan (*execute actions*), juga sebagai dasar untuk pengembangan sistem perangkat lunak lebih lanjut. Berikut adalah simbol-simbol *class diagram* dapat dilihat pada tabel 3.3 dibawah ini:

Tabel 3.3 *Class Diagram*

No	Nama	Simbol	Keterangan
1	<i>Dependency</i>		Penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu class yang menggunakan class yang lain.
2	<i>Class</i>	Nama Class + atribut + atribut - method - method	<i>Class</i> adalah blok-blok pembangun pada pemrograman berorientasi objek. <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan atribut <i>class</i> . Bagian akhir mendefinisikan <i>method-method</i> .
3	<i>Association</i>		Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 class dan dilambangkan oleh <i>Association</i> sebuah garis yang menghubungkan antara 2 class. Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum - hukum multiplisitas pada sebuah <i>relationship</i> .
4	<i>Composition</i>		Jika sebuah class tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut.

Sumber: (Malius et al., 2021)

3.6.3 Alat Bantu Desain

Dalam proses pengembangan aplikasi digitalisasi sistem informasi karyawan dan presensi menggunakan *QR Code* berbasis *android*, tahap perancangan memiliki peranan penting untuk mengubah kebutuhan pengguna menjadi rancangan visual dan fungsional dari sistem yang akan dibangun. Tahap ini mencakup pembuatan tata letak tampilan, penentuan elemen antarmuka, pemilihan warna dan ikon, serta penyusunan alur interaksi antara pengguna dengan aplikasi. Proses desain dilakukan sebelum tahap implementasi agar pengembang memiliki panduan yang jelas dalam membangun tampilan dan fitur aplikasi sesuai kebutuhan pengguna.

Figma digunakan sebagai alat bantu perancangan antarmuka pengguna (*User Interface Design Tool*). Figma adalah salah satu *design tool* yang biasanya digunakan untuk membuat tampilan aplikasi *mobile*, *desktop*, *website* dan lain-lain (Al-Faruq et al., 2022). Figma bisa digunakan di sistem operasi windows, linux ataupun mac dengan terhubung ke internet. Melalui figma, peneliti dapat membuat rancangan visual berupa tata letak halaman, posisi elemen, ikon, warna, dan alur navigasi aplikasi sebelum diimplementasikan.

Penggunaan figma bertujuan untuk mempermudah proses perancangan tampilan agar hasil desain sesuai dengan kebutuhan pengguna serta mengikuti prinsip *user experience (UX)* yang baik. Selain itu, figma mendukung kolaborasi secara *real-time* dan penyimpanan berbasis *cloud*, sehingga proses revisi desain dapat dilakukan dengan lebih cepat dan efisien sebelum dikembangkan menggunakan *framework flutter* pada tahap implementasi.

3.6.4 Alat Bantu *Debugging* dan Pengembangan

Dalam proses pengembangan aplikasi, tahap *debugging* memiliki peranan penting untuk memastikan bahwa sistem berfungsi sesuai dengan rancangan dan bebas dari kesalahan (*error*). Kegiatan ini dilakukan untuk menemukan, menganalisis, serta memperbaiki kesalahan pada kode program yang dapat memengaruhi tampilan maupun kinerja aplikasi. Pada penelitian ini, proses *debugging* dilakukan menggunakan *visual studio Code* dengan bantuan fitur *debugger* yang terintegrasi dengan flutter. Aplikasi dijalankan baik melalui

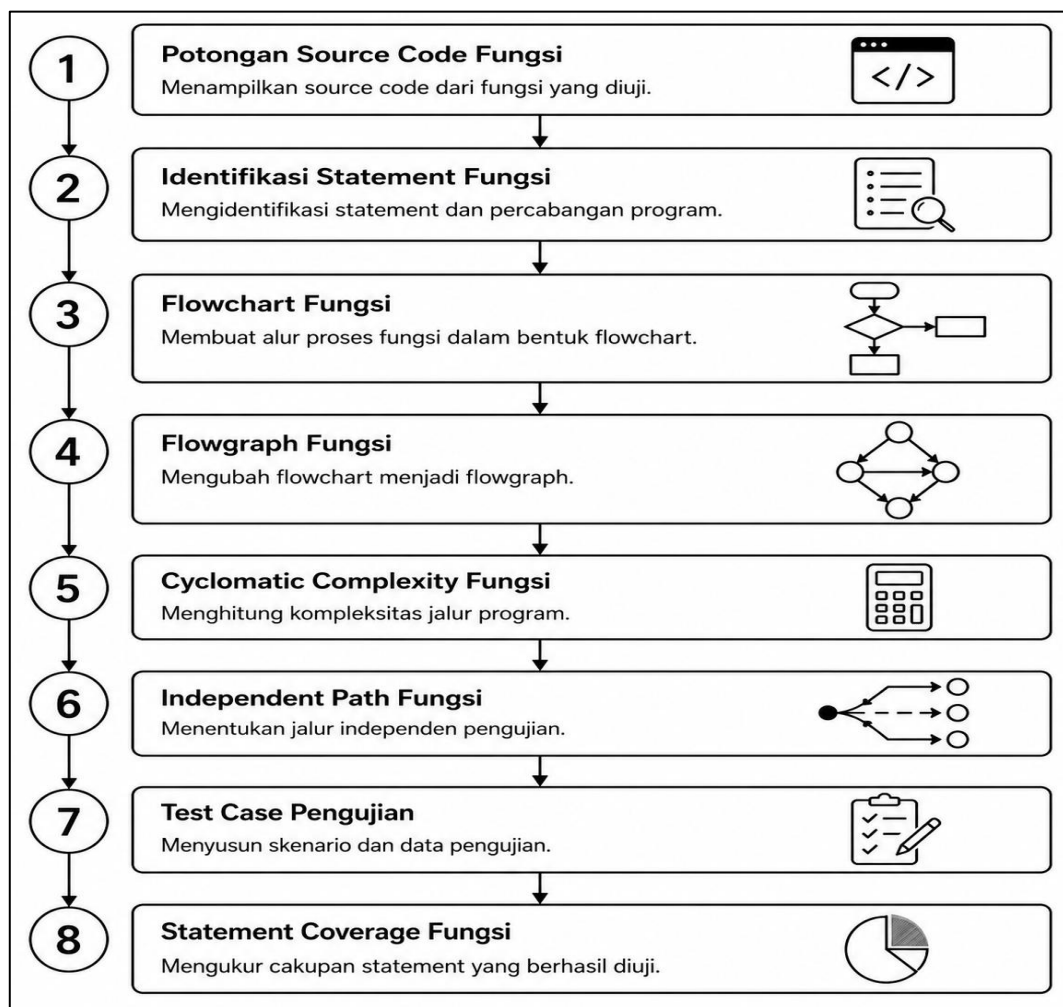
emulator android maupun langsung pada perangkat *smartphone* untuk menguji kestabilan dan fungsionalitas sistem secara nyata.

Melalui proses ini, pengembang dapat memantau jalannya program secara *real-time*, mengenali bagian kode yang bermasalah, serta melakukan perbaikan dengan cepat menggunakan fitur *hot reload* yang disediakan oleh flutter. Ini merupakan fitur yang berfungsi untuk mempercepat dan mempermudah proses pengembangan aplikasi. Setelah melakukan perubahan pada kode program, cukup menekan *hot reload*. Aplikasi akan diperbaharui dalam kurun waktu kurang dari 1 detik (Setiawan et al., 2025). Dengan demikian, proses pengujian dan penyempurnaan aplikasi dapat dilakukan secara lebih efisien, terarah, dan sesuai dengan kebutuhan sistem yang dikembangkan.

3.7 Pengujian Sistem

Pengujian sistem merupakan tahap penting dalam proses pengembangan aplikasi untuk memastikan bahwa seluruh fungsi dan logika program berjalan sesuai dengan kebutuhan serta bebas dari kesalahan (*error*). Tahap ini dilakukan untuk mengevaluasi keandalan, efisiensi, dan stabilitas sistem sebelum diterapkan secara penuh di lingkungan operasional. Pada penelitian ini, pengujian dilakukan menggunakan metode *white-box testing*, adalah metode pengujian yang memeriksa dan menganalisis kode program untuk menemukan kesalahan atau kekurangan dalam suatu aplikasi atau perangkat lunak (Zen & Nuryasin, 2024).

Pengujian sistem merupakan tahap penting dalam proses pengembangan aplikasi untuk memastikan bahwa seluruh fungsi dan logika program berjalan sesuai dengan kebutuhan serta bebas dari kesalahan (*error*). Tahap ini dilakukan untuk mengevaluasi keandalan, efisiensi, dan stabilitas sistem sebelum diterapkan secara penuh di lingkungan operasional. Adapun alur pengujian *white-box testing process* yang dilakukan dalam penelitian ini dapat dilihat pada Gambar 3.3. dibawah ini:



Sumber: (Solissa et al., 2023)

Gambar 3. 3 White Box Testing Process

Berdasarkan Gambar 3.3 di atas, alur pengujian *white-box testing* dilakukan melalui beberapa tahapan yang saling berkaitan. Tahapan-tahapan tersebut dapat dijelaskan sebagai berikut:

1. Potongan *Source Code* Fungsi

Pada tahap ini, penguji menampilkan potongan *source code* dari fungsi yang akan diuji. Tujuan dari tahap ini adalah untuk mengetahui struktur kode program, proses yang dijalankan, serta logika yang digunakan pada fungsi tersebut sehingga pengujian dapat dilakukan secara lebih terarah.

2. Identifikasi *Statement* Fungsi

Tahap selanjutnya adalah mengidentifikasi statement yang terdapat pada fungsi program, seperti percabangan, kondisi, dan proses yang dijalankan

sistem. Identifikasi ini bertujuan untuk mengetahui bagian-bagian kode yang perlu diuji pada proses *white-box testing*.

3. *Flowchart* Fungsi

Pada tahap ini, *source code* fungsi diubah ke dalam bentuk *flowchart*. *Flowchart* digunakan untuk menggambarkan alur proses program secara visual sehingga urutan proses, percabangan, dan keputusan pada fungsi dapat dipahami dengan lebih mudah.

4. *Flowgraph* Fungsi

Setelah *flowchart* dibuat, tahap berikutnya adalah mengubah *flowchart* menjadi *flowgraph*. *Flowgraph* digunakan untuk merepresentasikan hubungan antar *node* dan *edge* pada alur logika program sehingga proses analisis jalur program dapat dilakukan dengan lebih jelas.

5. *Cyclomatic Complexity* Fungsi

Tahap selanjutnya adalah menghitung nilai *cyclomatic complexity* berdasarkan *flowgraph* yang telah dibuat. Nilai *cyclomatic complexity* digunakan untuk mengetahui tingkat kompleksitas logika program serta menentukan jumlah jalur independen yang harus diuji pada fungsi tersebut.

6. *Independent Path* Fungsi

Pada tahap ini, penguji menentukan jalur independen (*independent path*) berdasarkan hasil *flowgraph* dan *cyclomatic complexity*. Jalur independen merupakan jalur logika unik yang harus dilalui dalam proses pengujian agar seluruh alur program dapat diuji secara menyeluruh.

7. *Test Case* Pengujian

Setelah jalur independen diperoleh, penguji menyusun test case pengujian berdasarkan jalur-jalur tersebut. *Test case* digunakan untuk memastikan bahwa setiap proses, kondisi, dan percabangan pada fungsi dapat berjalan sesuai dengan hasil yang diharapkan.

8. *Statement Coverage* Fungsi

Tahap terakhir adalah melakukan *statement coverage* untuk mengetahui tingkat cakupan statement yang berhasil diuji. Pengujian ini bertujuan untuk memastikan bahwa seluruh statement pada fungsi program telah dieksekusi dan diuji dengan baik.

Berdasarkan tahapan tersebut, dapat disimpulkan bahwa *white-box testing* merupakan metode pengujian yang dilakukan secara sistematis dan terstruktur dengan menitikberatkan pada analisis *source code* serta alur logika program. Setiap tahapan saling berkaitan, dimulai dari potongan *source code* fungsi hingga *statement coverage*. Melalui proses ini, pengujian dapat memastikan bahwa seluruh jalur logika program telah diuji dengan baik sehingga kesalahan pada program dapat dideteksi lebih awal dan kualitas sistem menjadi lebih optimal.



BAB IV

ANALISA DAN PERANCANGAN

4.1 Analisa Sistem

4.1.1 Analisa dan Evaluasi Sistem Berjalan:

a. Analisa sistem berjalan

Analisa sistem berjalan dilakukan untuk mengetahui kondisi nyata pengelolaan data karyawan dan presensi yang saat ini diterapkan di PT Kendi Arindo. Berdasarkan hasil observasi, wawancara, dan studi terhadap proses yang berlangsung, diketahui bahwa perusahaan masih menggunakan sistem presensi berbasis *fingerprint* dalam mencatat kehadiran karyawan. Setiap karyawan melakukan pemindaian sidik jari pada saat datang dan pulang kerja, kemudian data kehadiran tersebut direkap secara berkala oleh bagian administrasi.

Meskipun sistem *fingerprint* telah memanfaatkan teknologi digital, dalam penerapannya masih ditemukan beberapa kendala. Salah satu permasalahan utama adalah keterbatasan sistem dalam menyajikan laporan kehadiran secara detail dan *real-time*. Informasi seperti jumlah kehadiran bulanan, data keterlambatan, izin, cuti, maupun hari libur (*off*) belum tersedia secara otomatis dalam satu laporan yang terintegrasi. Proses pengolahan data presensi masih memerlukan rekap lanjutan oleh bagian administrasi sehingga memperlambat penyajian informasi kepada pimpinan perusahaan.

Selain itu, sistem yang berjalan belum terintegrasi dengan data detail karyawan seperti jabatan, divisi, dan status kepegawaian. Akibatnya, pengelolaan data karyawan dan data presensi masih dilakukan secara terpisah. Kondisi ini menyebabkan pihak manajemen mengalami kesulitan dalam memperoleh informasi yang menyeluruh terkait kehadiran dan status karyawan secara cepat dan akurat. Permasalahan lain yang ditemukan adalah masih adanya potensi kecurangan dalam proses presensi, seperti praktik titip absen. Walaupun *fingerprint* menggunakan identitas biometrik, pada kondisi tertentu sistem

masih belum mampu menjamin keamanan dan pengawasan yang optimal. Di samping itu, ketergantungan pada perangkat presensi fisik menyebabkan fleksibilitas sistem menjadi terbatas karena proses kehadiran hanya dapat dilakukan pada titik perangkat tertentu. Berdasarkan kondisi tersebut, diperlukan suatu sistem yang mampu mengintegrasikan data karyawan dan data presensi dalam satu aplikasi digital berbasis *android*. Sistem yang dikembangkan harus mampu memberikan kemudahan dalam proses pencatatan kehadiran, penyimpanan data secara *real-time*, serta penyajian laporan yang lengkap, akurat, dan mudah diakses oleh pihak yang berwenang.

Dalam penelitian ini, analisa sistem berjalan dikaitkan dengan pendekatan *Design Thinking*, yaitu metode pengembangan sistem yang berpusat pada kebutuhan pengguna. Melalui pendekatan ini, proses analisa tidak hanya berfokus pada aspek teknis, tetapi juga memperhatikan pengalaman, kebutuhan, dan kendala yang dialami oleh pengguna sistem, baik dari sisi karyawan sebagai pengguna akhir maupun pihak administrasi dan pimpinan sebagai pengelola serta pengambil keputusan. Dengan demikian, sistem yang akan dikembangkan diharapkan mampu menjadi solusi yang sesuai dengan kondisi nyata di lapangan.

b. Evaluasi sistem berjalan

Evaluasi sistem berjalan dilakukan untuk menilai efektivitas sistem pengelolaan data karyawan dan presensi yang saat ini digunakan di PT Kendi Arindo. Evaluasi ini didasarkan pada hasil analisa yang menunjukkan bahwa sistem presensi *fingerprint* belum mampu memenuhi kebutuhan perusahaan secara optimal, khususnya dalam hal integrasi data, kecepatan akses informasi, dan penyajian laporan.

Berdasarkan hasil evaluasi, sistem yang berjalan saat ini masih memiliki beberapa kelemahan. Pertama, data presensi belum tersaji secara *real-time* dan belum terhubung langsung dengan data detail karyawan. Kedua, proses rekapitulasi masih memerlukan pekerjaan manual oleh bagian administrasi, sehingga berpotensi menimbulkan keterlambatan dan kesalahan pencatatan. Ketiga, laporan kehadiran belum dapat diakses secara cepat dan komprehensif oleh pimpinan perusahaan untuk mendukung *monitoring* dan pengambilan keputusan. Keempat, sistem

fingerprint belum memberikan fleksibilitas yang cukup dalam mendukung pengelolaan presensi yang efisien dan aman.

Berdasarkan hasil evaluasi tersebut, solusi yang ditawarkan dalam penelitian ini adalah pengembangan aplikasi sistem informasi karyawan dan presensi berbasis *android* menggunakan *QR Code*. Aplikasi ini dirancang untuk mengintegrasikan data karyawan dan data presensi ke dalam satu sistem yang terpusat, menyediakan proses presensi yang lebih praktis, serta menyajikan laporan kehadiran secara *real-time* bagi pihak yang berwenang. Dengan adanya sistem baru ini, diharapkan permasalahan yang terdapat pada sistem lama dapat diminimalkan dan proses administrasi kepegawaian menjadi lebih efektif dan efisien.

4.1.2 Analisa Kebutuhan:

a. Kebutuhan Pengguna (*User*):

Kebutuhan pengguna dalam sistem ini mengacu pada pihak-pihak yang akan berinteraksi langsung dengan aplikasi sistem informasi karyawan dan presensi berbasis *android*. Berdasarkan hasil analisis kebutuhan, pengguna sistem terdiri dari tiga kategori utama, yaitu karyawan, *admin*/staf administrasi, dan pimpinan perusahaan.

Karyawan merupakan pengguna akhir (*end user*) yang memanfaatkan sistem untuk melakukan presensi melalui pemindaian *QR Code* menggunakan perangkat *smartphone*. Karyawan membutuhkan sistem yang mudah digunakan, cepat, dan praktis agar proses presensi tidak mengganggu aktivitas kerja.

Admin atau staf administrasi merupakan pengguna yang bertugas mengelola data karyawan, memantau data presensi, serta menyusun laporan kehadiran. Pengguna ini membutuhkan sistem yang mampu menyimpan data secara otomatis, terstruktur, dan mudah diproses agar pekerjaan administrasi menjadi lebih efisien.

Pimpinan perusahaan merupakan pengguna yang membutuhkan akses terhadap laporan presensi dan informasi karyawan secara cepat, akurat, dan *real-time*. Informasi tersebut digunakan sebagai dasar untuk *monitoring* kedisiplinan, evaluasi kinerja, dan pengambilan keputusan manajerial.

b. Kebutuhan Data:

Dalam pengembangan aplikasi sistem informasi karyawan dan presensi, sistem membutuhkan beberapa jenis data yang mendukung operasional aplikasi. Data yang dibutuhkan meliputi:

1. Data identitas karyawan. Digunakan sebagai informasi utama pegawai, meliputi nama, NIK/NIP, jabatan, divisi, status kepegawaian, dan informasi pendukung lainnya.
2. Data akun pengguna. Digunakan untuk proses autentikasi *login* ke dalam sistem dan pembatasan hak akses sesuai peran pengguna.
3. Data *QR Code* karyawan. Digunakan sebagai identitas unik dalam proses presensi sehingga setiap karyawan memiliki kode yang berbeda.
4. Data presensi karyawan. Meliputi data tanggal, jam masuk, jam pulang, status kehadiran, keterlambatan, izin, cuti, dan informasi kehadiran lainnya.
5. Data foto/*selfie* presensi. Digunakan sebagai *validasi* tambahan dalam proses kehadiran jika sistem menerapkannya.
6. Data lokasi presensi (*GPS*). Digunakan untuk mencatat posisi pengguna pada saat presensi sebagai bentuk *validasi* lokasi.
7. Data laporan kehadiran. Digunakan untuk menampilkan rekapitulasi kehadiran per hari, per bulan, dan per karyawan.

Sumber data diperoleh dari *input admin*, data perusahaan, serta data yang dihasilkan langsung dari proses presensi karyawan melalui aplikasi. Pengelolaan data ini bertujuan untuk memastikan bahwa sistem dapat menyajikan informasi yang akurat, lengkap, dan *real-time*.

c. Kebutuhan Fungsional:

Kebutuhan fungsional merupakan fungsi-fungsi utama yang harus tersedia dalam sistem agar dapat berjalan sesuai tujuan penelitian. Adapun fungsi yang harus ada dalam aplikasi ini meliputi:

1. Fitur *login* pengguna sesuai hak akses.
2. Fitur pengelolaan data karyawan.
3. Fitur pembuatan dan penyimpanan *QR Code* karyawan.
4. Fitur pemindaian *QR Code* untuk proses presensi.

5. Fitur *validasi* presensi dengan foto/*selfie*.
6. Fitur pencatatan lokasi (*GPS*) saat presensi.
7. Fitur penyimpanan data presensi secara *real-time*.
8. Fitur pengelolaan laporan presensi oleh *admin*.
9. Fitur akses laporan presensi secara *real-time* bagi pimpinan.
10. Fitur *logout* untuk keamanan akun pengguna.

Fungsi-fungsi tersebut dirancang untuk mendukung kemudahan penggunaan sistem, keakuratan data, serta efektivitas pengelolaan data karyawan dan kehadiran.

d. Kebutuhan Non-Fungsional:

Kebutuhan non-fungsional berkaitan dengan kualitas sistem yang harus dipenuhi agar sistem dapat berjalan secara optimal. Beberapa kebutuhan non-fungsional dalam aplikasi ini antara lain:

- 1 **Keamanan (*Security*)**
Sistem harus mampu melindungi data karyawan dan data presensi dari akses pihak yang tidak berwenang.
- 2 **Performa (*Performance*)**
Aplikasi harus berjalan dengan cepat, ringan, dan responsif pada perangkat *android*.
- 3 **Kemudahan Penggunaan (*Usability*)**
Antarmuka aplikasi harus sederhana, jelas, dan mudah dipahami oleh karyawan maupun *admin*.
- 4 **Keandalan (*Reliability*)**
Sistem harus stabil dan mampu menyimpan data presensi secara konsisten tanpa kehilangan data.
- 5 **Ketersediaan Data Secara *Real-Time***
Sistem harus mampu menampilkan data yang selalu diperbarui agar laporan dapat diakses saat dibutuhkan.
- 6 **Kompatibilitas (*Compatibility*)**
Aplikasi harus dapat berjalan pada perangkat *android* dengan spesifikasi minimum tertentu.

e. Analisa Perangkat Keras & Lunak (*Hardware/Software*):

Agar aplikasi dapat berjalan dengan baik, diperlukan spesifikasi perangkat keras dan perangkat lunak minimum sebagai berikut:

1. Perangkat Keras (*Hardware*)

Untuk Pengguna (*User*):

- a. *Smartphone* berbasis *android*
- b. *OS android 12*
- c. *RAM* minimal 4 *GB*
- d. Penyimpanan internal minimal 32 *GB*
- e. Koneksi internet stabil
- f. Kamera untuk *scan QR Code* dan *selfie*
- g. *GPS* aktif

Untuk Pengembang (*Developer*):

- a. *Laptop/PC* dengan prosesor minimal *Intel Core* Pentium atau setara
- b. *RAM* minimal 4 *GB*
- c. Penyimpanan minimal 256 *GB*
- d. Koneksi internet

2. Perangkat Lunak (*Software*)

- a. Sistem Operasi *Windows 10* atau lebih tinggi (untuk pengembangan)
- b. *Visual Studio Code* sebagai *tools* pengembangan aplikasi
- c. *Database* menggunakan *Firebase*
- d. *Android OS* minimal versi 8.0 (*Oreo*) untuk pengguna
- e. *Framework* Flutter
- f. Bahasa Pemrograman Dart
- g. *Emulator android* atau perangkat *android* fisik

Dengan analisa kebutuhan perangkat keras dan perangkat lunak ini, diharapkan sistem dapat dikembangkan dan diimplementasikan secara optimal sesuai dengan tujuan penelitian.

4.2 Perancangan Sistem (Desain) yang diusulkan

Perancangan sistem yang diusulkan merupakan tahapan lanjutan setelah dilakukan analisa terhadap sistem yang berjalan di PT Kendi Arindo. Pada tahap

ini, peneliti merancang solusi sistem yang mampu mengatasi permasalahan yang telah diidentifikasi sebelumnya, yaitu belum terintegrasinya sistem informasi karyawan dengan sistem presensi serta belum tersedianya laporan kehadiran yang dapat diakses secara *real-time*.

Sistem yang diusulkan berupa aplikasi *mobile* berbasis *android* yang dirancang untuk mempermudah proses pengelolaan data karyawan dan presensi menggunakan teknologi *QR Code*. Melalui sistem ini, proses pencatatan kehadiran dapat dilakukan dengan cara pemindaian *QR Code* menggunakan *smartphone*, sehingga lebih praktis, cepat, dan terintegrasi dengan basis data secara langsung. Selain itu, sistem juga menyediakan laporan presensi yang dapat diakses secara *real-time* oleh pihak yang berwenang.

Perancangan sistem ini dilakukan dengan pendekatan *design thinking*, yang berfokus pada kebutuhan dan kenyamanan pengguna, baik dari sisi karyawan sebagai pengguna presensi, maupun pihak *HRD* dan pimpinan sebagai pengelola dan pemantau data. Dengan pendekatan ini, sistem diharapkan dapat memberikan solusi yang sesuai dengan kondisi nyata di lapangan. Perancangan sistem dibagi menjadi dua bagian utama, yaitu perancangan sistem secara logis dan perancangan sistem secara fisik. Perancangan logis menggambarkan bagaimana sistem bekerja secara konseptual, sedangkan perancangan fisik menjelaskan implementasi teknis sistem yang akan dibangun.

4.2.1 Perancangan Sistem (Logis):

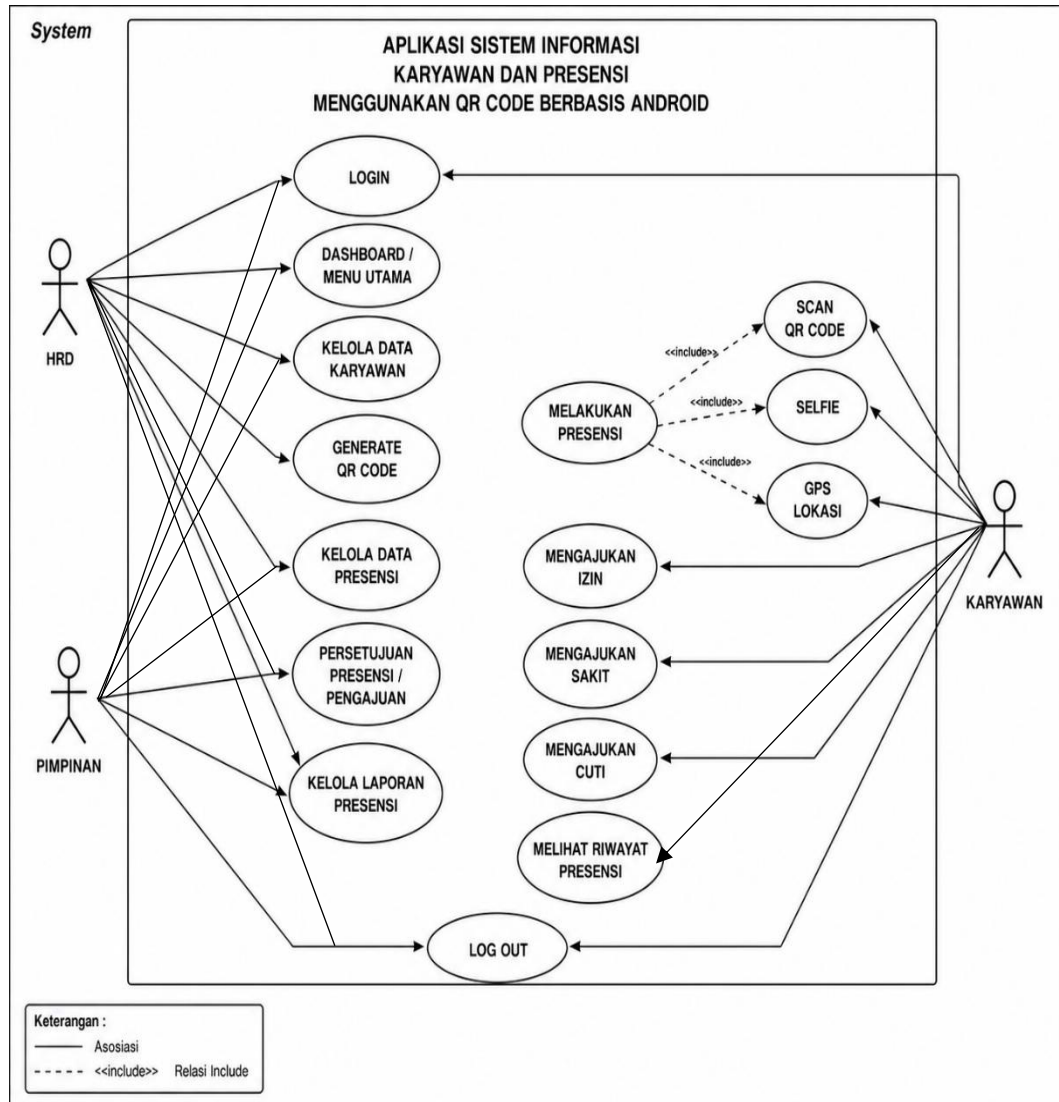
Perancangan sistem secara logis bertujuan untuk menggambarkan hubungan antara pengguna dengan sistem serta fungsi-fungsi yang tersedia dalam aplikasi tanpa membahas aspek teknis secara detail. Pada tahap ini digunakan *model* pemodelan *Unified Modeling Language (UML)* untuk merepresentasikan rancangan sistem.

a. Use Case Diagram:

Use case diagram merupakan gambaran dari fungsionalitas suatu sistem yang menunjukkan interaksi antara aktor dengan sistem. Diagram ini membantu pengguna dalam memahami bagaimana sistem bekerja dan fitur apa saja yang tersedia dalam aplikasi yang dibangun. Berikut adalah gambar *Use Case Diagram*

Aplikasi Sistem Informasi Karyawan dan presensi dapat dilihat pada gambar 4.1 berikut:

1. Use Case Diagram Utama



Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 1 Use Case Diagram Utama

Use case diagram Sistem Informasi Karyawan dan Presensi menggunakan QR Code berbasis android di atas menggambarkan interaksi antara tiga aktor utama, yaitu HRD, Pimpinan, dan Karyawan dengan sistem yang dibangun. Diagram ini menunjukkan hak akses dan fungsi yang dapat digunakan oleh masing-masing aktor sesuai dengan perannya dalam perusahaan.

HRD berperan sebagai administrator sistem yang memiliki hak akses penuh terhadap pengelolaan data dan proses presensi. HRD dapat melakukan login,

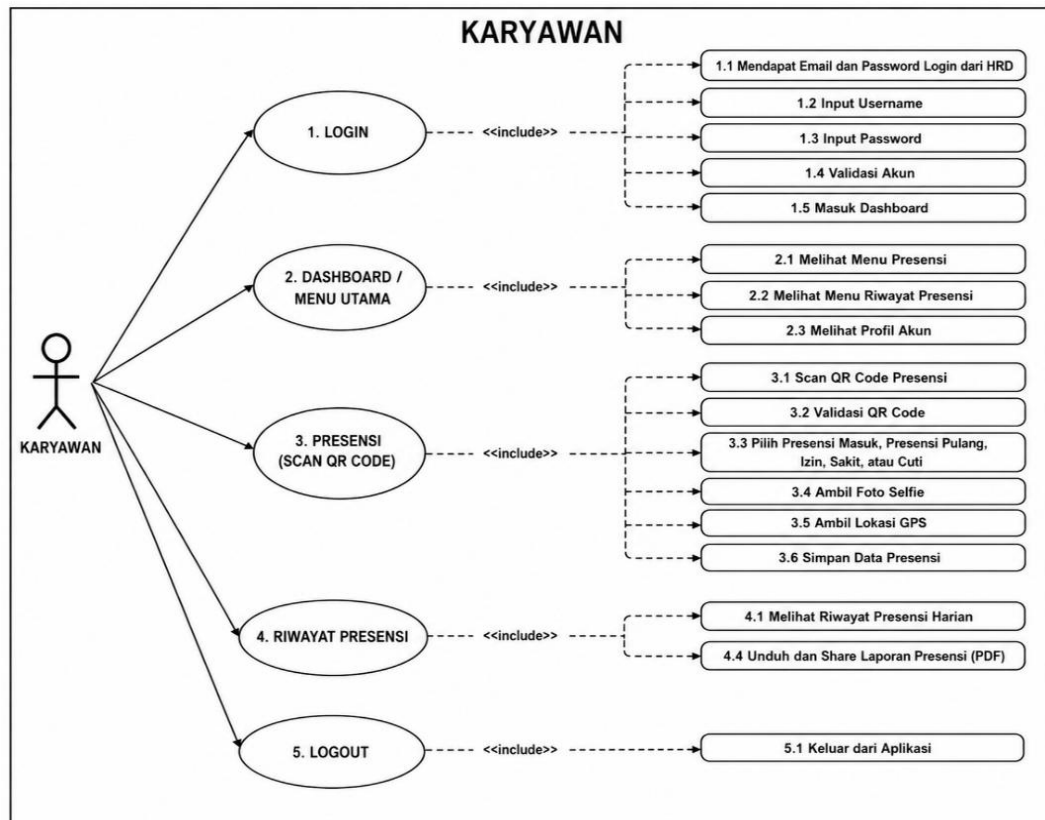
mengakses menu utama, mengelola data karyawan, *generate QR Code*, mengelola data presensi, melakukan persetujuan pengajuan izin, sakit, dan cuti, serta melihat laporan presensi dan rekap kehadiran karyawan. Setelah selesai menggunakan sistem, *HRD* dapat melakukan *logout*.

Pimpinan berperan dalam pengawasan dan pengambilan keputusan terkait kehadiran karyawan. Pimpinan dapat melakukan *login*, mengakses menu utama, melihat laporan presensi dan rekap kehadiran, serta memberikan persetujuan pengajuan cuti yang telah diproses oleh *HRD*. Setelah selesai menggunakan sistem, pimpinan dapat melakukan *logout*.

Karyawan berperan sebagai pengguna utama dalam proses presensi. Karyawan dapat melakukan *login*, mengakses menu utama, melakukan presensi menggunakan *QR Code*, melakukan *selfie* presensi, serta validasi lokasi *GPS* sebagai verifikasi kehadiran. Selain itu, karyawan juga dapat melihat riwayat presensi dan mengajukan izin, sakit, atau cuti melalui sistem. Setelah selesai menggunakan aplikasi, karyawan dapat melakukan *logout*.

Berdasarkan *use case diagram* tersebut, sistem yang dibangun mampu mengintegrasikan pengelolaan data karyawan, presensi berbasis *QR Code*, validasi *selfie* dan *GPS*, pengajuan serta persetujuan izin, sakit, dan cuti, hingga pelaporan kehadiran dalam satu aplikasi yang terintegrasi sehingga dapat mendukung pengelolaan kehadiran karyawan secara lebih efektif dan efisien.

2. Use Case Diagram Karyawan



Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 2 Use Case Diagram Karyawan

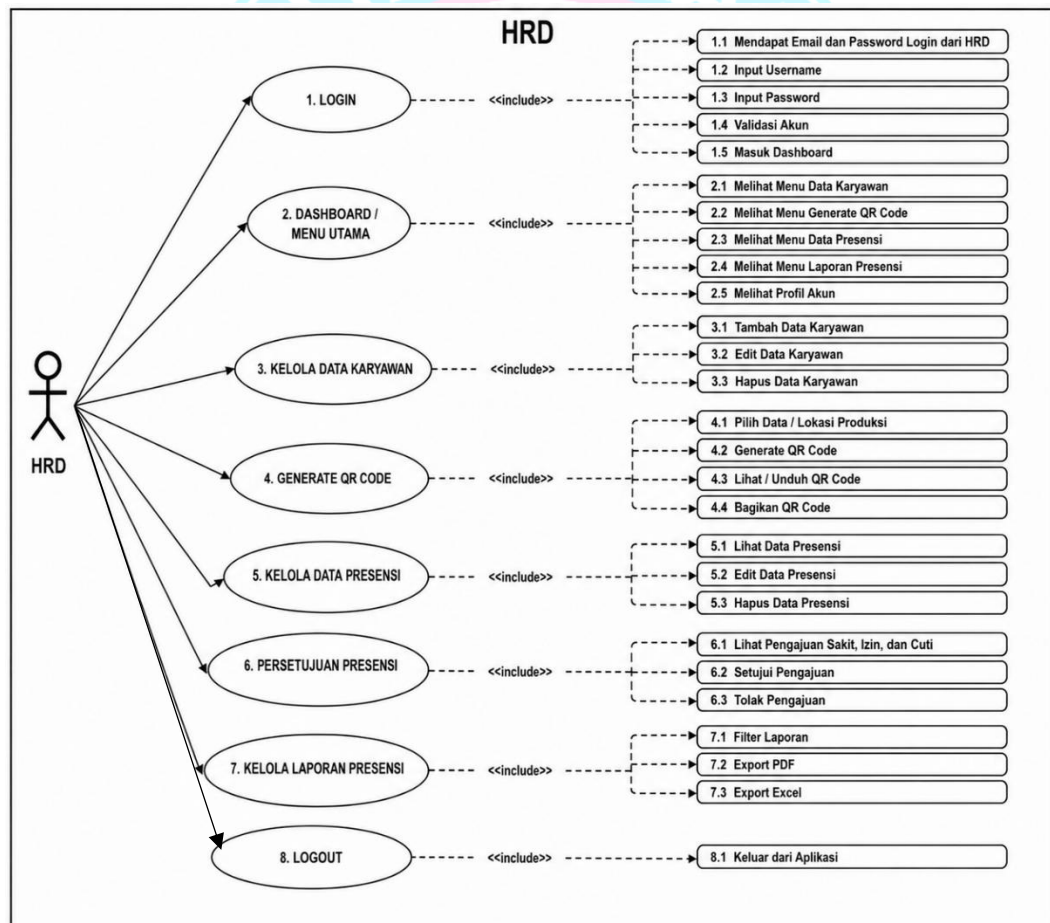
Use case diagram tersebut menggambarkan peran aktor karyawan dalam menggunakan sistem digitalisasi informasi karyawan dan presensi berbasis *mobile*. Karyawan memiliki akses untuk melakukan presensi masuk, presensi pulang, melihat riwayat presensi, serta mengakses informasi akun melalui aplikasi. Sebelum menggunakan sistem, karyawan harus melakukan proses *login* menggunakan *username* dan *password* yang telah diberikan oleh *hrd*. Setelah akun berhasil divalidasi, karyawan dapat masuk ke *dashboard* atau menu utama aplikasi.

Pada *dashboard*, karyawan dapat melihat menu presensi, menu riwayat presensi, serta profil akun. Dalam fitur presensi masuk, karyawan melakukan *scan qr code* presensi yang telah disediakan perusahaan. Setelah *qr code* berhasil dipindai dan divalidasi, sistem akan mengambil foto *selfie* dan lokasi *gps* sebagai bukti kehadiran, kemudian menyimpan data presensi masuk secara otomatis ke dalam sistem.

Selain presensi masuk, sistem juga menyediakan fitur presensi pulang dengan mekanisme yang hampir sama. Karyawan harus melakukan *scan qr code* presensi, validasi *qr code*, mengambil foto *selfie*, serta mengambil lokasi *gps* sebelum data presensi pulang disimpan ke dalam sistem. Fitur ini bertujuan untuk memastikan keakuratan data kehadiran karyawan.

Pada menu riwayat presensi, karyawan dapat melihat riwayat presensi harian, melihat detail presensi, serta melakukan filter riwayat presensi berdasarkan kebutuhan tertentu. Setelah selesai menggunakan aplikasi, karyawan dapat keluar dari sistem melalui fitur *logout* yang berfungsi untuk mengakhiri sesi penggunaan aplikasi dengan aman.

3. Use Case Diagram HRD



Sumber: data diolah oleh penulis, 2026

Gambar 4. 3 Use Case Diagram HRD

Use case diagram tersebut menggambarkan peran aktor *hrd* dalam menggunakan sistem informasi karyawan dan presensi berbasis *android*. *Hrd*

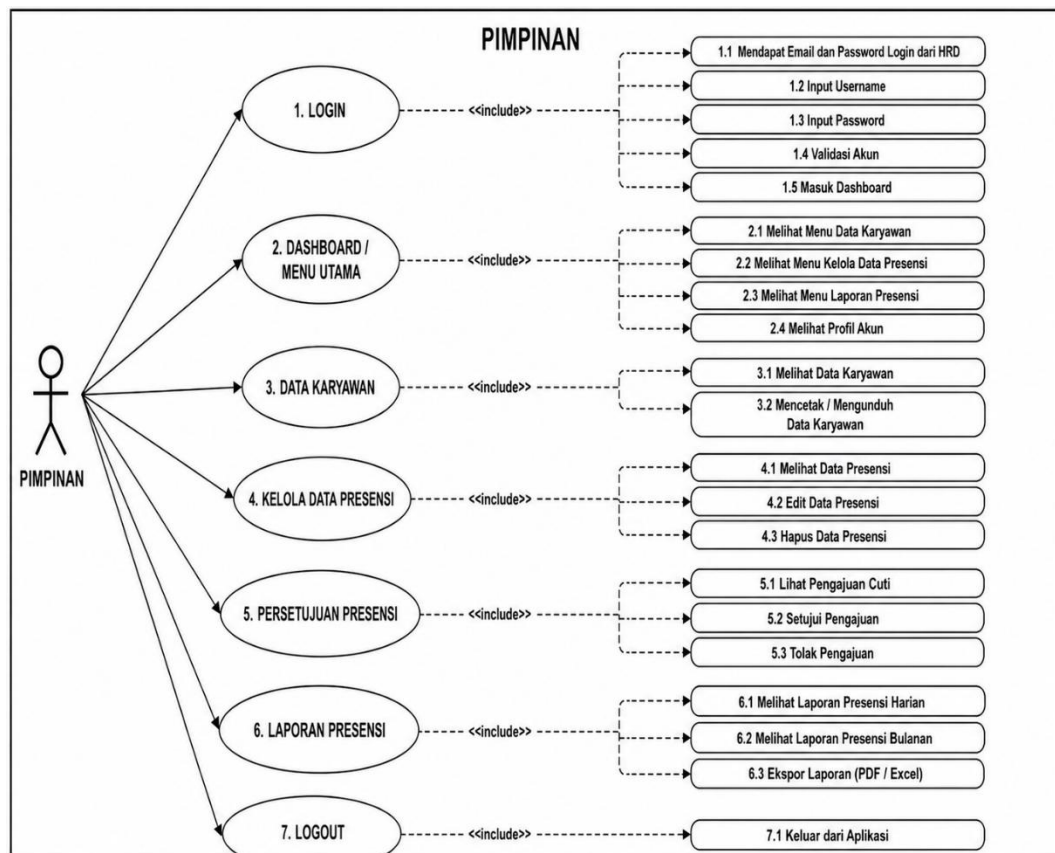
memiliki hak akses utama dalam mengelola data karyawan, *QR Code*, data presensi, persetujuan pengajuan, serta laporan presensi. Sebelum mengakses sistem, *hrd* harus melakukan *login* menggunakan *username* dan *password* yang telah terdaftar. Setelah data *login* berhasil divalidasi, *hrd* dapat masuk ke *dashboard* atau menu utama aplikasi.

Pada *dashboard*, *hrd* dapat melihat berbagai menu utama seperti data karyawan, *generate QR Code*, data presensi, laporan presensi, dan profil akun. Melalui fitur kelola data karyawan, *hrd* dapat menambahkan, mengubah, dan menghapus data karyawan sesuai kebutuhan perusahaan.

Sistem juga menyediakan fitur *generate QR Code* yang digunakan *hrd* untuk memilih data atau lokasi produksi, membuat *QR Code*, melihat atau mengunduh *QR Code*, serta membagikannya kepada karyawan untuk keperluan presensi. Selain itu, *hrd* dapat mengelola data presensi dengan melihat, mengubah, dan menghapus data presensi karyawan yang tersimpan dalam sistem. Pada fitur persetujuan presensi, *hrd* dapat melihat pengajuan sakit, izin, dan cuti yang diajukan oleh karyawan.

Hrd memiliki wewenang untuk menyetujui atau menolak pengajuan tersebut sesuai dengan ketentuan perusahaan. Selanjutnya, pada fitur laporan presensi, *hrd* dapat melakukan *filter* laporan serta mengekspor laporan ke dalam format *pdf* maupun *excel* untuk kebutuhan administrasi dan dokumentasi perusahaan. Setelah seluruh aktivitas selesai dilakukan, *hrd* dapat keluar dari aplikasi melalui fitur *logout* yang berfungsi untuk mengakhiri sesi penggunaan aplikasi dengan aman.

4. Use Case Diagram Pimpinan



Sumber: data diolah oleh penulis, 2026

Gambar 4. 4 Use Case Diagram Pimpinan

Use case diagram pimpinan diatas menggambarkan peran aktor Pimpinan dalam menggunakan sistem informasi karyawan dan presensi berbasis *android*. Pimpinan memiliki akses untuk masuk ke dalam sistem melalui proses *login* menggunakan akun yang telah diberikan oleh *HRD*. Setelah akun berhasil divalidasi, pimpinan dapat masuk ke *dashboard* atau menu utama aplikasi untuk mengakses berbagai fitur yang tersedia. Pada *dashboard*, pimpinan dapat melihat menu data karyawan, kelola data presensi, laporan presensi, serta profil akun.

Melalui fitur data karyawan, pimpinan dapat melihat informasi data karyawan serta mencetak atau mengunduh data tersebut apabila diperlukan. Selain itu, pimpinan juga memiliki akses pada fitur kelola data presensi yang digunakan untuk melihat, mengubah, dan menghapus data presensi karyawan sesuai kebutuhan pengelolaan data perusahaan. Pimpinan juga memiliki wewenang untuk melakukan persetujuan pengajuan cuti yang telah diajukan oleh karyawan dan diproses oleh

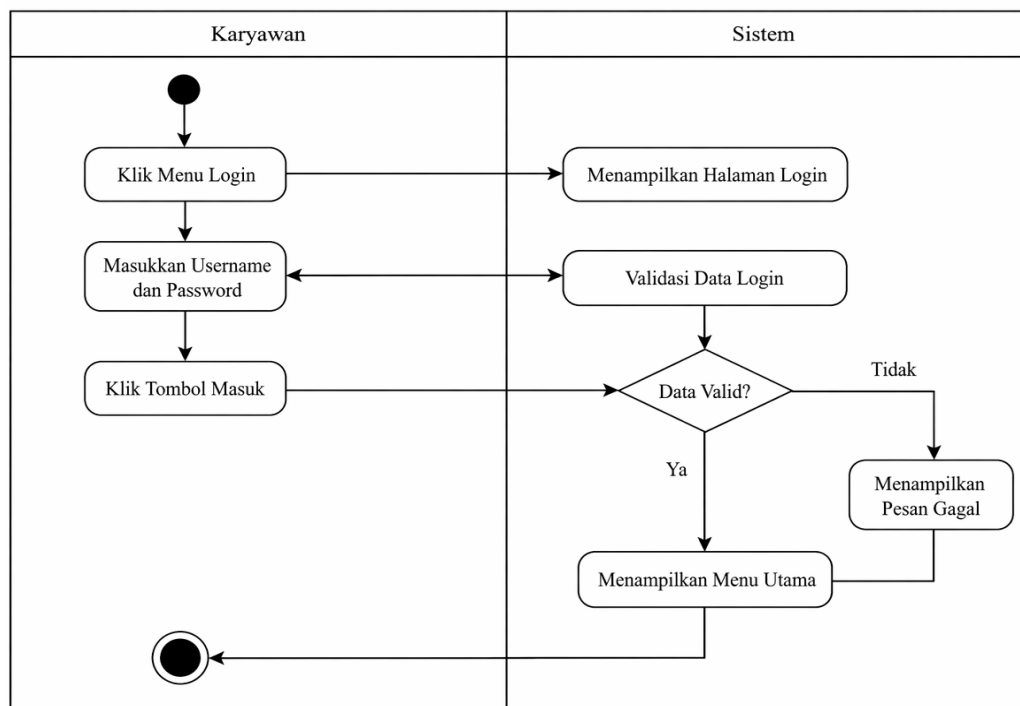
HRD. Pada fitur ini, pimpinan dapat melihat pengajuan cuti, memberikan persetujuan, atau menolak pengajuan sesuai kebijakan perusahaan.

Sistem menyediakan fitur laporan presensi yang memungkinkan pimpinan melakukan penyaringan (*filter*) laporan berdasarkan kebutuhan tertentu serta mengekspor laporan ke dalam format *pdf* maupun *excel*. Fitur ini memudahkan pimpinan dalam melakukan *monitoring* kehadiran dan penyusunan laporan administrasi perusahaan. Setelah selesai menggunakan aplikasi, pimpinan dapat keluar dari sistem melalui fitur *logout* untuk mengakhiri sesi penggunaan aplikasi dengan aman.

b. Activity Diagram

Activity diagram menyediakan gambaran untuk memodelkan proses atau alur kerja *model* dalam suatu sistem informasi. Berikut adalah gambar *Activity Diagram* Aplikasi Sistem Informasi Karyawan dan Presensi:

1. Activity Diagram Login Karyawan yang diusulkan.



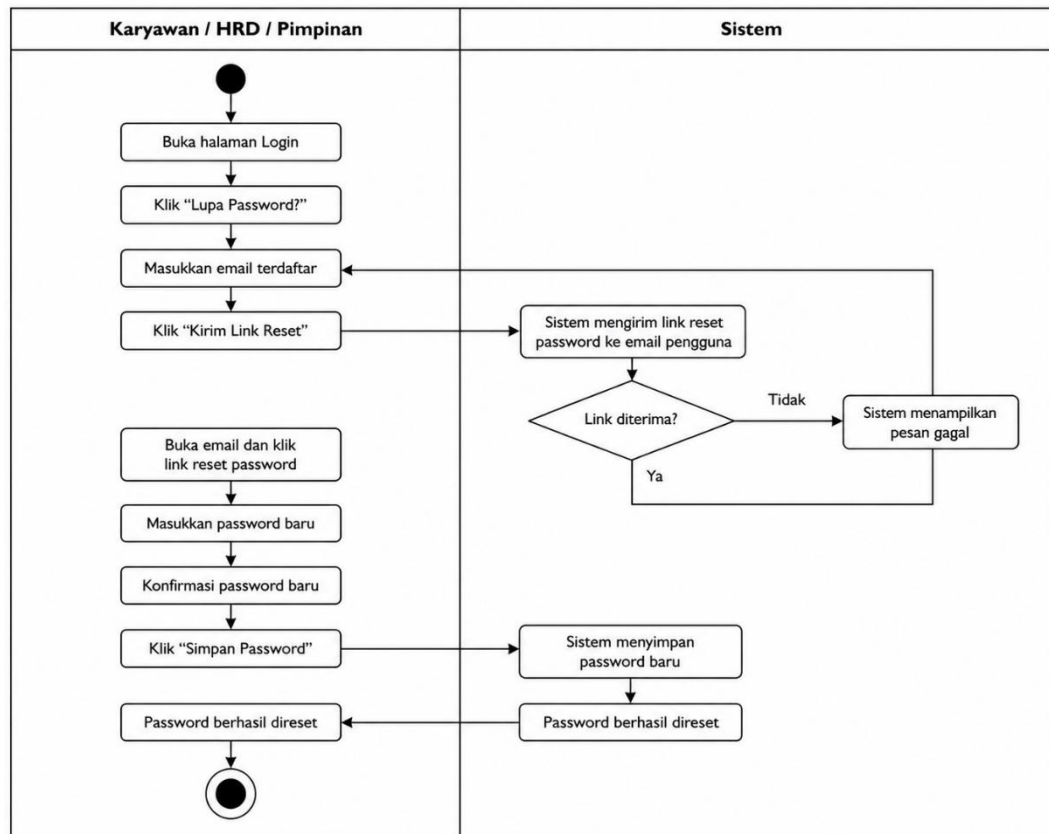
Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 5 Activity Diagram Login Karyawan yang diusulkan.

Activity diagram ini menggambarkan proses karyawan dalam melakukan *login* ke dalam sistem. Proses dimulai dari karyawan membuka menu *login*,

kemudian memasukkan *username* dan *password*. Sistem akan melakukan *validasi* data *login*. Jika data *valid*, maka karyawan akan diarahkan ke menu utama. Jika tidak *valid*, sistem akan menampilkan pesan gagal.

2. Activity Diagram Reset Password yang diusulkan.

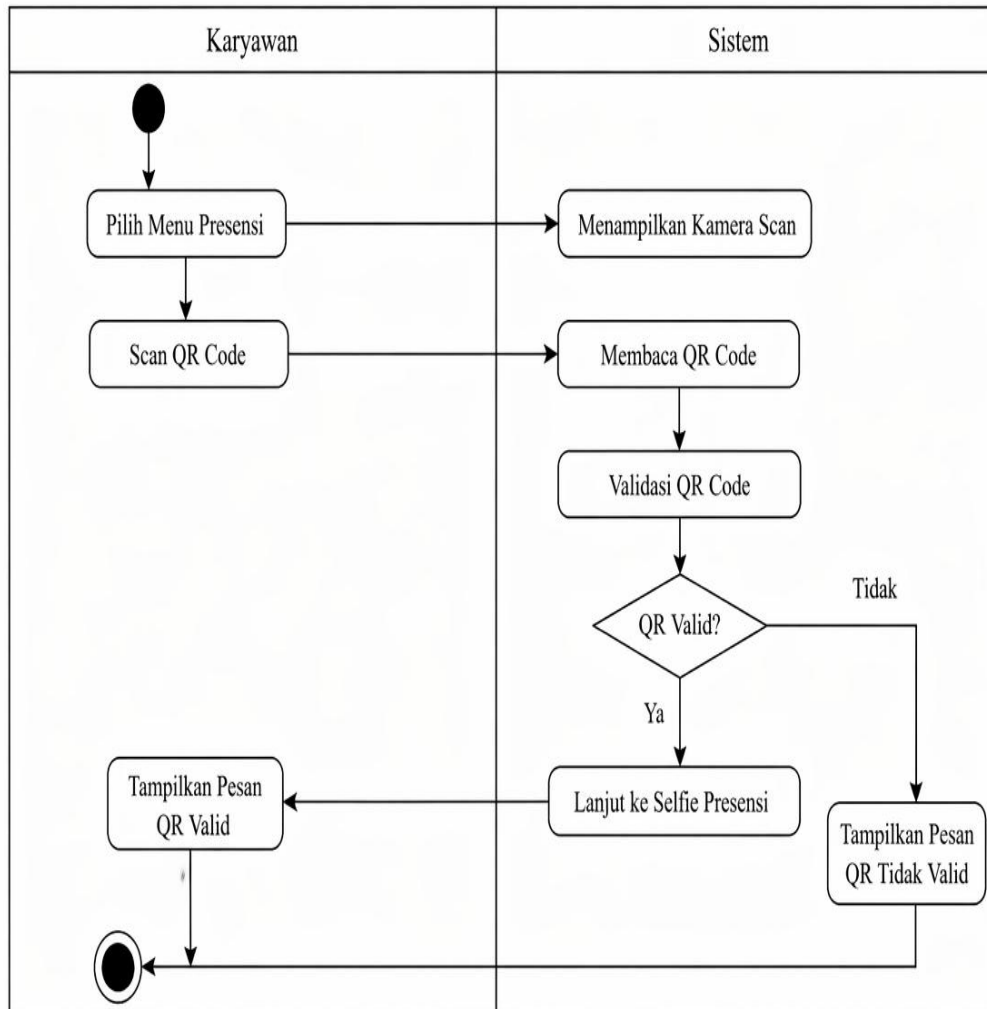


Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 6 Activity Diagram Reset Password yang diusulkan.

Activity diagram ini menggambarkan menggambarkan proses pengaturan ulang *password* yang dapat dilakukan oleh karyawan, *HRD*, dan pimpinan ketika lupa *password* akun aplikasi. Proses dimulai dari memilih menu *lupa password*, memasukkan *email* terdaftar, lalu sistem mengirimkan *link reset password* ke *email* pengguna. Selanjutnya pengguna membuka *link reset password*, membuat *password* baru, dan menyimpan perubahan *password*. Setelah proses berhasil, sistem akan menampilkan pesan bahwa *password* berhasil di *reset* sehingga pengguna dapat kembali *login* ke dalam aplikasi.

3. Activity Diagram Presensi QR Code yang diusulkan

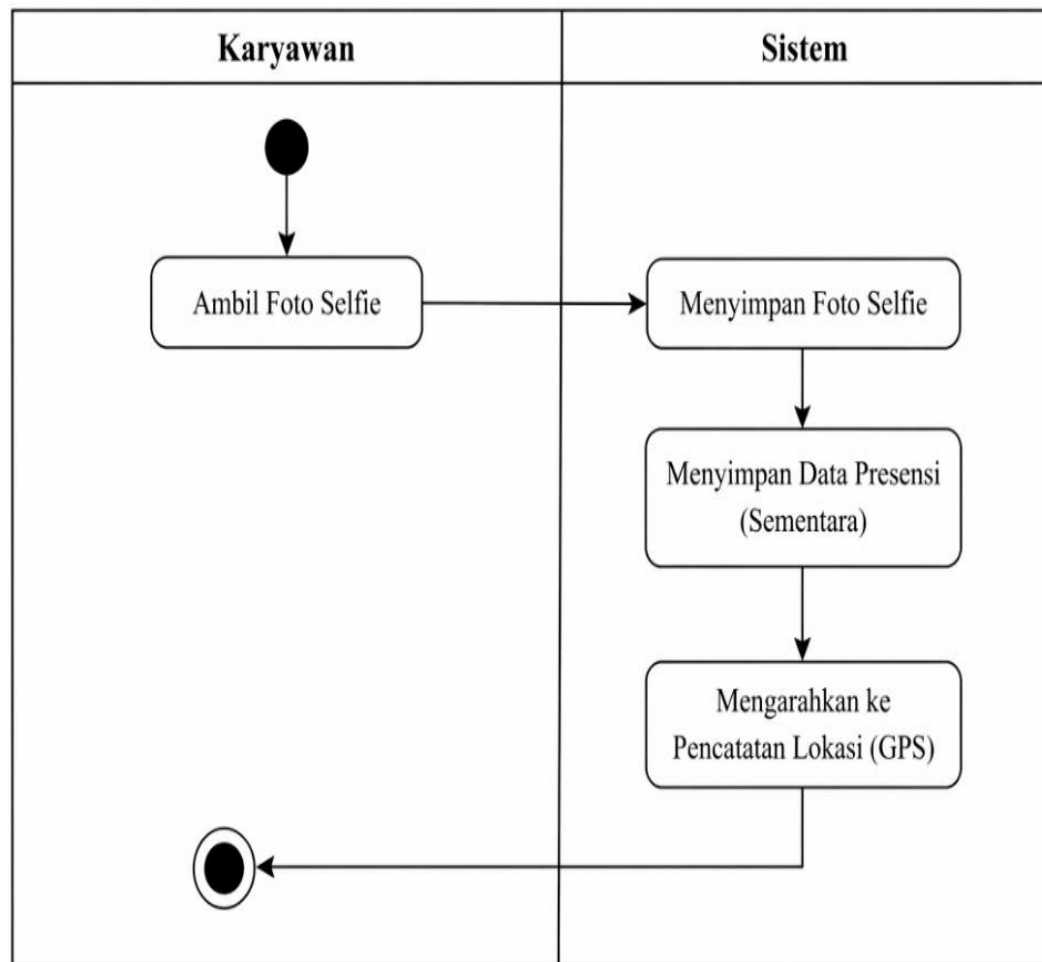


Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 7 Activity Diagram Presensi QR Code yang diusulkan

Diagram ini menggambarkan proses karyawan melakukan presensi menggunakan QR Code. Karyawan memilih menu presensi, kemudian sistem menampilkan kamera untuk melakukan scan QR Code. Setelah QR berhasil dibaca, sistem melakukan validasi. Jika valid, proses dilanjutkan ke tahap selfie presensi. Jika tidak valid, sistem akan menampilkan pesan gagal.

4. *Activity Diagram Selfie Presensi* yang diusulkan

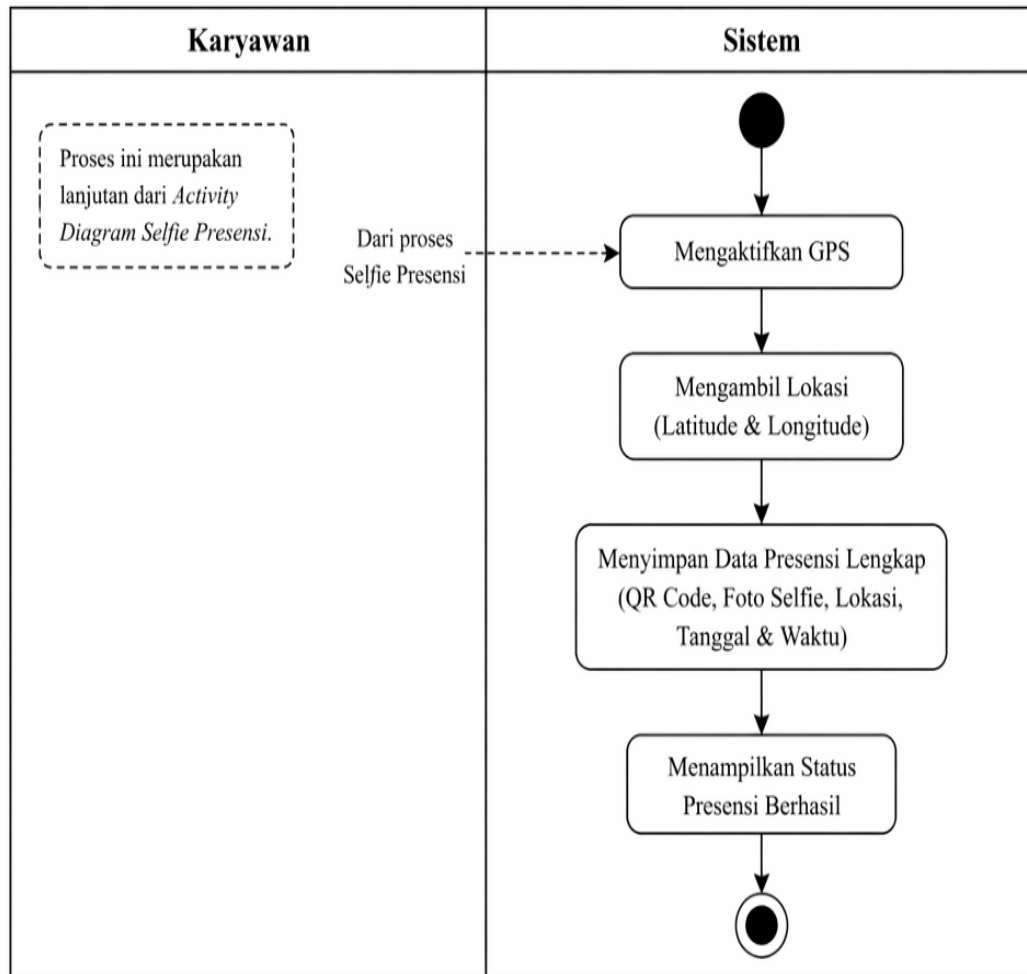


Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 8 *Activity Diagram Selfie Presensi* yang diusulkan

Diagram ini menunjukkan proses verifikasi presensi menggunakan foto *selfie*. Setelah *QR Code valid*, karyawan mengambil foto *selfie* sebagai bukti kehadiran. Sistem kemudian menyimpan foto dan data presensi secara sementara. Selanjutnya sistem mengarahkan ke proses pencatatan Lokasi (*GPS*). Pada tahap ini Presensi belum dinyatakan berhasil.

5. *Activity Diagram* Pencatatan Lokasi (*GPS*) yang diusulkan

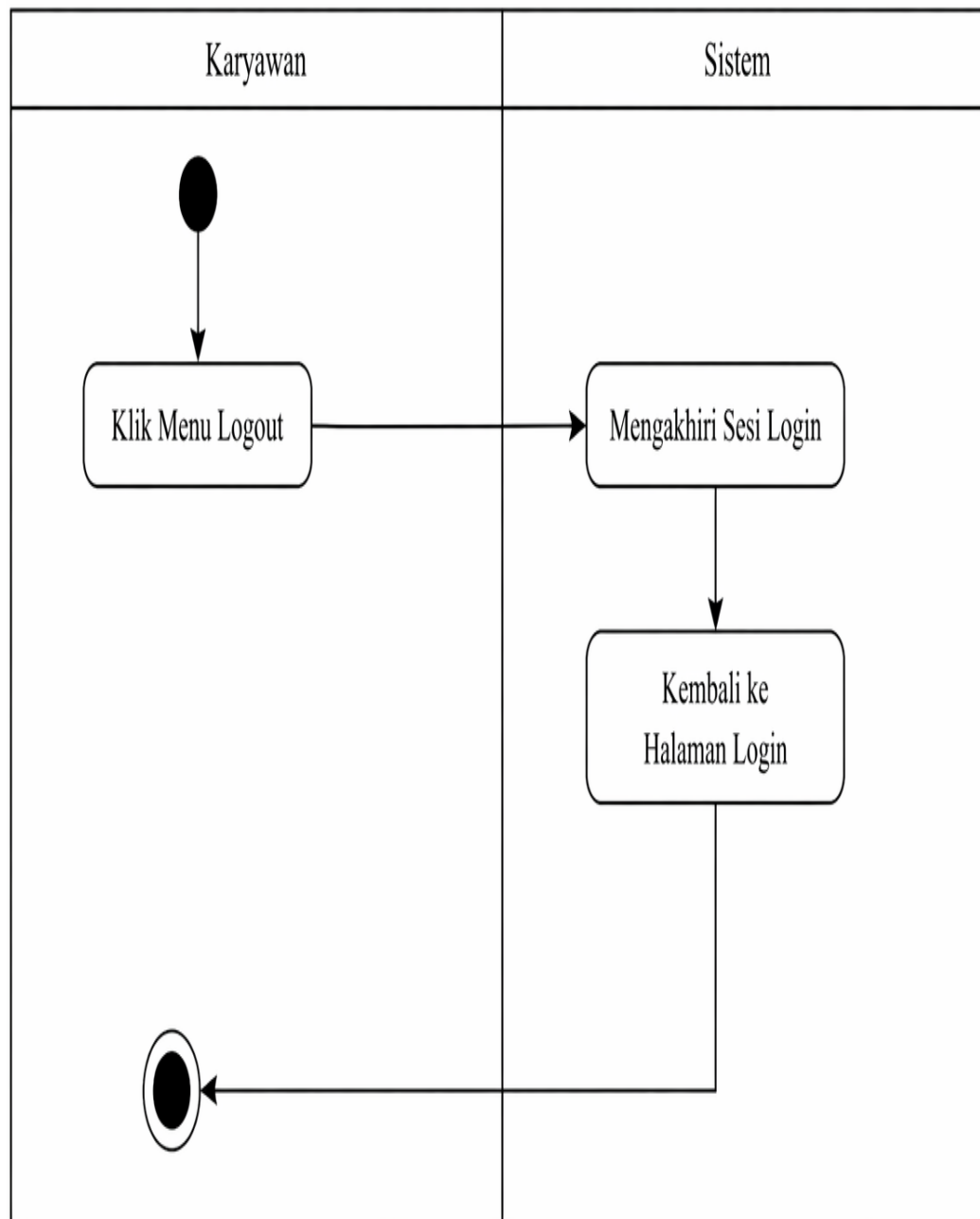


Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 9 *Activity Diagram* Pencatatan Lokasi (*GPS*) yang diusulkan

Diagram ini menunjukkan proses pencatatan Lokasi menggunakan *GPS* sebagai tahap akhir dari proses presensi. Sistem mengaktifkan *GPS*, mengambil koordinat lokasi (*Latitude* dan *Longitude*), kemudian menyimpan seluruh data presensi secara lengkap yang terdiri dari *QR Code*, foto *selfie*, lokasi, tanggal, dan waktu. Setelah seluruh data tersimpan dengan berhasil, sistem akan menampilkan status bahwa presensi berhasil dilakukan.

6. *Activity Diagram Log Out Karyawan yang diusulkan*

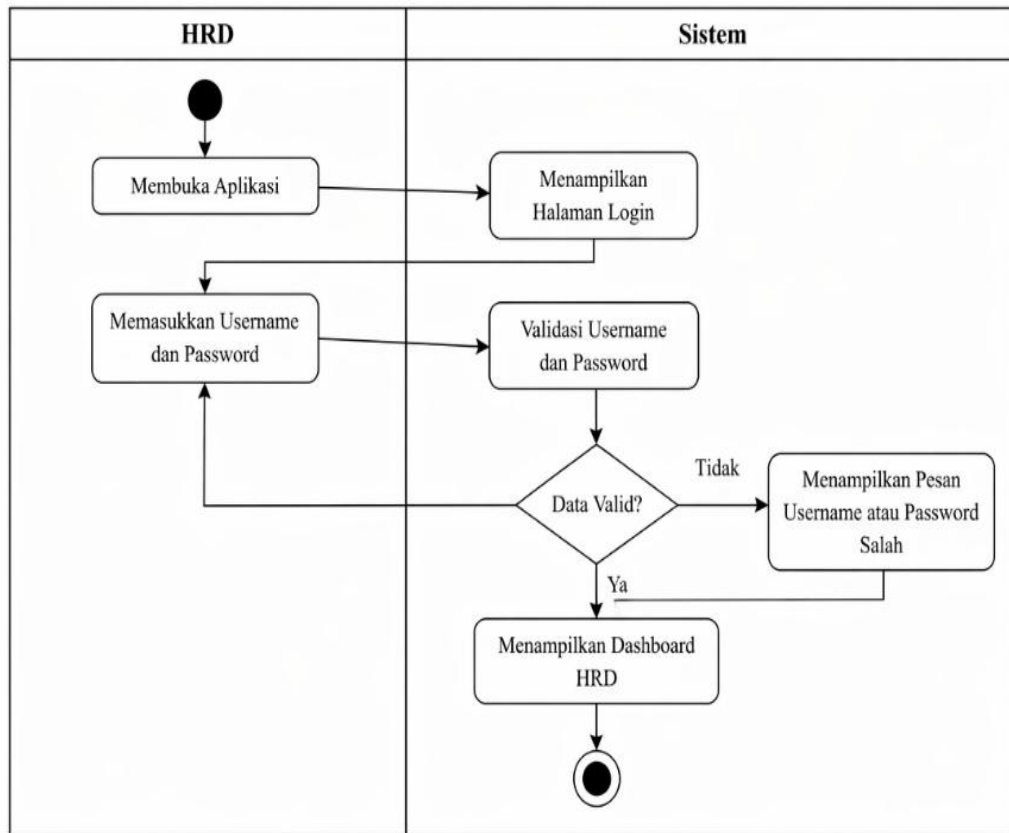


Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 10 *Activity Diagram Log Out Karyawan yang diusulkan*

Diagram ini menggambarkan proses *logout* karyawan dari sistem. Karyawan memilih menu *logout*, kemudian sistem mengakhiri sesi *login* dan mengarahkan kembali ke halaman *login*.

7. Activity Diagram Login HRD yang diusulkan

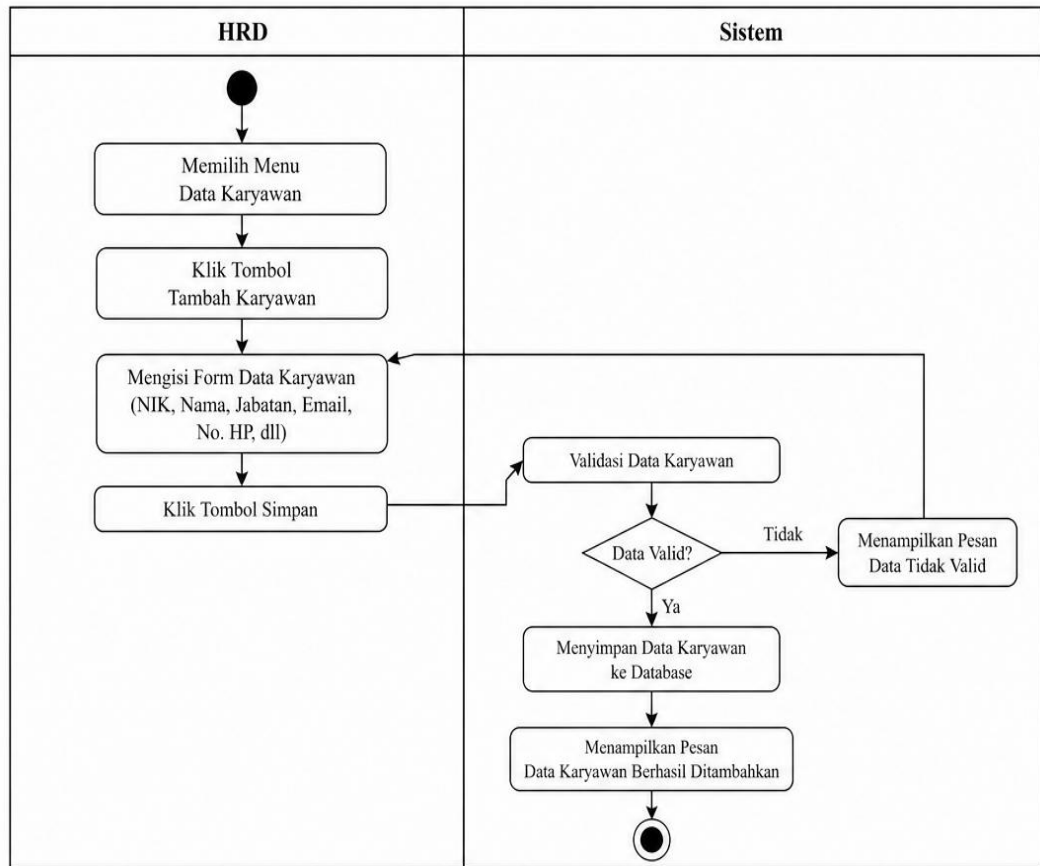


Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 11 Activity Diagram Login HRD yang diusulkan

Diagram ini menggambarkan proses *HRD* dalam melakukan *login* ke sistem. *HRD* membuka aplikasi kemudian sistem menampilkan. *HRD* membuka aplikasi, kemudian sistem akan menampilkan halaman *login*. Selanjutnya *HRD* memasukkan *username* dan *password*. Sistem akan memvalidasi data yang dimasukkan. Jika data *valid* maka, sistem akan menampilkan *dashboard HRD*. Jika data tidak *valid*, sistem akan menampilkan pesan kesalahan dan *HRD* diminta untuk memasukkan *username* dan *password* kembali.

8. *Activity Diagram Kelola Data Karyawan (Tambah)* yang diusulkan

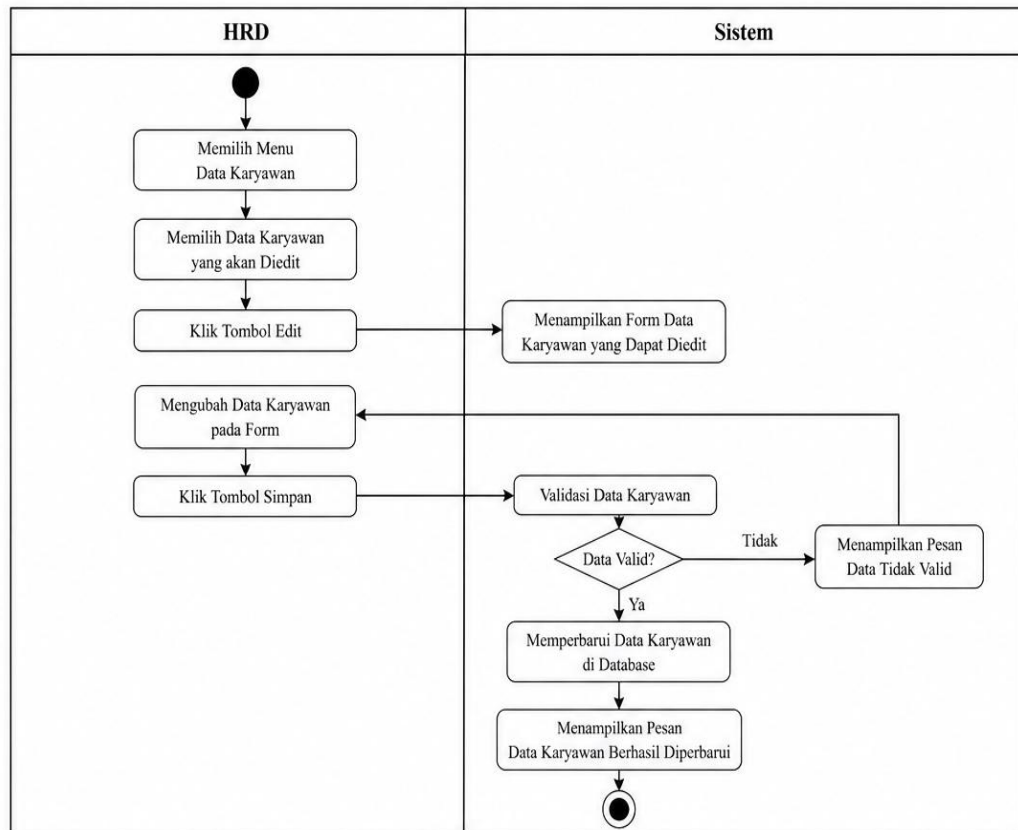


Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 12 *Activity Diagram Kelola Data Karyawan (Tambah)* yang diusulkan

Diagram ini menggambarkan proses *HRD* dalam menambahkan data karyawan ke dalam sistem. Proses dimulai ketika *HRD* memilih menu data karyawan, kemudian mengklik tombol tambah karyawan. Selanjutnya *HRD* mengisi *form* data karyawan yang meliputi NIK, nama, jabatan, *email*, nomor telepon, dan informasi lainnya yang diperlukan. Setelah data diisi, *HRD* menekan tombol simpan dan sistem akan melakukan *validasi* terhadap data yang dimasukkan. Jika data tidak *valid*, sistem akan menampilkan pesan kesalahan dan *HRD* diminta untuk memperbaiki data tersebut. Namun jika data *valid*, sistem akan menyimpan data karyawan ke dalam *database* dan menampilkan pesan bahwa data karyawan berhasil ditambahkan.

9. *Activity Diagram Kelola Data Karyawan (Edit) yang diusulkan*

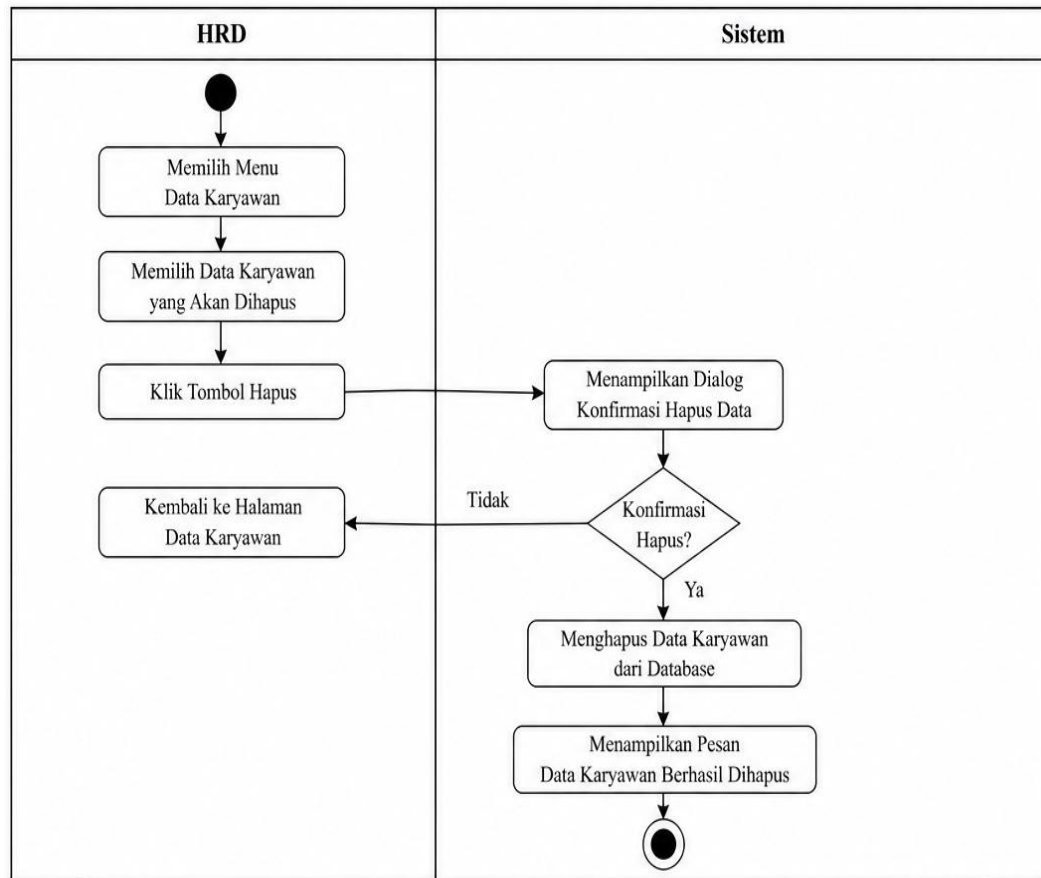


Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 13 *Activity Diagram Kelola Data Karyawan (Edit) yang diusulkan*

Diagram ini menggambarkan proses *HRD* dalam mengedit data karyawan yang telah tersedia di dalam sistem. Proses dimulai ketika *HRD* memilih menu data karyawan, kemudian memilih salah satu data karyawan yang akan diubah. Setelah itu *HRD* menekan tombol *edit*, dan sistem akan menampilkan *form* data karyawan yang dapat diperbarui. *HRD* kemudian melakukan perubahan data sesuai kebutuhan dan menekan tombol simpan. Selanjutnya sistem melakukan *validasi* terhadap data yang telah di *input*. Jika data tidak *valid*, sistem akan menampilkan pesan kesalahan dan *HRD* diminta untuk memperbaiki data tersebut. Namun jika data *valid*, sistem akan memperbarui data karyawan di dalam *database* dan menampilkan pesan bahwa data karyawan berhasil diperbarui.

10. *Activity Diagram* Kelola Data Karyawan (Hapus) yang diusulkan

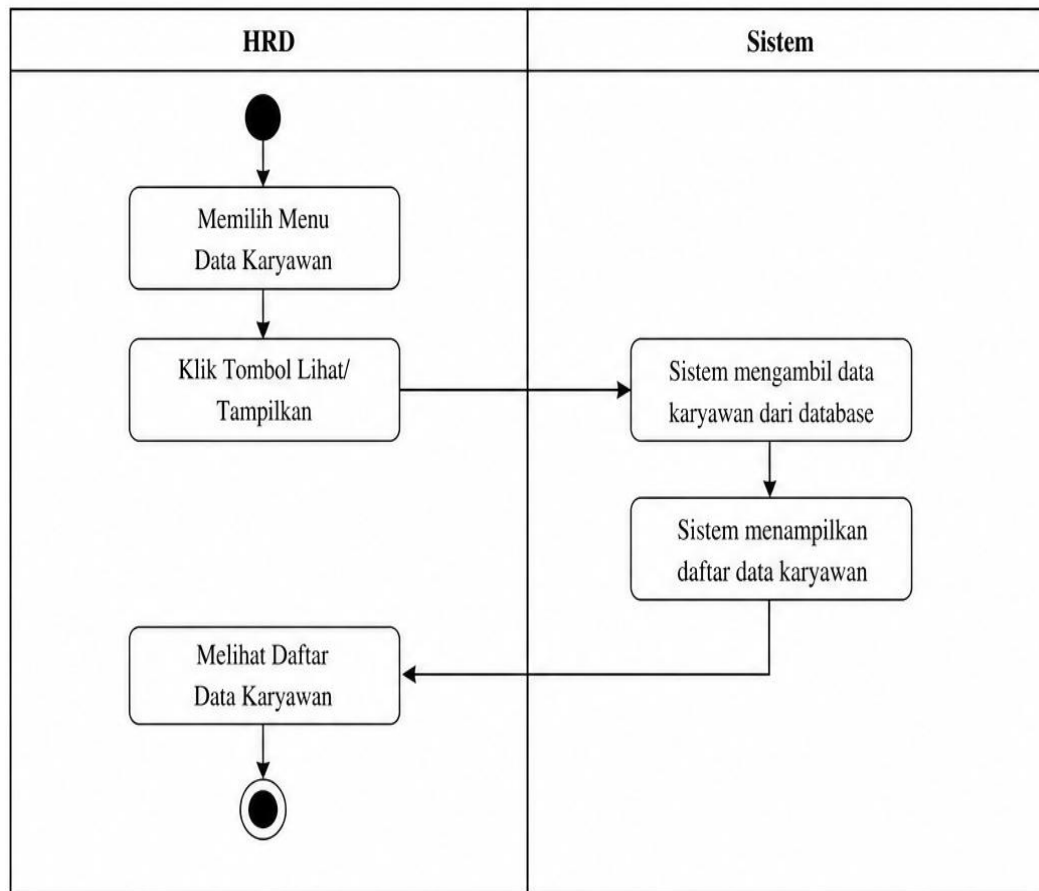


Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 14 *Activity Diagram* Kelola Data Karyawan(Hapus) yang diusulkan

Diagram ini menggambarkan proses *HRD* dalam menghapus data karyawan dari sistem. Proses dimulai ketika *HRD* memilih menu data karyawan, kemudian memilih salah satu data karyawan yang akan dihapus. Setelah itu *HRD* menekan tombol hapus, dan sistem akan menampilkan dialog konfirmasi untuk memastikan tindakan penghapusan. Jika *HRD* memilih “Tidak”, maka proses dibatalkan dan sistem akan kembali ke halaman data karyawan tanpa menghapus data. Namun jika *HRD* memilih “Ya”, sistem akan menghapus data karyawan dari *database* sesuai dengan implementasi yang digunakan, kemudian menampilkan pesan bahwa data karyawan berhasil dihapus.

11. Activity Diagram Melihat Data Karyawan

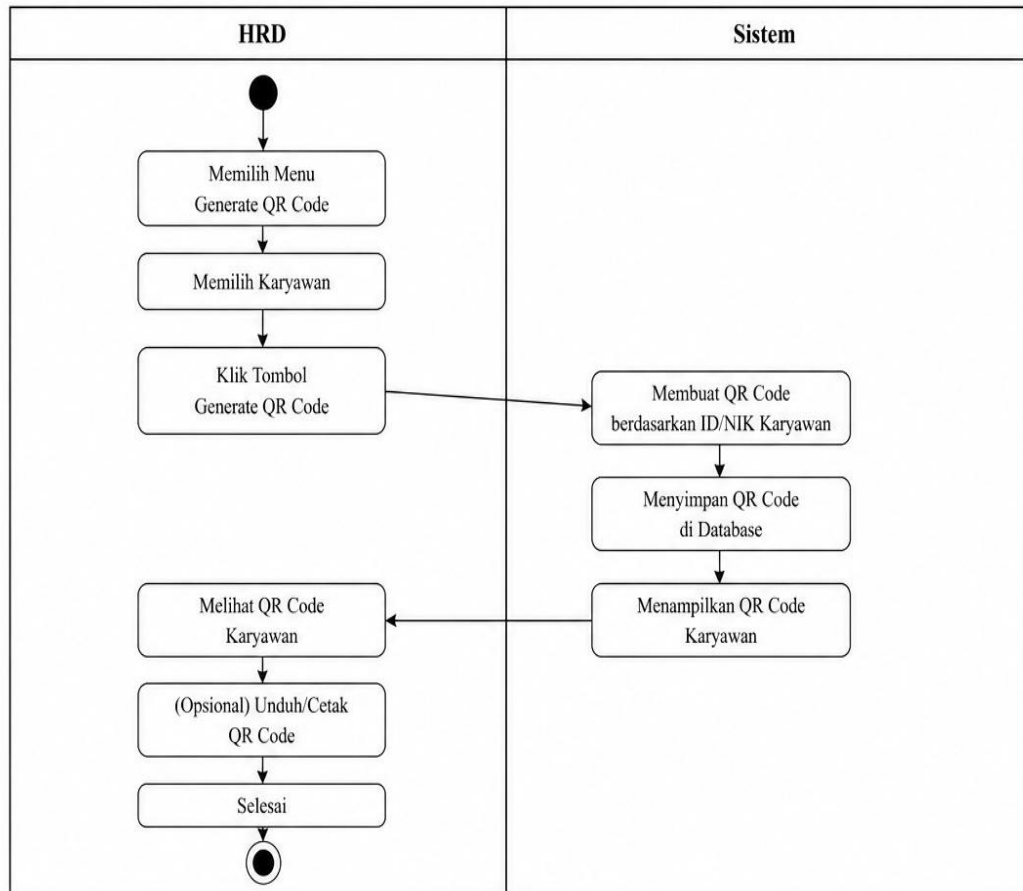


Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 15 Activity Diagram Melihat Data Karyawan

Diagram ini menggambarkan proses *HRD* dalam melihat seluruh data karyawan yang tersimpan di dalam sistem. Proses dimulai ketika *HRD* memilih menu data karyawan, kemudian menekan tombol lihat atau tampilkan. Selanjutnya sistem akan mengambil data karyawan dari *database* dan menampilkannya dalam bentuk daftar data karyawan. *HRD* kemudian dapat melihat seluruh informasi karyawan yang tersedia dalam sistem.

12. *Activity Diagram Generate QR Code yang diusulkan*

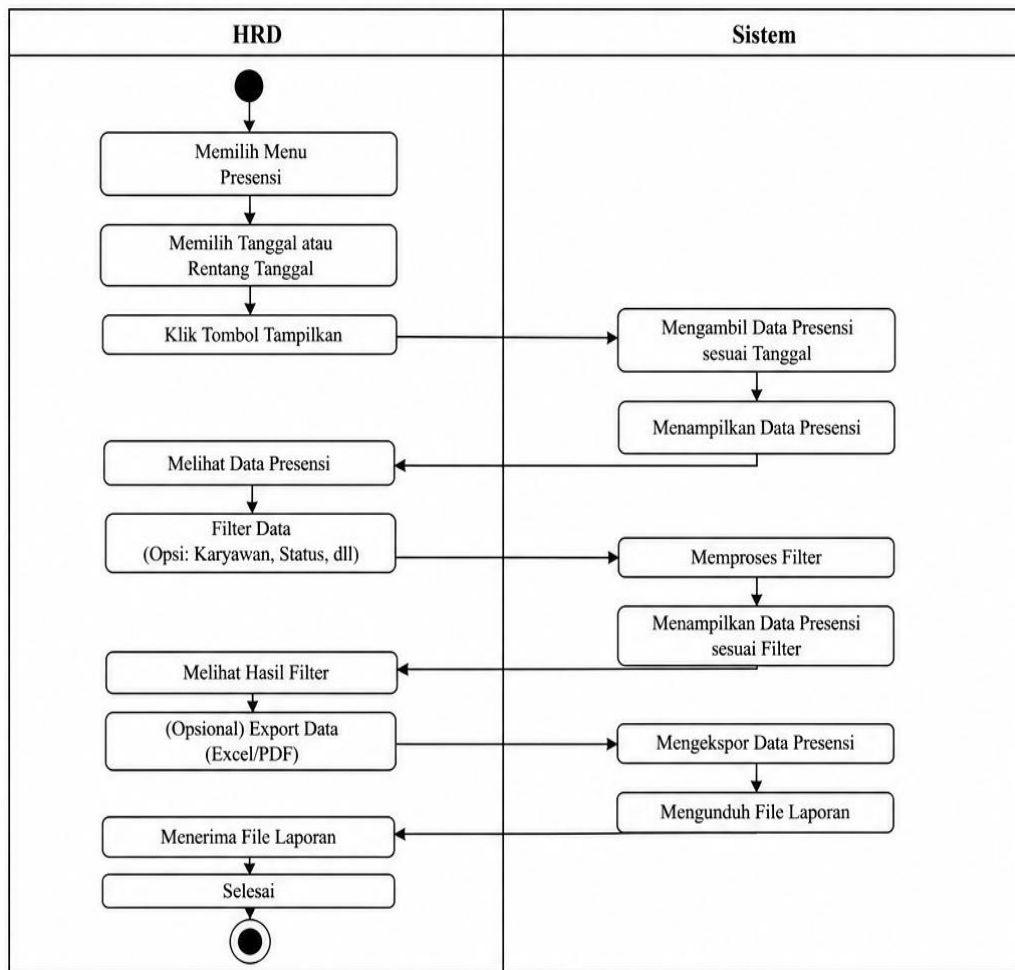


Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 16 *Activity Diagram Generate QR Code yang diusulkan*

Diagram ini menggambarkan proses *HRD* dalam membuat *QR Code* untuk karyawan. Proses dimulai ketika *HRD* memilih menu *generate QR Code*, kemudian memilih data karyawan yang akan dibuatkan *QR Code*. Setelah itu *HRD* menekan tombol *generate QR Code* dan sistem akan membuat *QR Code* berdasarkan ID atau NIK karyawan. *QR Code* yang dihasilkan kemudian disimpan ke dalam *database* dan ditampilkan kepada *HRD*. Selanjutnya *HRD* dapat melihat *QR Code* tersebut dan secara opsional dapat mengunduh atau mencetak *QR Code* untuk digunakan oleh karyawan dalam proses presensi.

13. Activity Diagram Kelola Data Presensi yang diusulkan

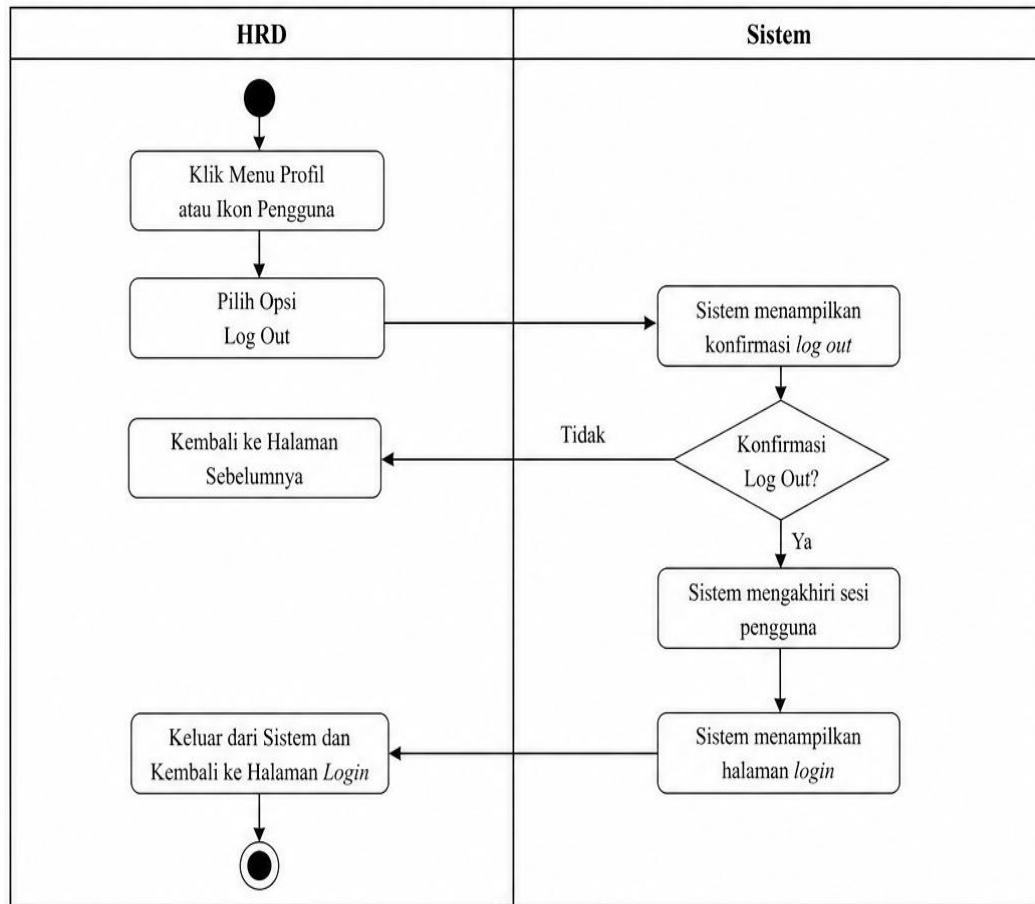


Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 17 Activity Diagram Kelola Presensi yang diusulkan

Diagram ini menggambarkan proses *HRD* dalam mengelola data presensi karyawan. Proses dimulai ketika *HRD* memilih menu presensi, kemudian menentukan tanggal atau rentang tanggal yang ingin dilihat dan menekan tombol tampilkan. Sistem akan mengambil data presensi sesuai dengan periode yang dipilih dan menampilkannya kepada *HRD*. Selanjutnya *HRD* dapat melakukan filter data berdasarkan karyawan, status, atau kriteria lainnya untuk mempermudah pencarian informasi. Sistem akan memproses filter tersebut dan menampilkan hasil sesuai dengan kriteria yang dipilih. Selain itu, *HRD* juga dapat mengekspor data presensi ke dalam *format* Excel atau PDF. Sistem kemudian memproses permintaan ekspor dan menyediakan file laporan yang dapat diunduh oleh *HRD* untuk kebutuhan pelaporan.

14. Activity Diagram Log Out HRD

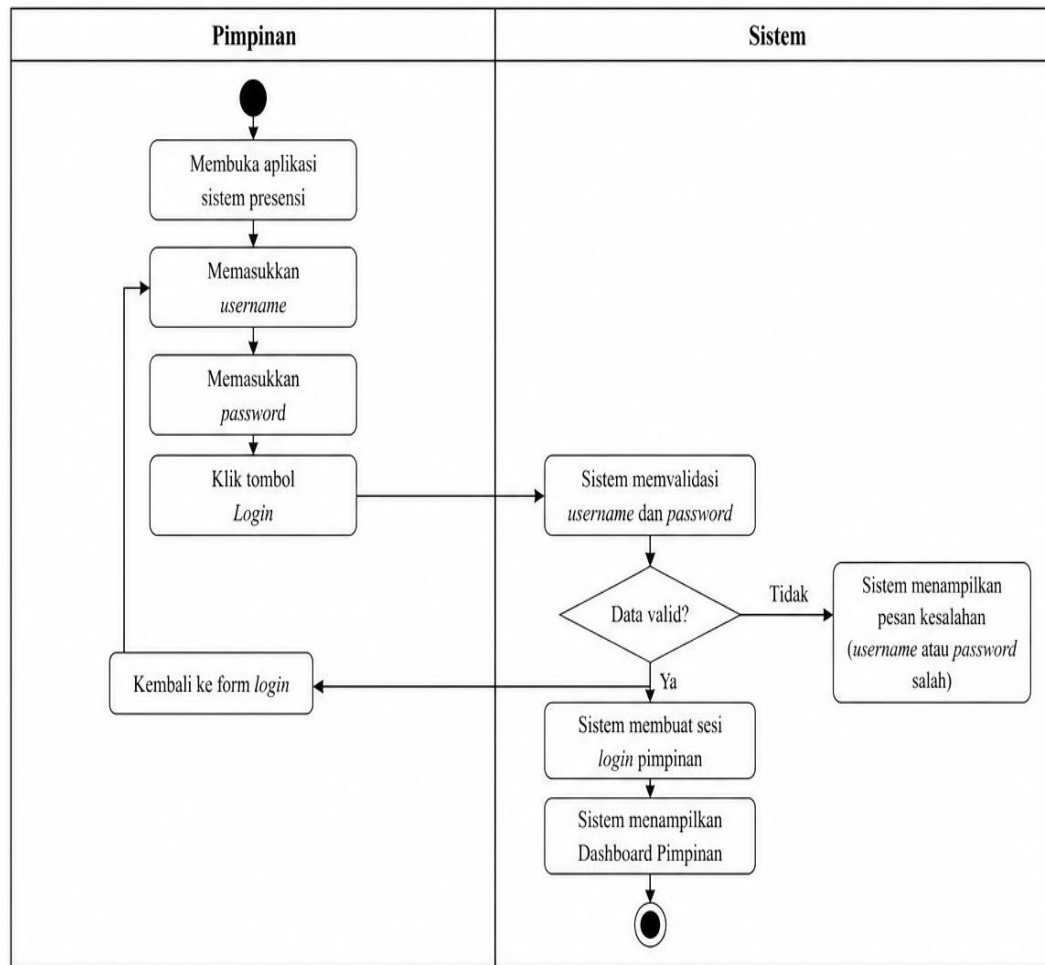


Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 18 Activity Diagram Log Out HRD yang diusulkan

Diagram ini menggambarkan proses *HRD* dalam melakukan *logout* dari sistem. Proses dimulai ketika *HRD* mengklik menu profil atau ikon pengguna, kemudian memilih opsi *log out*. Sistem akan menampilkan konfirmasi untuk memastikan bahwa pengguna ingin keluar dari sistem. Jika *HRD* memilih “Tidak”, maka sistem akan membatalkan proses dan kembali ke halaman sebelumnya. Namun jika *HRD* memilih “Ya”, sistem akan mengakhiri sesi pengguna dan menampilkan halaman *login*. Setelah itu *HRD* keluar dari sistem dan proses *logout* selesai.

15. Activity Diagram Login Pimpinan

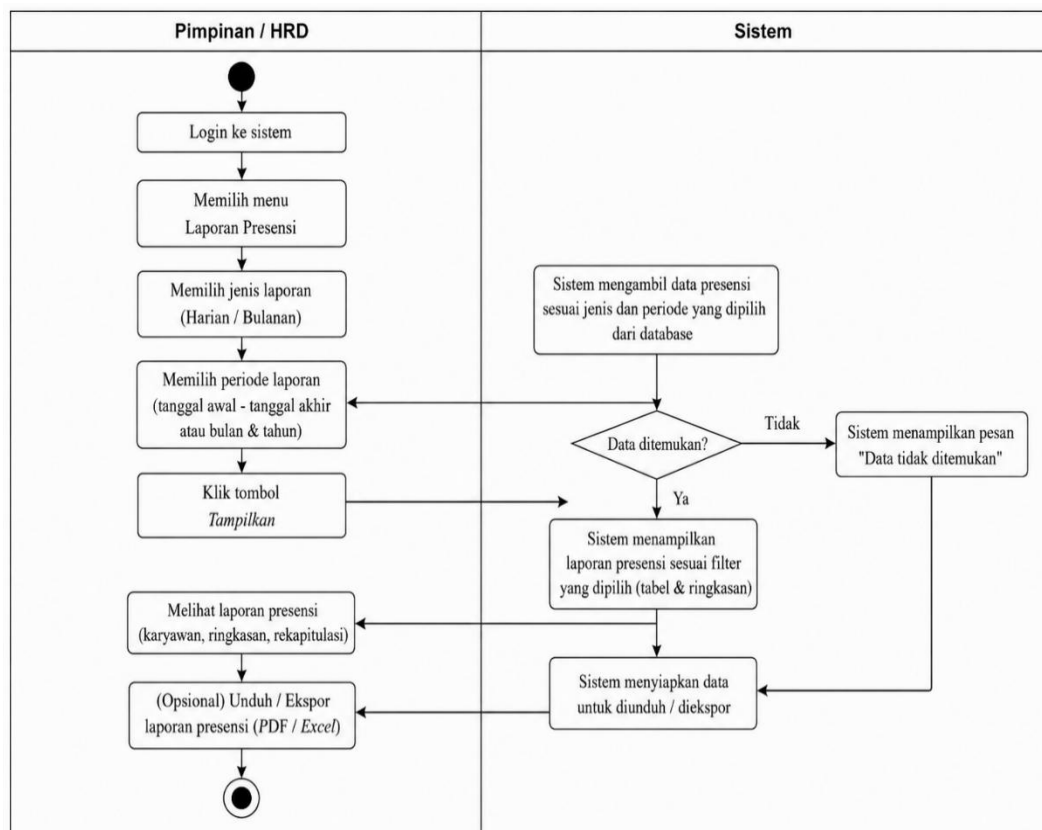


Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 19 Activity Login Pimpinan yang diusulkan

Diagram ini menggambarkan proses *login* yang dilakukan oleh pimpinan untuk mengakses sistem. Proses dimulai ketika pimpinan membuka aplikasi, kemudian memasukkan *username* dan *password*, lalu menekan tombol *login*. Sistem akan melakukan *validasi* terhadap data yang dimasukkan. Jika data tidak *valid*, sistem akan menampilkan pesan kesalahan dan pengguna diminta untuk kembali ke *form login*. Namun jika data *valid*, sistem akan membuat sesi *login* untuk pimpinan dan menampilkan *dashboard* pimpinan sehingga pengguna dapat mengakses fitur yang tersedia.

16. Activity Diagram Laporan Presensi yang diusulkan

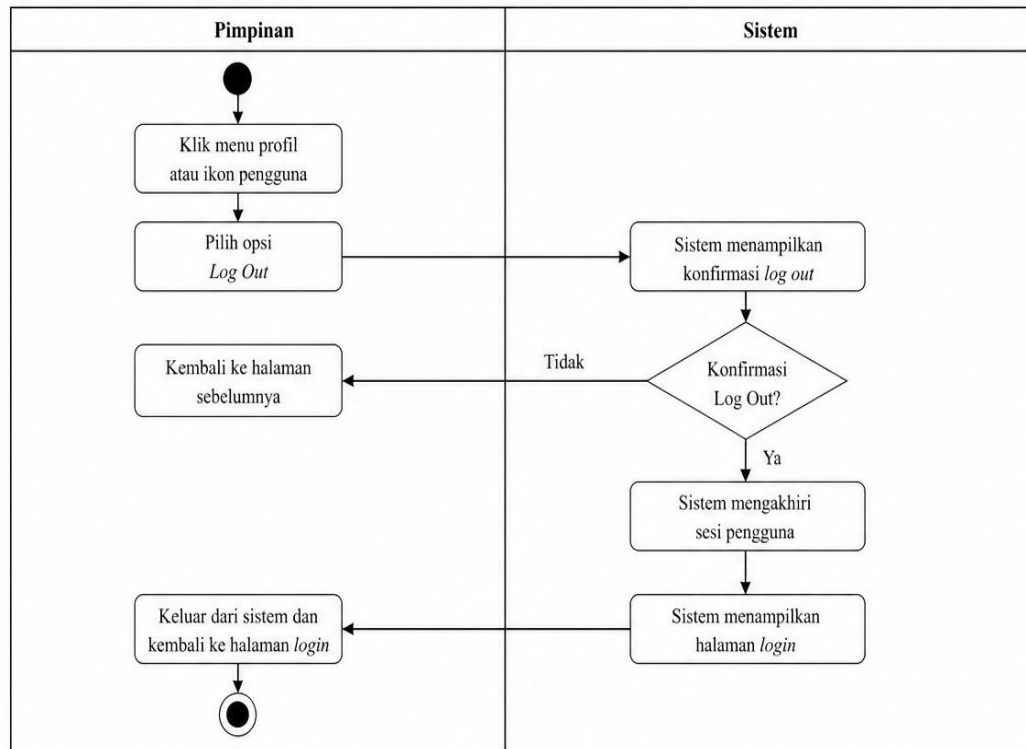


Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 20 Activity Diagram Laporan Presensi yang diusulkan

Diagram ini menggambarkan proses pimpinan dalam melihat rekap kehadiran karyawan secara keseluruhan. Proses dimulai ketika pimpinan *login* ke dalam sistem, kemudian memilih menu rekap kehadiran. Selanjutnya pimpinan menentukan periode yang ingin dilihat dengan memilih tanggal awal hingga tanggal akhir, lalu menekan tombol tampilkan. Sistem akan mengambil data kehadiran dari *database* sesuai dengan periode yang dipilih. Jika data tidak ditemukan, sistem akan menampilkan pesan bahwa data tidak tersedia dan pimpinan diminta untuk memilih periode lain. Namun jika data ditemukan, sistem akan menampilkan rekap kehadiran dalam bentuk tabel dan ringkasan informasi. Selain itu, pimpinan juga dapat mengunduh atau mengekspor rekap kehadiran dalam *format* Excel atau PDF untuk keperluan laporan.

17. Activity Diagram Log Out Pimpinan



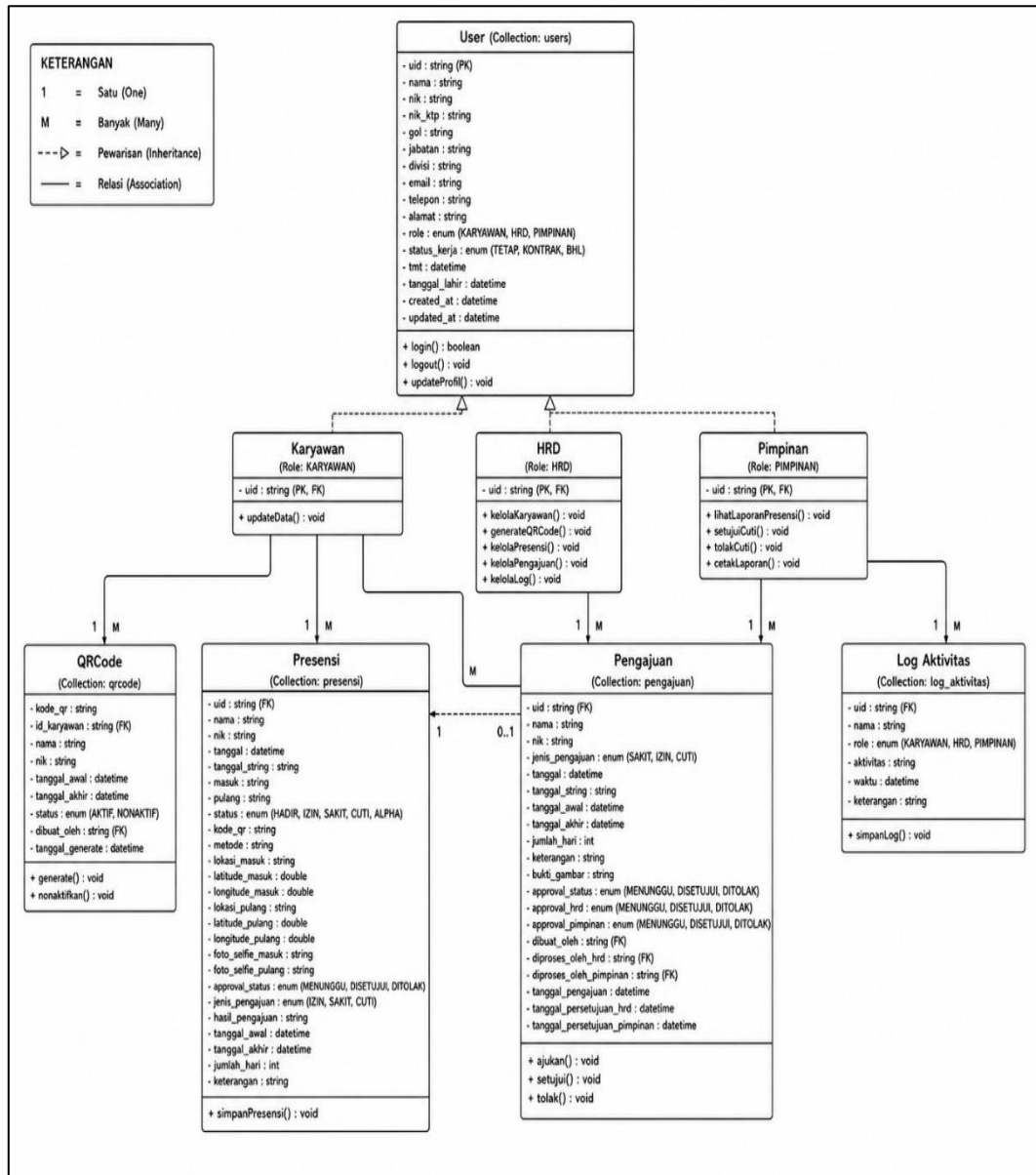
Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 21 Activity Diagram Log Out Pimpinan yang diusulkan

Diagram ini menggambarkan proses pimpinan dalam melakukan *logout* dari sistem. Proses dimulai ketika pimpinan mengklik menu profil atau ikon pengguna, kemudian memilih opsi *log out*. Sistem akan menampilkan konfirmasi untuk memastikan bahwa pengguna ingin keluar dari sistem. Jika pimpinan memilih “Tidak”, maka sistem akan membatalkan proses dan kembali ke halaman sebelumnya. Namun jika pimpinan memilih “Ya”, sistem akan mengakhiri sesi pengguna dan menampilkan halaman *login*. Setelah itu pimpinan keluar dari sistem dan proses *log out* selesai.

c. Class Diagram

Berdasarkan dari *use case diagram* dan *activity diagram* yang diusulkan, maka *class diagram* dari perancangan sistem informasi karyawan dan presensi menggunakan *QR Code* yang diusulkan sebagai berikut:



Sumber: Data diolah oleh Penulis, 2026

Gambar 4. 22 Class Diagram yang Diusulkan

4.2.2 Perancangan Database (Fisik)

Adapun beberapa perancangan *database* yang akan digunakan pada pengembangan aplikasi sistem informasi karyawan dan presensi menggunakan *QR Code* berbasis *android* adalah sebagai berikut:

1. Tabel *Users*

Nama: Tabel *Users*

Fungsi: Berfungsi untuk menyimpan data akun pengguna yang dapat mengakses sistem, baik *HRD*, pimpinan, maupun karyawan. *Users*

Tabel 4. 1 *Users*

<i>Field Name</i>	<i>Data Type</i>	<i>Keterangan</i>
nama	<i>string</i>	Nama pengguna/karyawan
nik	<i>string</i>	Nomor induk karyawan
nik_ktp	<i>string</i>	Nomor induk kependudukan
gol	<i>string</i>	Golongan karyawan
jabatan	<i>string</i>	Jabatan karyawan
divisi	<i>string</i>	Divisi/bagian karyawan
<i>email</i>	<i>string</i>	<i>Email login</i> pengguna
telepon	<i>string</i>	Nomor telepon
alamat	<i>string</i>	Alamat karyawan
<i>role</i>	<i>string</i>	Hak akses pengguna: <i>HRD</i> , Pimpinan, atau Karyawan
status	<i>string</i>	Status kerja karyawan: Tetap atau Kontrak
tmt	<i>timestamp</i>	Tanggal mulai terhitung bekerja
tanggal_lahir	<i>timestamp</i>	Tanggal lahir karyawan
<i>updated_at</i>	<i>timestamp</i>	Tanggal dan waktu data pengguna terakhir diperbarui
<i>created_at</i>	<i>timestamp</i>	Tanggal dan waktu data pengguna pertama kali dibuat

Sumber: data diolah oleh penulis, 2026

2. Tabel *QR Code*

Nama: Tabel *QR Code*

Fungsi: Berfungsi untuk menyimpan data *QR Code* unik yang digunakan karyawan dalam proses presensi.

Tabel 4. 2 *QR Code*

<i>Field Name</i>	<i>Data Type</i>	<i>Keterangan</i>
<i>kode_qr</i>	<i>string</i>	Kode unik <i>QR Code</i>
<i>id_karyawan</i>	<i>string</i>	ID karyawan atau UID pengguna
<i>nama</i>	<i>string</i>	Nama karyawan
<i>nik</i>	<i>string</i>	NIK karyawan
<i>tanggal_awal</i>	<i>timestamp</i>	Tanggal awal masa berlaku <i>QR Code</i>
<i>tanggal_akhir</i>	<i>timestamp</i>	Tanggal akhir masa berlaku <i>QR Code</i>
<i>status</i>	<i>string</i>	Status <i>QR Code</i> : aktif atau nonaktif
<i>dibuat_oleh</i>	<i>string</i>	UID pengguna yang membuat <i>QR Code</i>
<i>tanggal_generate</i>	<i>timestamp</i>	Tanggal pembuatan <i>QR Code</i>

Sumber: data diolah oleh penulis, 2026

3. Tabel Presensi

Nama: Tabel Presensi

Fungsi: Berfungsi untuk menyimpan data kehadiran karyawan berdasarkan hasil *scan QR Code*, *selfie*, dan pencatatan lokasi *GPS*.

Tabel 4. 3 Presensi

<i>Field Name</i>	<i>Data Type</i>	<i>Keterangan</i>
<i>uid</i>	<i>string</i>	UID karyawan
<i>nama</i>	<i>string</i>	Nama karyawan
<i>nik</i>	<i>string</i>	NIK karyawan
<i>tanggal</i>	<i>timestamp</i>	Tanggal dan waktu presensi dibuat
<i>tanggal_string</i>	<i>string</i>	Tanggal presensi dalam <i>format yyyy-MM-dd</i>
<i>masuk</i>	<i>string</i>	Jam presensi masuk
<i>pulang</i>	<i>string</i>	Jam presensi pulang
<i>status</i>	<i>string</i>	Status presensi: hadir, sakit, izin, cuti, atau <i>alpha</i>
<i>kode_qr</i>	<i>string</i>	Kode <i>QR</i> yang digunakan saat presensi

metode	string	Metode presensi, misalnya <i>QR Code</i> + <i>Selfie</i> + <i>GPS</i>
lokasi_masuk	string	Alamat lokasi saat presensi masuk
latitude_masuk	double	Titik <i>latitude</i> saat presensi masuk
longitude_masuk	double	Titik <i>longitude</i> saat presensi masuk
lokasi_pulang	string	Alamat lokasi saat presensi pulang
latitude_pulang	double	Titik <i>latitude</i> saat presensi pulang
longitude_pulang	double	Titik <i>longitude</i> saat presensi pulang
foto_selfie_masuk	string	<i>URL</i> foto <i>selfie</i> saat presensi masuk
foto_selfie_pulang	string	<i>URL</i> foto <i>selfie</i> saat presensi pulang
approval_status	string	Status persetujuan pengajuan: menunggu, disetujui, atau ditolak
jenis_pengajuan	string	Jenis pengajuan: izin, sakit, atau cuti
hasil_pengajuan	string	Hasil pengajuan: disetujui atau ditolak
keterangan	string	Keterangan presensi atau alasan pengajuan

Sumber: data diolah oleh Penulis, 2026

4. Tabel Pengajuan

Nama: Tabel Pengajuan

Fungsi: Berfungsi untuk menyimpan data pengajuan sakit, izin, dan cuti yang diajukan oleh karyawan sebelum disetujui atau ditolak oleh *HRD*.

Tabel 4. 4 Pengajuan

Field Name	Data Type	Keterangan
uid	string	UID karyawan
nama	string	Nama karyawan
nik	string	Nomor induk karyawan
jenis_pengajuan	string	Jenis pengajuan: sakit, izin, atau cuti
tanggal	timestamp	Tanggal pengajuan tidak masuk kerja
tanggal_string	string	Tanggal dalam format yyyy-MM-dd
tanggal_awal	timestamp	Tanggal mulai cuti
tanggal_akhir	timestamp	Tanggal akhir cuti

jumlah_hari	integer	Jumlah hari cuti
keterangan	string	Alasan atau keterangan pengajuan
bukti_gambar	string	URL foto bukti pendukung pengajuan
approval_status	string	Status pengajuan: menunggu, disetujui, atau ditolak
approval_hrd	string	Status persetujuan oleh HRD
approval_pimpinan	string	Status persetujuan oleh pimpinan
dibuat_oleh	string	UID karyawan yang membuat pengajuan
diproses_oleh_hrd	string	UID HRD yang memproses pengajuan
diproses_oleh_pimpinan	string	UID pimpinan yang memproses pengajuan cuti
tanggal_pengajuan	timestamp	Waktu pengajuan dibuat
tanggal_persetujuan_hrd	timestamp	Waktu pengajuan disetujui atau ditolak oleh HRD
tanggal_persetujuan_pimpinan	timestamp	Waktu pengajuan disetujui atau ditolak oleh pimpinan

Sumber: data diolah oleh Penulis, 2026

5. Tabel Log Aktivitas

Nama: Tabel Log Aktivitas

Fungsi: Berfungsi untuk mencatat aktivitas pengguna di dalam sistem, seperti login, logout, tambah data, edit data, hapus data, generate QR Code, serta persetujuan pengajuan presensi

Tabel 4. 5 Log Aktivitas

Field Name	Data Type	Keterangan
uid	string	UID pengguna
nama	string	Nama pengguna
role	string	Karyawan/HRD/Pimpinan
aktivitas	string	Aktivitas yang dilakukan pengguna
waktu	timestamp	Waktu aktivitas dilakukan
keterangan	string	Keterangan tambahan aktivitas

Sumber: Data diolah oleh Penulis, 2026

4.2.3 Perancangan Antarmuka (UI/UX)

Berikut merupakan perancangan antarmuka (UI/UX) untuk pengembangan aplikasi Sistem Informasi Karyawan dan Presensi *QR Code*.

1. Halaman *Splash Screen*

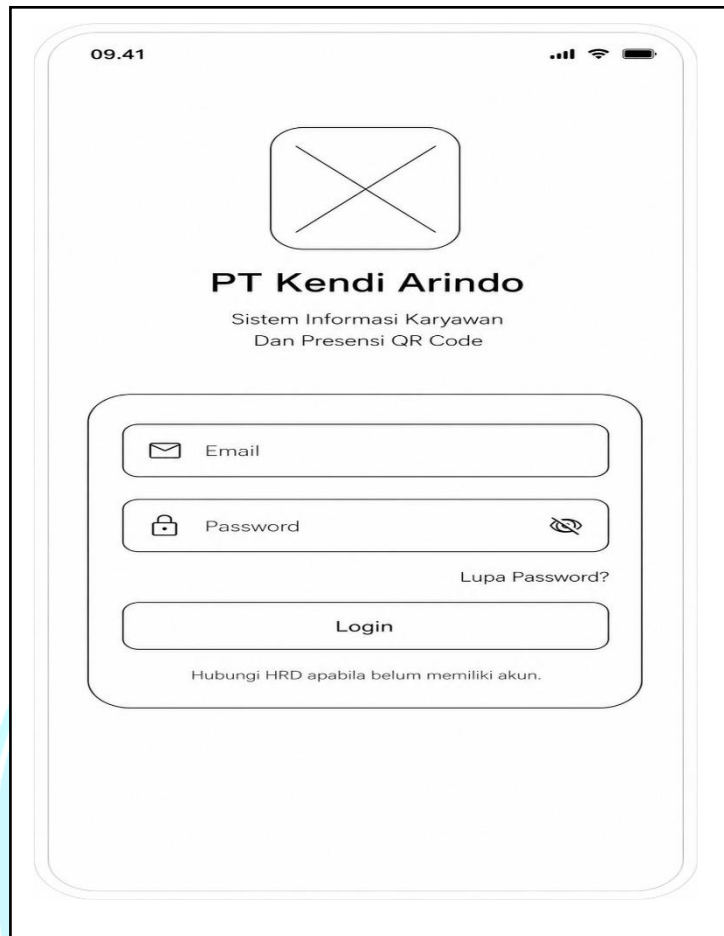


Sumber: data diolah oleh penulis, 2026

Gambar 4. 23 Halaman *Splash Screen*

Pada gambar tampilan Halaman *splash screen* merupakan tampilan awal yang muncul saat aplikasi “Digitalisasi Sistem Informasi Karyawan dan Presensi Menggunakan *QR Code* Berbasis *Android*” dijalankan. Tampilan ini berfungsi sebagai halaman pembuka sebelum pengguna masuk ke halaman *login* atau halaman utama aplikasi. Pada halaman *splash screen* ditampilkan logo perusahaan PT Kendi Arindo sebagai identitas aplikasi, disertai dengan nama sistem yaitu “Sistem Informasi Karyawan dan Presensi *QR Code*”.

2. Halaman *Login*



09.41

PT Kendi Arindo
Sistem Informasi Karyawan
Dan Presensi QR Code

Email

Password [Lupa Password?](#)

Login

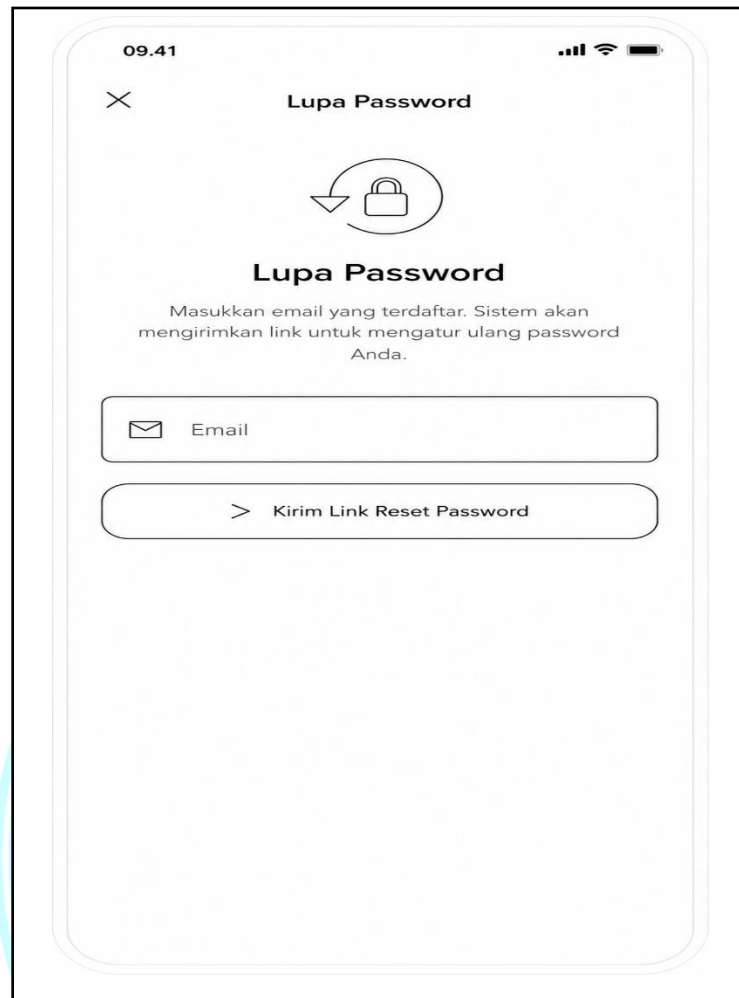
Hubungi HRD apabila belum memiliki akun.

Sumber: data diolah oleh penulis, 2026

Gambar 4. 24 Halaman *Login* Karyawan

Pada gambar tampilan *login* di atas, dijelaskan bahwa untuk dapat mengakses aplikasi, pengguna harus memasukkan *email* dan *password* terlebih dahulu. Setelah pengguna memasukkan *email* dan *password* dengan benar, sistem akan melakukan proses validasi. Jika data yang dimasukkan sesuai, maka pengguna akan berhasil masuk ke dalam aplikasi dan diarahkan ke halaman utama. Namun, jika data yang dimasukkan salah, sistem akan menampilkan pesan kesalahan dan pengguna diminta untuk menginput ulang data *login* dengan benar. Selain itu, tersedia fitur tambahan seperti lupa *password* dan *login* menggunakan Google untuk memudahkan pengguna dalam mengakses aplikasi.

3. Halaman Lupa Password



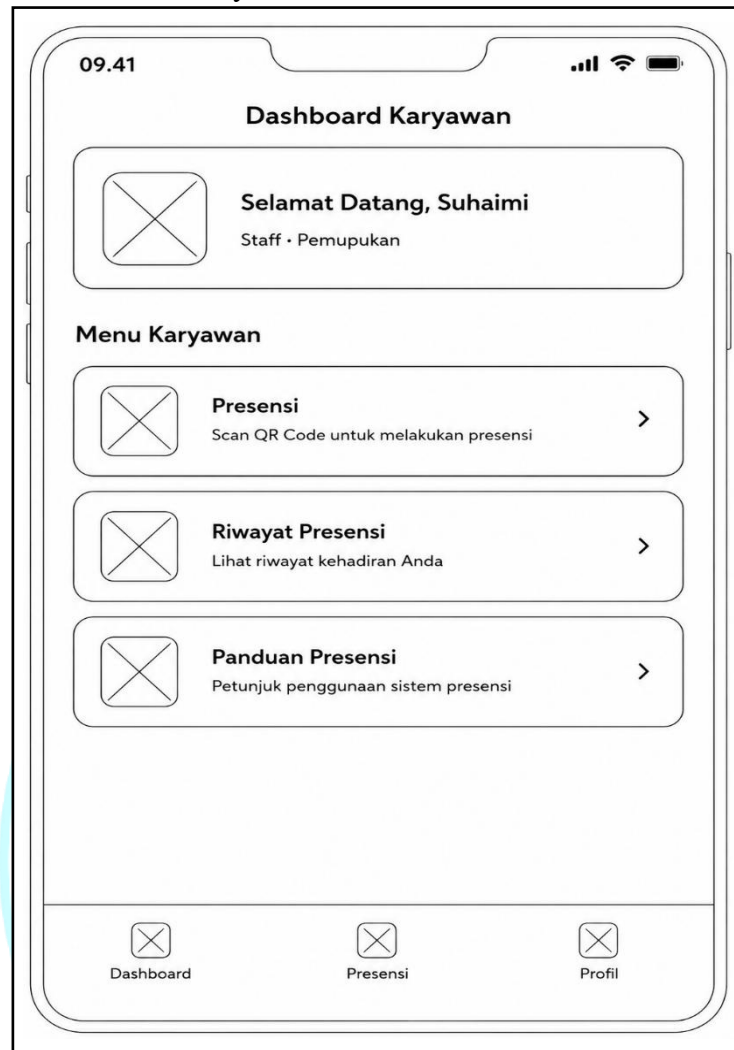
Sumber: data diolah oleh penulis, 2026

Gambar 4. 25 Halaman Lupa Password

Halaman lupa *password* merupakan halaman yang digunakan pengguna untuk melakukan proses *reset password* apabila lupa kata sandi akun. Pada halaman ini, pengguna dapat memasukkan *email* yang telah terdaftar pada sistem untuk menerima *link reset password* melalui *email*.

Pada bagian tengah halaman ditampilkan ikon beserta judul lupa *Password* sebagai informasi fungsi halaman. Di bawahnya terdapat penjelasan singkat mengenai proses pengiriman *link reset password* oleh sistem. Selanjutnya terdapat *field Email* yang digunakan pengguna untuk memasukkan alamat *email* akun yang terdaftar. Setelah *email* diisi, pengguna dapat menekan tombol kirim *link reset password* untuk mengirim permintaan *reset password*.

4. Halaman *Dashboard* Karyawan

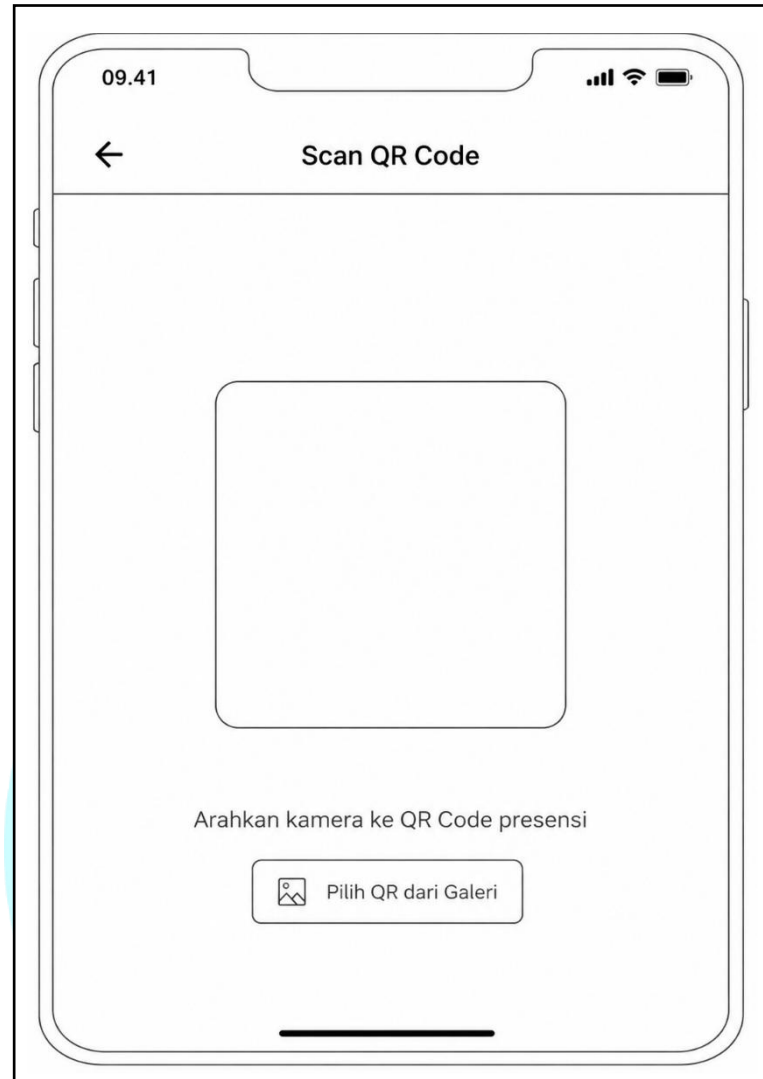


Sumber: data diolah oleh penulis, 2026

Gambar 4. 26 Halaman *Dashboard* Karyawan

Pada gambar tampilan beranda di atas, dijelaskan bahwa setelah berhasil melakukan *login*, pengguna akan diarahkan ke halaman *dashboard* karyawan. Pada halaman terdapat berbagai menu utama yang dapat digunakan oleh karyawan, seperti presensi (*scan QR Code*), riwayat presensi, panduan presensi. Di bagian bawah, terdapat navigasi utama untuk berpindah antar halaman, yaitu *dashboard*, presensi dan profil. Halaman *dashboard* ini berfungsi sebagai pusat informasi dan akses cepat ke fitur-fitur penting dalam aplikasi.

5. Halaman Presensi *Scan QR Code* Karyawan



Sumber: data diolah oleh penulis, 2026

Gambar 4. 27 Halaman Presensi *Scan QR Code* Karyawan

Pada gambar tampilan presensi di atas, dijelaskan bahwa pengguna dapat melakukan presensi dengan cara melakukan *scan QR Code* yang telah disediakan. Proses dimulai ketika pengguna membuka menu presensi, kemudian sistem akan menampilkan halaman *scan QR Code*. Pengguna diarahkan untuk mengarahkan kamera ke *QR Code* hingga terbaca dengan jelas oleh sistem. Setelah *QR Code* berhasil dipindai, sistem akan memproses data presensi dan melanjutkan ke tahap berikutnya, yaitu verifikasi tambahan seperti pengambilan foto *selfie* dan pencatatan lokasi *GPS*.

6. Halaman Verifikasi Presensi (Masuk dan Pulang)

The screenshot displays the 'Verifikasi Presensi' interface. At the top, the status bar shows the time 09.41 and signal/battery icons. Below the title bar, a back arrow is on the left, and the title 'Verifikasi Presensi' is centered. The main content area includes:

- A box labeled 'QR Code terbaca' with a placeholder icon and the text 'QR-21818181-1780250289581'.
- A section titled 'Jenis Presensi' containing five radio button options: 'Presensi Masuk' (selected), 'Presensi Pulang', 'Sakit', 'Izin', and 'Cuti'.
- A 'Foto Selfie' section with a circular placeholder icon and a button labeled 'Ambil Selfie'.
- A 'Pencatatan Lokasi (GPS)' section showing 'Lokasi: -' and a button labeled 'Catat Lokasi'.
- A large button at the bottom labeled 'Konfirmasi Presensi MASUK'.

Sumber: data diolah oleh penulis, 2026

Gambar 4. 28 Halaman Verifikasi Presensi

Pada gambar tampilan verifikasi presensi di atas, dijelaskan bahwa setelah pengguna berhasil melakukan *scan QR Code*, sistem akan mengarahkan pengguna ke halaman verifikasi presensi. Pada halaman ini, pengguna diwajibkan untuk mengambil foto *selfie* sebagai bukti kehadiran dengan memastikan wajah terlihat jelas di dalam area yang telah disediakan. Setelah itu, sistem secara otomatis mendeteksi lokasi pengguna melalui *GPS* dan menampilkan informasi lokasi seperti alamat dan tingkat akurasi. Selanjutnya, sistem juga menampilkan ringkasan data presensi yang meliputi tanggal, waktu, dan metode presensi yang digunakan. Setelah semua proses verifikasi selesai, pengguna dapat menekan tombol konfirmasi presensi untuk menyimpan data ke dalam sistem. Halaman ini berfungsi

untuk memastikan bahwa proses presensi dilakukan secara valid dan akurat melalui verifikasi identitas dan lokasi pengguna.

7. Halaman Verifikasi Presensi (Sakit dan Izin)

The screenshot displays the 'Verifikasi Presensi' screen on a mobile device. At the top, the status 'Sakit' is selected with a radio button. Below this, the 'Form Pengajuan SAKIT' section includes a date picker showing '04-06-2026', a text input field for 'Keterangan', and a note: 'Catatan: pengajuan akan berstatus menunggu persetujuan HRD.' The 'Upload Bukti' section features a photo icon and an 'Upload Bukti Gambar' button. The bottom of the screen has a 'Kirim Pengajuan SAKIT' button.

Sumber: data diolah oleh penulis, 2026

Gambar 4. 29 Halaman Verifikasi Presensi

Pada gambar tampilan verifikasi presensi di atas, dijelaskan bahwa pengguna dapat memilih jenis presensi yang akan diajukan, yaitu sakit, izin, atau cuti. Pada contoh tampilan tersebut, pengguna memilih opsi sakit sehingga sistem menampilkan *form* pengajuan sakit. Selanjutnya, pengguna mengisi keterangan mengenai alasan ketidakhadiran pada kolom yang telah disediakan. Sistem juga menampilkan tanggal pengajuan secara otomatis sebagai informasi waktu pengajuan. Setelah itu, pengguna diwajibkan mengunggah bukti pendukung, seperti

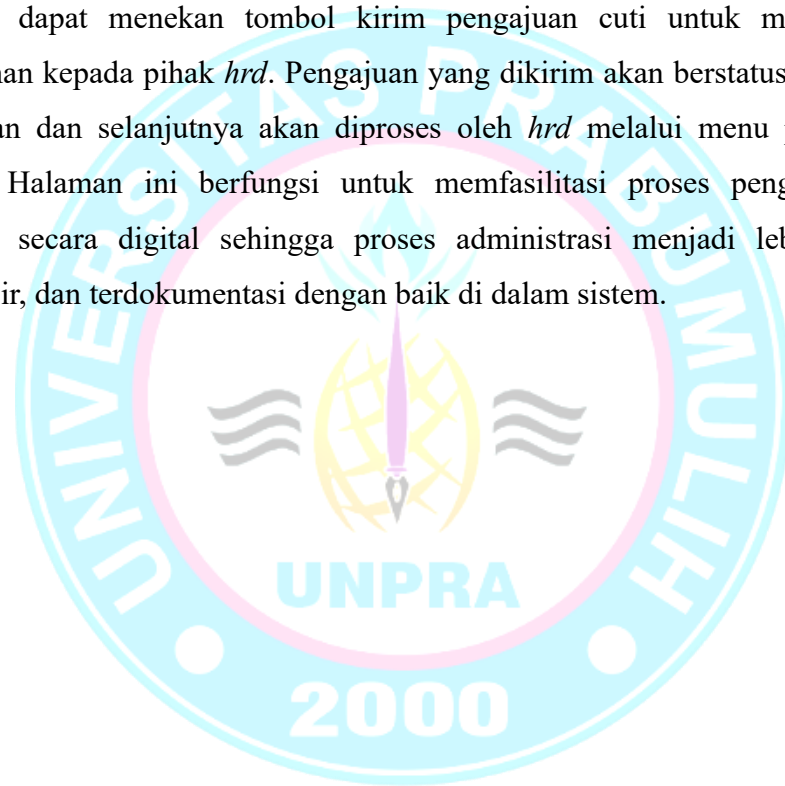
surat keterangan dokter atau dokumen lain yang relevan, melalui tombol *upload* bukti gambar. Setelah seluruh data terisi dengan lengkap dan bukti telah diunggah, pengguna dapat menekan tombol kirim pengajuan sakit untuk mengirimkan permohonan kepada pihak *hrd*. Pengajuan yang dikirim akan berstatus menunggu persetujuan dan selanjutnya dapat ditinjau oleh *hrd* melalui menu persetujuan presensi. Halaman ini berfungsi untuk memfasilitasi proses pengajuan ketidakhadiran karyawan secara digital sehingga lebih mudah, terstruktur, dan terdokumentasi dengan baik dalam sistem.

8. Halaman Verifikasi Presensi (Cuti)

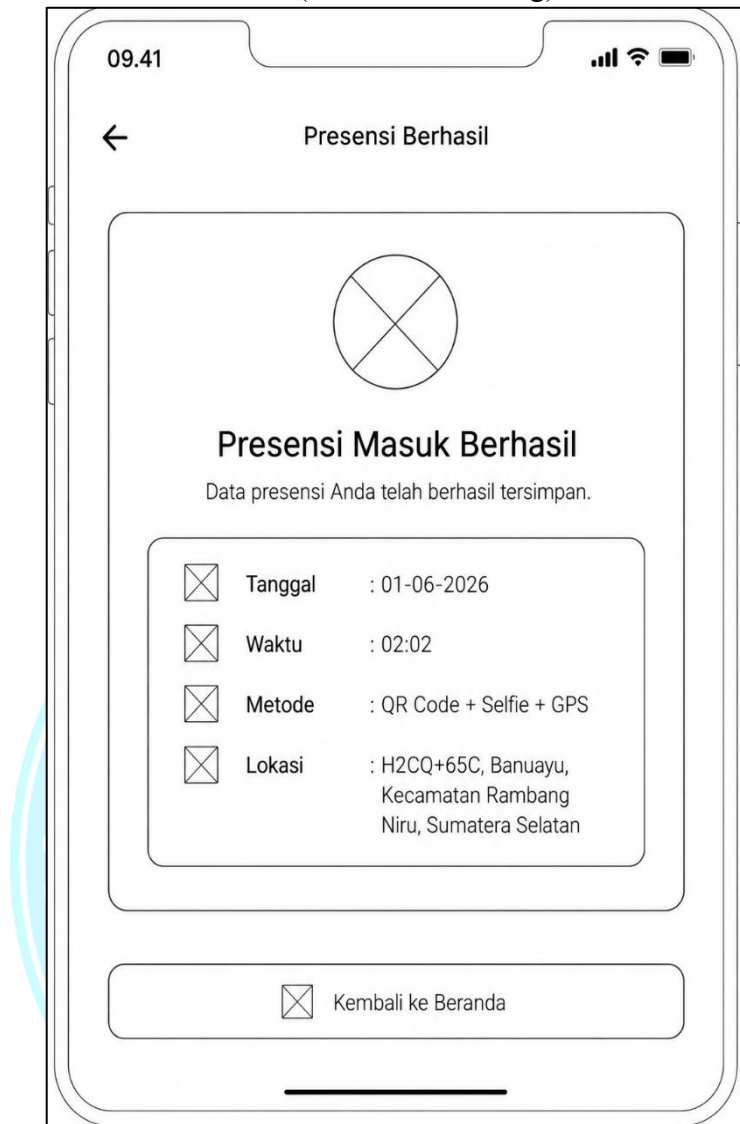
Sumber: data diolah oleh penulis, 2026

Gambar 4. 30 Halaman Verifikasi Presensi (Cuti)

Pada gambar tampilan verifikasi presensi di atas, dijelaskan bahwa pengguna dapat memilih jenis presensi yang akan diajukan, yaitu sakit, izin, atau cuti. Pada contoh tampilan tersebut, pengguna memilih opsi cuti sehingga sistem menampilkan *form* pengajuan cuti. Selanjutnya, pengguna diminta untuk menentukan tanggal awal cuti dan tanggal akhir cuti melalui fitur pemilihan tanggal yang telah disediakan. Setelah menentukan rentang waktu cuti, pengguna mengisi keterangan yang berisi alasan pengajuan cuti pada kolom yang tersedia. Pengguna juga diwajibkan mengunggah dokumen atau bukti pendukung apabila diperlukan melalui tombol *upload* bukti gambar. Setelah seluruh data terisi dengan lengkap, pengguna dapat menekan tombol kirim pengajuan cuti untuk mengirimkan permohonan kepada pihak *hrd*. Pengajuan yang dikirim akan berstatus menunggu persetujuan dan selanjutnya akan diproses oleh *hrd* melalui menu persetujuan presensi. Halaman ini berfungsi untuk memfasilitasi proses pengajuan cuti karyawan secara digital sehingga proses administrasi menjadi lebih efektif, terorganisir, dan terdokumentasi dengan baik di dalam sistem.



9. Halaman Presensi Berhasil (Masuk dan Pulang)



Sumber: data diolah oleh penulis, 2026

Gambar 4. 31 Halaman Presensi Berhasil (Masuk dan Pulang)

Pada gambar tampilan presensi berhasil di atas, dijelaskan bahwa setelah pengguna menyelesaikan seluruh proses presensi yang meliputi *scan QR Code*, pengambilan foto *selfie*, dan pencatatan lokasi *GPS*, sistem akan menampilkan halaman konfirmasi bahwa presensi telah berhasil dilakukan. Pada halaman ini ditampilkan informasi ringkasan presensi seperti tanggal, waktu, dan metode presensi yang digunakan. Dan pencatatan data lokasi saat melakukan presensi. Selain itu, terdapat informasi bahwa data presensi telah berhasil disimpan ke dalam sistem dan dapat dilihat kembali pada riwayat presensi. Pengguna juga disediakan tombol “Kembali ke Beranda” untuk kembali ke halaman utama aplikasi. Halaman

ini berfungsi sebagai indikator bahwa proses presensi telah selesai dan tersimpan dengan baik di dalam sistem.

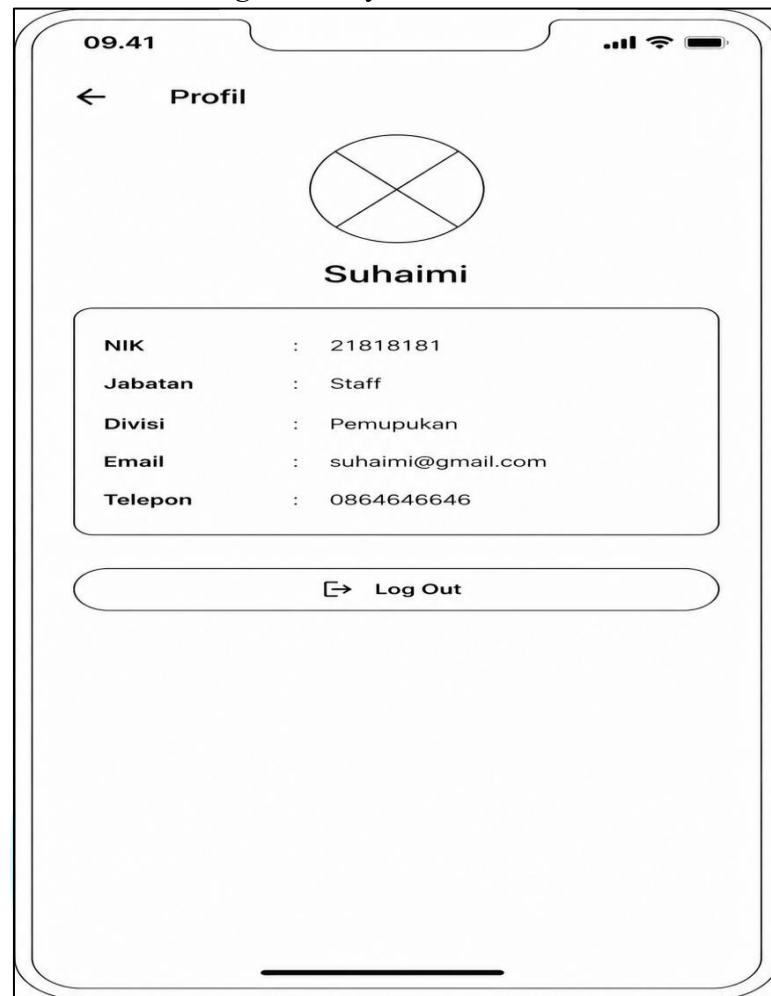
10. Halaman Riwayat Presensi



Sumber: data diolah oleh penulis, 2026

Gambar 4. 32 Halaman Riwayat Presensi

Pada gambar tampilan riwayat presensi di atas, dijelaskan bahwa halaman ini digunakan untuk melihat data kehadiran karyawan yang telah dilakukan sebelumnya. Pada tampilan ini ditampilkan informasi secara sederhana berupa tanggal presensi serta status kehadiran pada hari tersebut. Pengguna dapat melihat daftar riwayat presensi berdasarkan tanggal tanpa detail yang terlalu kompleks. Halaman ini berfungsi untuk memberikan gambaran singkat mengenai kehadiran karyawan sehingga mudah dipahami dan digunakan.

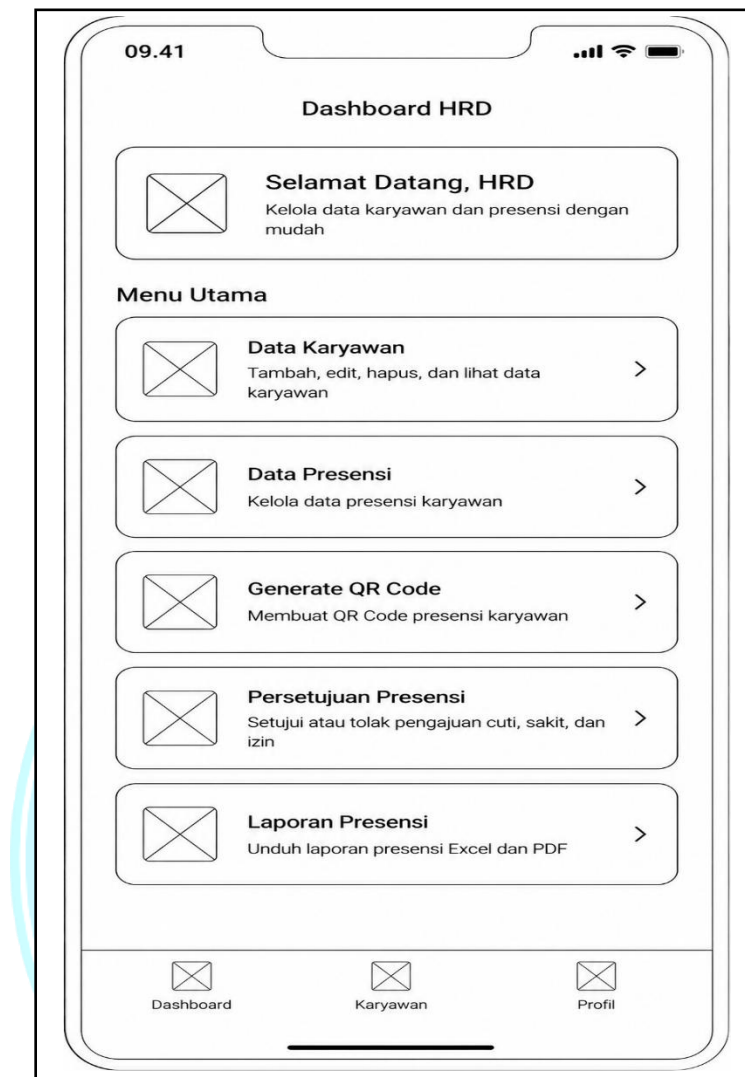
11. Halaman Profil dan *Log Out* Karyawan

Sumber: data diolah oleh penulis, 2026

Gambar 4. 33 Halaman Profil dan *Logout* Karyawan

Pada gambar tampilan profil di atas, dijelaskan bahwa halaman ini digunakan untuk menampilkan informasi data diri karyawan seperti nama, NIK, jabatan, divisi, *email* dan telepon. Selain itu, pada halaman ini juga tersedia tombol *logout* yang digunakan untuk keluar dari aplikasi. Ketika pengguna menekan tombol *logout*, maka sistem akan mengakhiri sesi pengguna dan mengarahkan kembali ke halaman *login*. Halaman ini berfungsi sebagai pusat informasi akun sekaligus tempat untuk mengelola akses pengguna dalam aplikasi.

12. Halaman *Dashboard HRD*



Sumber: data diolah oleh penulis, 2026

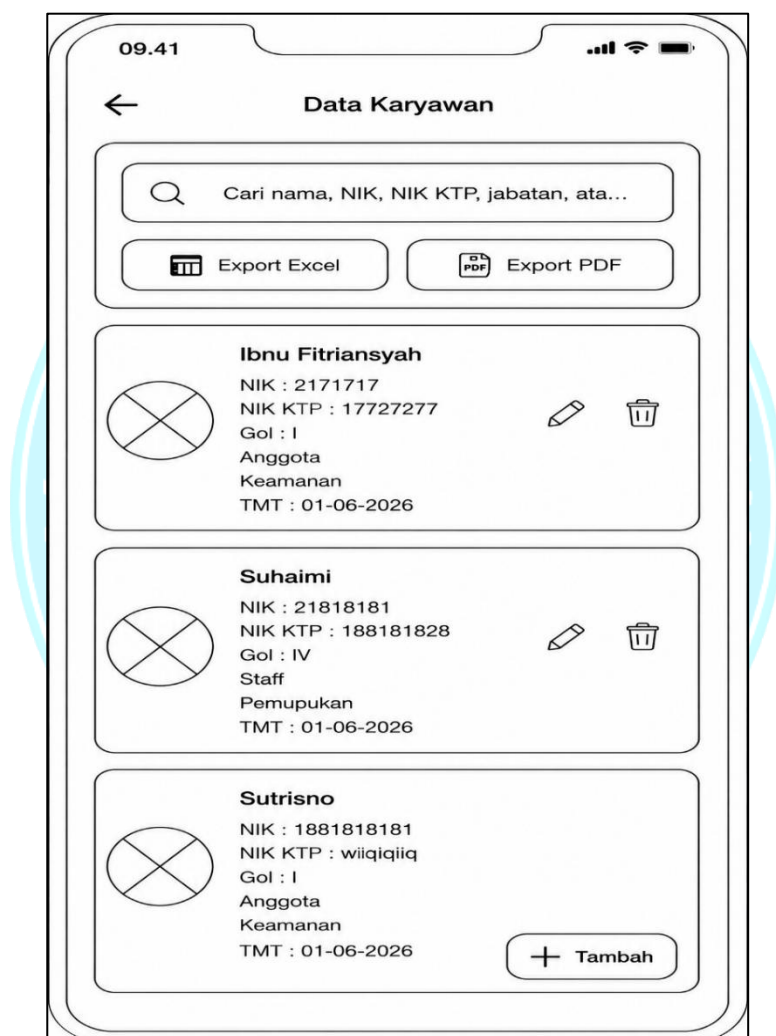
Gambar 4. 34 Halaman *Dashboard HRD*

Pada gambar tampilan dashboard *HRD* di atas, dijelaskan bahwa setelah *HRD* berhasil melakukan login, sistem akan menampilkan halaman dashboard *HRD* sebagai halaman utama aplikasi. Pada bagian atas halaman ditampilkan informasi sambutan yang menunjukkan bahwa pengguna sedang masuk sebagai *HRD* serta keterangan singkat mengenai fungsi sistem dalam mengelola data karyawan dan presensi. Selanjutnya, terdapat beberapa menu utama yang dapat digunakan oleh *HRD* untuk mengakses berbagai fitur yang tersedia dalam sistem.

Menu Data Karyawan digunakan untuk menambah, mengubah, menghapus, dan melihat data karyawan yang terdaftar. Menu Data Presensi digunakan untuk melihat dan mengelola data kehadiran karyawan. Menu *Generate QR Code*

digunakan untuk membuat *QR Code* yang akan digunakan dalam proses presensi karyawan. Menu Persetujuan Presensi digunakan untuk meninjau serta memberikan persetujuan atau penolakan terhadap pengajuan sakit, izin, dan cuti yang diajukan oleh karyawan. Sementara itu, menu Laporan Presensi digunakan untuk melihat serta mengunduh laporan presensi karyawan dalam format *Excel* maupun *PDF*.

13. Halaman Data Karyawan



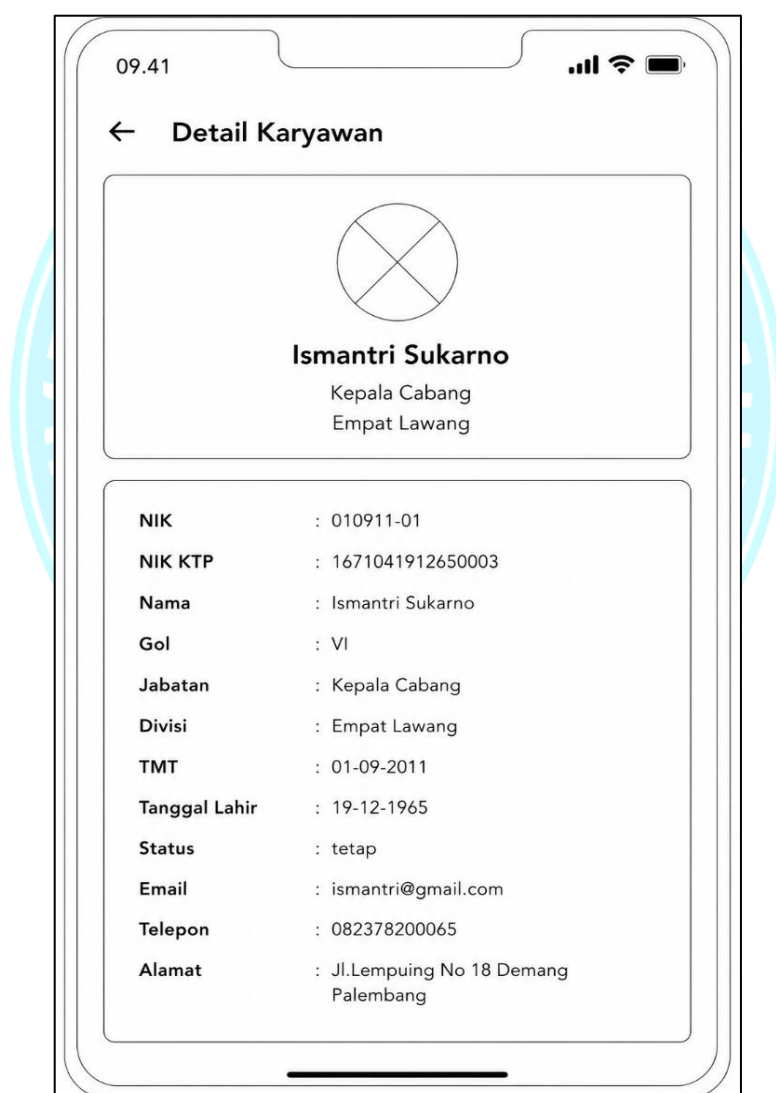
Sumber: data diolah oleh penulis, 2026

Gambar 4. 35 Halaman Data Karyawan

Pada gambar tampilan data karyawan di atas, dijelaskan bahwa halaman ini digunakan oleh *HRD* dan pimpinan untuk mengelola data karyawan yang terdapat dalam sistem. Pada halaman ini ditampilkan daftar karyawan secara lengkap yang meliputi nama karyawan, NIK, NIK KTP, gol, divisi, jabatan dan TMT. Setiap data

karyawan dilengkapi dengan ikon aksi seperti *edit* dan *hapus* yang memungkinkan *HRD* dan pimpinan untuk memperbarui atau menghapus data karyawan sesuai kebutuhan. Selain itu, terdapat fitur pencarian yang memudahkan *HRD* dan pimpinan dalam menemukan data karyawan berdasarkan nama, NIK, atau jabatan, serta tombol “Tambah” yang digunakan untuk menambahkan data karyawan baru ke dalam sistem. Halaman ini berfungsi sebagai pusat pengelolaan data karyawan agar dapat dikelola secara efektif dan efisien.

14. Halaman Detail Karyawan

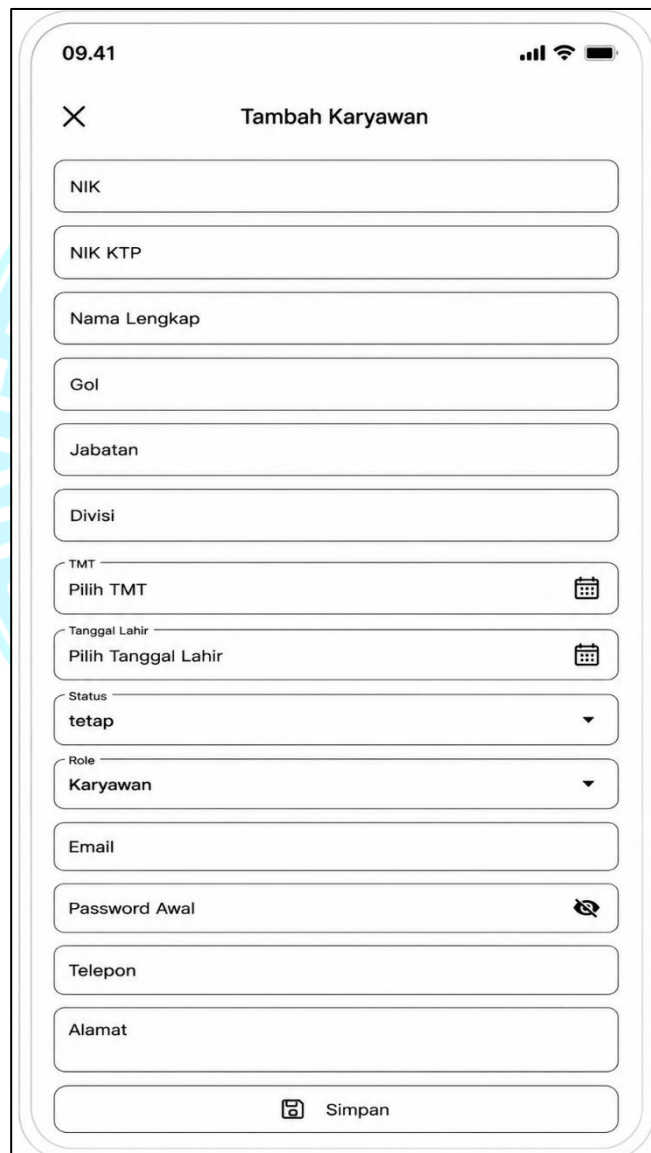


Sumber: data diolah oleh penulis, 2026

Gambar 4. 36 Halaman Detail Karyawan

Halaman detail karyawan merupakan halaman yang digunakan untuk menampilkan informasi lengkap data karyawan yang tersimpan di dalam sistem. Halaman ini membantu *HRD* maupun pimpinan dalam melihat detail identitas dan data kepegawaian setiap karyawan secara lebih lengkap. Pada bagian atas halaman terdapat *card* informasi karyawan yang menampilkan foto profil *default*, nama karyawan, jabatan, dan divisi kerja.

15. Halaman Tambah Karyawan



The screenshot shows a mobile application interface for adding a new employee. The title bar at the top says "Tambah Karyawan" with a close button (X) on the left. The status bar at the very top shows the time "09.41" and signal/battery icons. The form consists of the following fields from top to bottom:

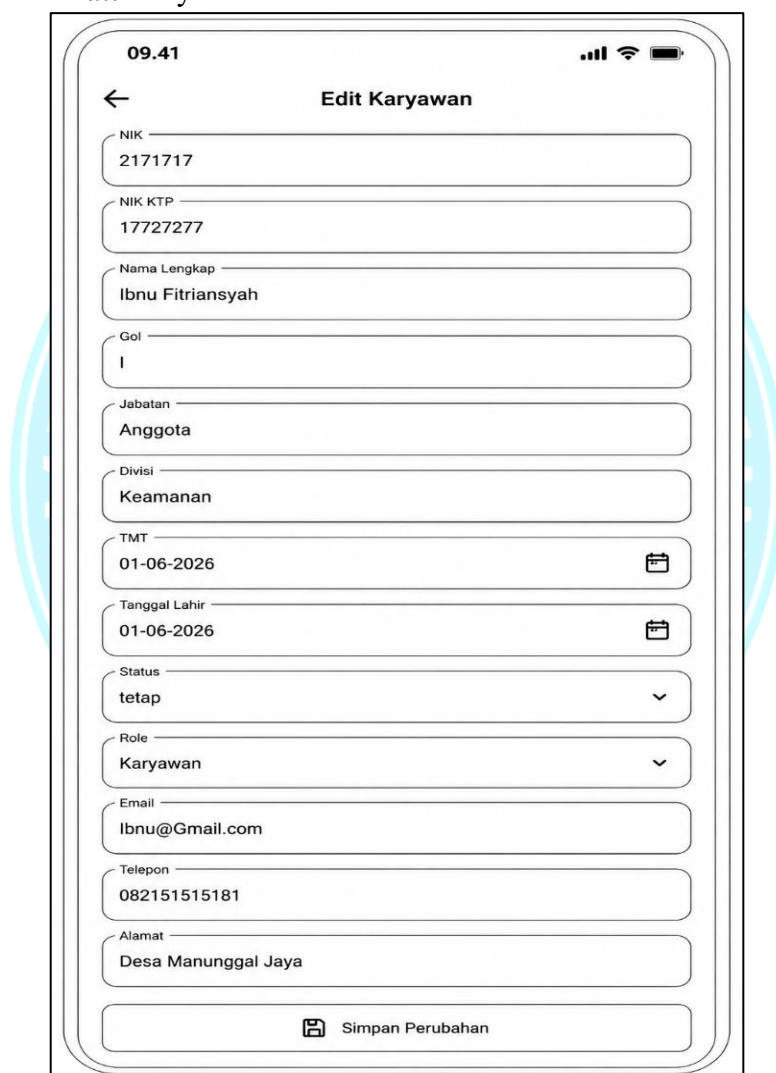
- NIK (text input)
- NIK KTP (text input)
- Nama Lengkap (text input)
- Gol (text input)
- Jabatan (text input)
- Divisi (text input)
- TMT (text input)
- Pilih TMT (dropdown menu with a calendar icon)
- Tanggal Lahir (text input)
- Pilih Tanggal Lahir (dropdown menu with a calendar icon)
- Status (dropdown menu, currently showing "tetap")
- Role (dropdown menu, currently showing "Karyawan")
- Email (text input)
- Password Awal (text input with an eye icon for toggling visibility)
- Telepon (text input)
- Alamat (text input)
- Simpan (button with a save icon)

Sumber: data diolah oleh penulis, 2026

Gambar 4. 37 Halaman Tambah Karyawan

Halaman tambah karyawan merupakan halaman yang digunakan oleh *HRD* untuk menambahkan data karyawan baru ke dalam sistem. Pada halaman ini, *HRD* dapat mengisi berbagai informasi identitas dan data kepegawaian. *Form input* pada halaman ini terdiri dari beberapa *field*, seperti NIK, NIK KTP, Nama Lengkap, golongan, jabatan, divisi, TMT, tanggal lahir, status, *role*, *email*, *password* awal, telepon dan Alamat.

16. Halaman *Edit Karyawan*



The screenshot shows a mobile application interface for editing employee data. The title bar at the top says "Edit Karyawan". The form contains the following fields and values:

Field	Value
NIK	2171717
NIK KTP	17727277
Nama Lengkap	Ibnu Fitriansyah
Gol	I
Jabatan	Anggota
Divisi	Keamanan
TMT	01-06-2026
Tanggal Lahir	01-06-2026
Status	tetap
Role	Karyawan
Email	Ibnu@Gmail.com
Telepon	082151515181
Alamat	Desa Manunggal Jaya

At the bottom of the form is a button labeled "Simpan Perubahan" (Save Changes).

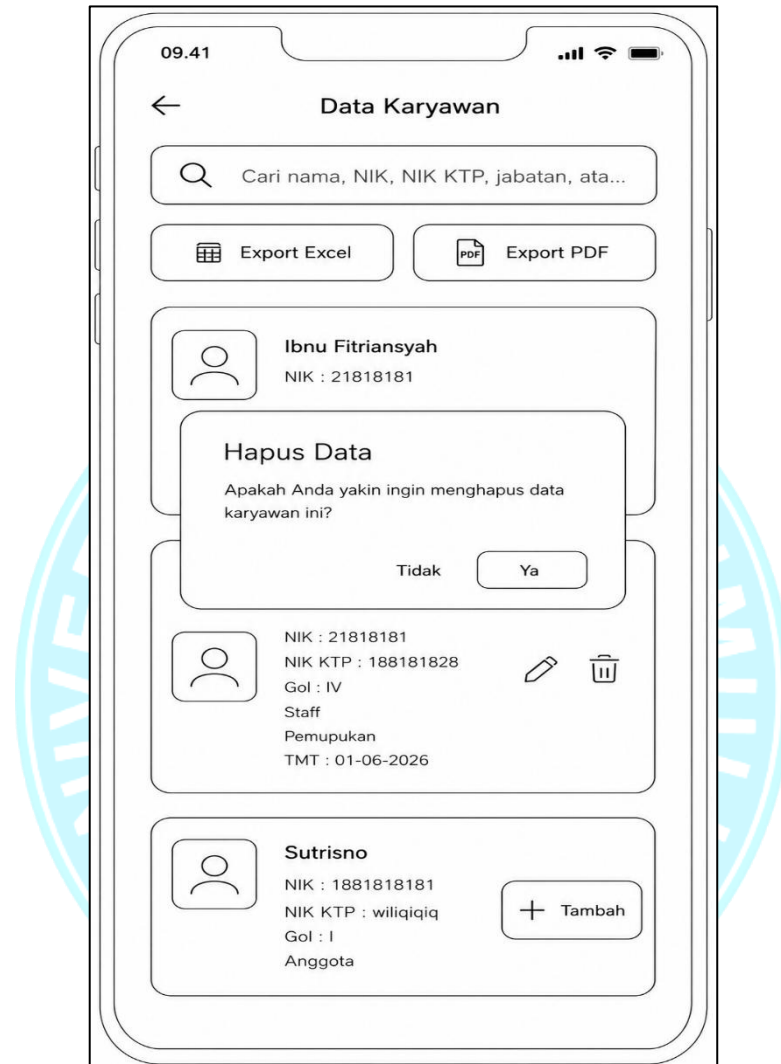
Sumber: data diolah oleh penulis, 2026

Gambar 4. 38 Halaman *Edit Karyawan*

Halaman *edit* karyawan merupakan halaman yang digunakan oleh *HRD* untuk mengubah dan memperbarui data karyawan yang telah tersimpan di dalam sistem.

Pada halaman ini, *HRD* dapat melakukan perubahan data apabila terdapat informasi karyawan yang berubah atau perlu diperbaiki

17. Halaman Hapus Data Karyawan



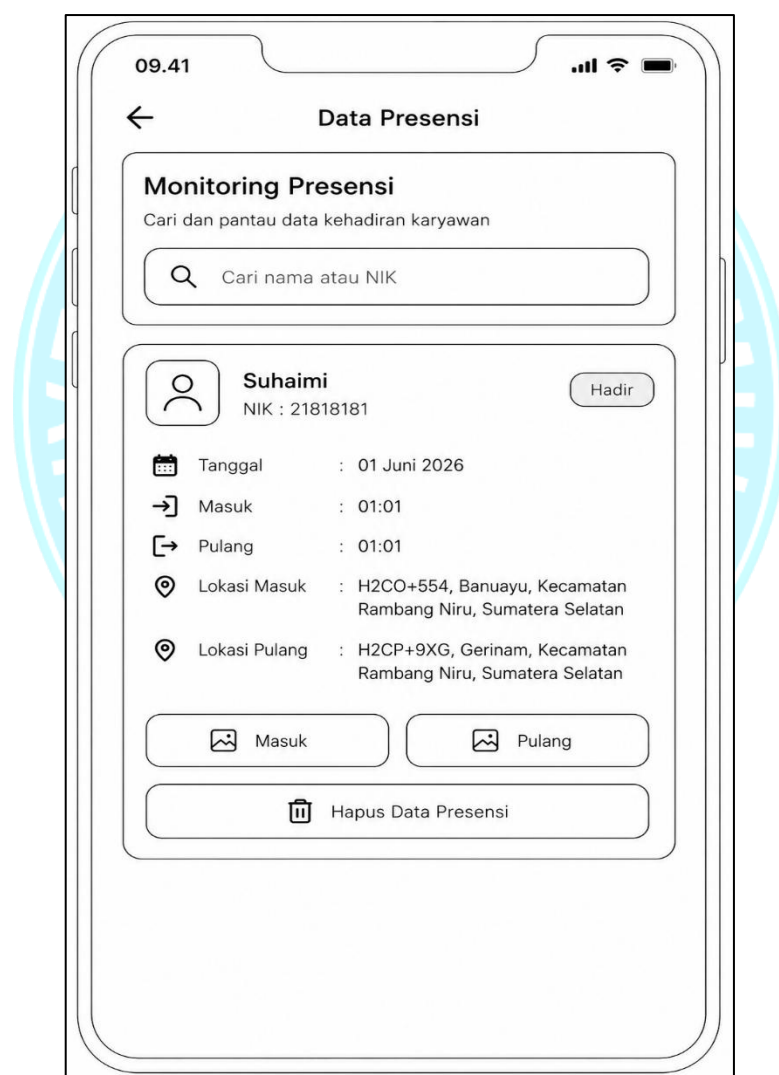
Sumber: data diolah oleh penulis, 2026

Gambar 4. 39 Halaman Hapus Data Karyawan

Halaman hapus data karyawan merupakan halaman yang digunakan oleh *HRD* untuk menghapus data karyawan yang telah tersimpan di dalam sistem. Pada halaman ini, sistem akan menampilkan kotak dialog konfirmasi ketika *HRD* menekan tombol hapus pada data karyawan tertentu. Konfirmasi tersebut bertujuan untuk memastikan bahwa tindakan penghapusan dilakukan dengan sengaja dan menghindari terjadinya penghapusan data secara tidak sengaja.

Pada kotak dialog konfirmasi, sistem menampilkan pesan yang menanyakan apakah *HRD* yakin ingin menghapus data karyawan yang dipilih. Selanjutnya, *HRD* dapat memilih tombol ya untuk melanjutkan proses penghapusan atau tombol tidak untuk membatalkan tindakan tersebut. Jika tombol ya dipilih, maka data karyawan akan dihapus dari *database* dan tidak lagi ditampilkan pada daftar karyawan. Sebaliknya, jika tombol Tidak dipilih, maka proses penghapusan dibatalkan dan data karyawan tetap tersimpan di dalam sistem.

18. Halaman Data Presensi



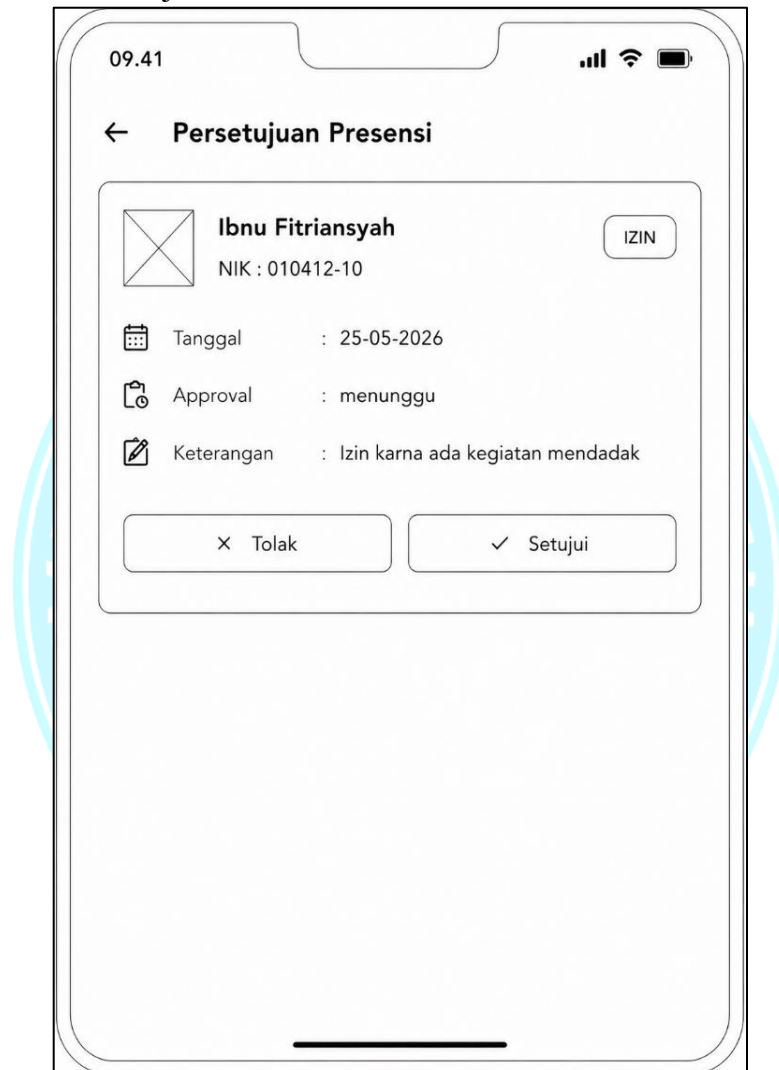
Sumber: data diolah oleh penulis, 2026

Gambar 4. 40 Halaman Data Presensi

Halaman data presensi merupakan halaman yang digunakan oleh *HRD* dan pimpinan untuk memantau data kehadiran karyawan yang tersimpan di dalam

sistem. Pada halaman ini, pengguna dapat melihat informasi presensi karyawan secara detail berdasarkan data yang telah tercatat pada database. Pada bagian atas halaman terdapat fitur pencarian yang digunakan untuk mencari data presensi berdasarkan nama atau NIK karyawan.

19. Halaman Persetujuan Presensi

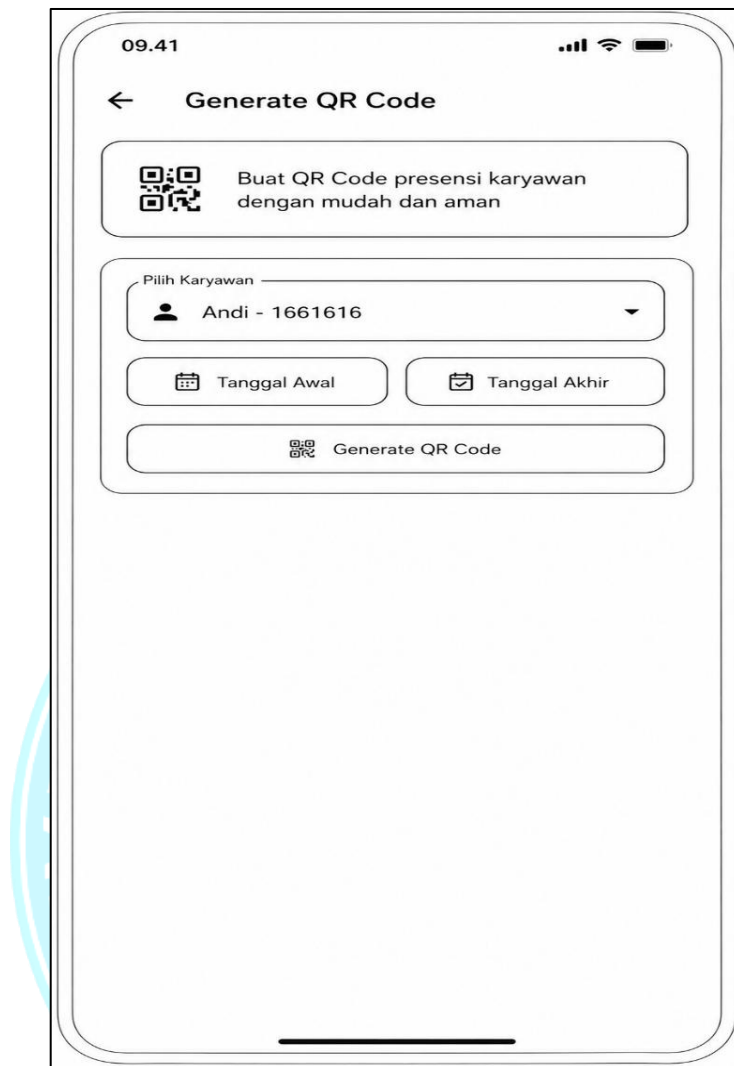


Sumber: data diolah oleh penulis, 2026

Gambar 4. 41 Halaman Persetujuan Presensi

Halaman persetujuan presensi merupakan halaman yang digunakan oleh *HRD* untuk memproses pengajuan presensi karyawan, seperti izin, sakit, dan cuti. Data pengajuan yang ditampilkan pada halaman ini berasal dari tabel pengajuan pada *database*. Pada halaman ini ditampilkan informasi pengajuan karyawan, seperti Nama, NIK, Tanggal, Status *Approval*, dan Keterangan Pengajuan.

20. Halaman *Generate QR Code*



09.41

← Generate QR Code

 Buat QR Code presensi karyawan dengan mudah dan aman

Pilih Karyawan

 Andi - 1661616

 Tanggal Awal

 Tanggal Akhir

 Generate QR Code

Sumber: data diolah oleh penulis, 2026

Gambar 4. 42 Halaman *Generate QR Code*

Pada gambar tampilan *generate QR Code* di atas, dijelaskan bahwa halaman ini digunakan oleh *HRD* untuk membuat *QR Code* yang akan digunakan dalam proses presensi karyawan. Pada halaman ini, *HRD* dapat menentukan rentang tanggal berlakunya *QR Code*, mulai dari tanggal awal hingga tanggal akhir. Selain itu, *HRD* juga dapat mengatur jam berlaku *QR Code*, yaitu waktu mulai dan waktu selesai penggunaan *QR Code* tersebut. Setelah semua data diatur, *HRD* dapat menekan tombol “*Generate QR Code*” untuk membuat *QR Code* presensi. Sistem kemudian akan menampilkan *QR Code* beserta informasi periode dan jam berlakunya. *QR Code* yang dihasilkan dapat diunduh atau dibuat ulang sesuai

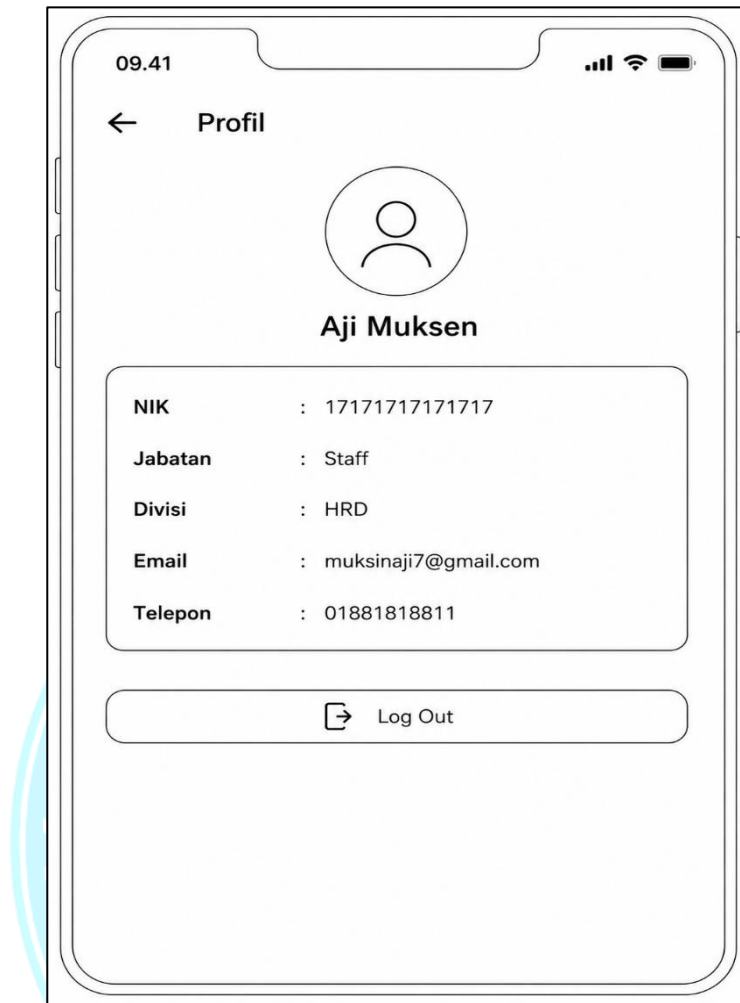
kebutuhan. Halaman ini berfungsi untuk mempermudah pembuatan *QR Code* presensi secara fleksibel dan terjadwal.

21. Halaman Laporan Presensi

Sumber: data diolah oleh penulis, 2026

Gambar 4. 43 Halaman Laporan Presensi

Pada gambar tampilan laporan presensi di atas, dijelaskan bahwa halaman ini digunakan oleh *HRD* dan pimpinan untuk mengunduh laporan kehadiran karyawan. Pada halaman ini, *HRD* dan pimpinan dapat menentukan rentang tanggal laporan yang ingin ditampilkan, mulai dari tanggal awal hingga tanggal akhir. Setelah memilih rentang tanggal, *HRD* dan pimpinan dapat langsung mengunduh laporan dalam *format* yang diinginkan, seperti Excel atau PDF. Tampilan dibuat sederhana agar memudahkan pengguna dalam melakukan proses pengambilan laporan tanpa fitur yang terlalu kompleks. Halaman ini berfungsi untuk mempermudah *HRD* dan pimpinan dalam memperoleh laporan presensi secara cepat dan efisien.

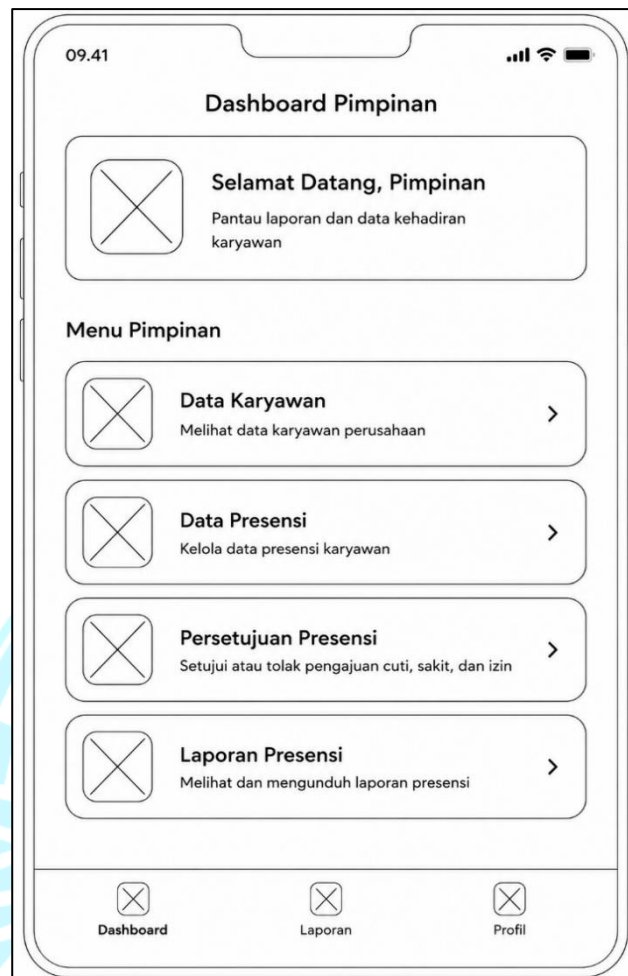
22. Halaman Profil dan *Log Out* HRD

Sumber: data diolah oleh penulis, 2026

Gambar 4. 44 Halaman Profil dan *Logout* HRD

Pada gambar tampilan profil di atas, dijelaskan bahwa halaman ini digunakan untuk menampilkan informasi data diri karyawan seperti nama, NIK, jabatan, divisi, *email* dan telepon. Selain itu, pada halaman ini juga tersedia tombol *logout* yang digunakan untuk keluar dari aplikasi. Ketika pengguna menekan tombol *logout*, maka sistem akan mengakhiri sesi pengguna dan mengarahkan kembali ke halaman *login*. Halaman ini berfungsi sebagai pusat informasi akun sekaligus tempat untuk mengelola akses pengguna dalam aplikasi.

23. Halaman *Dashboard* Pimpinan



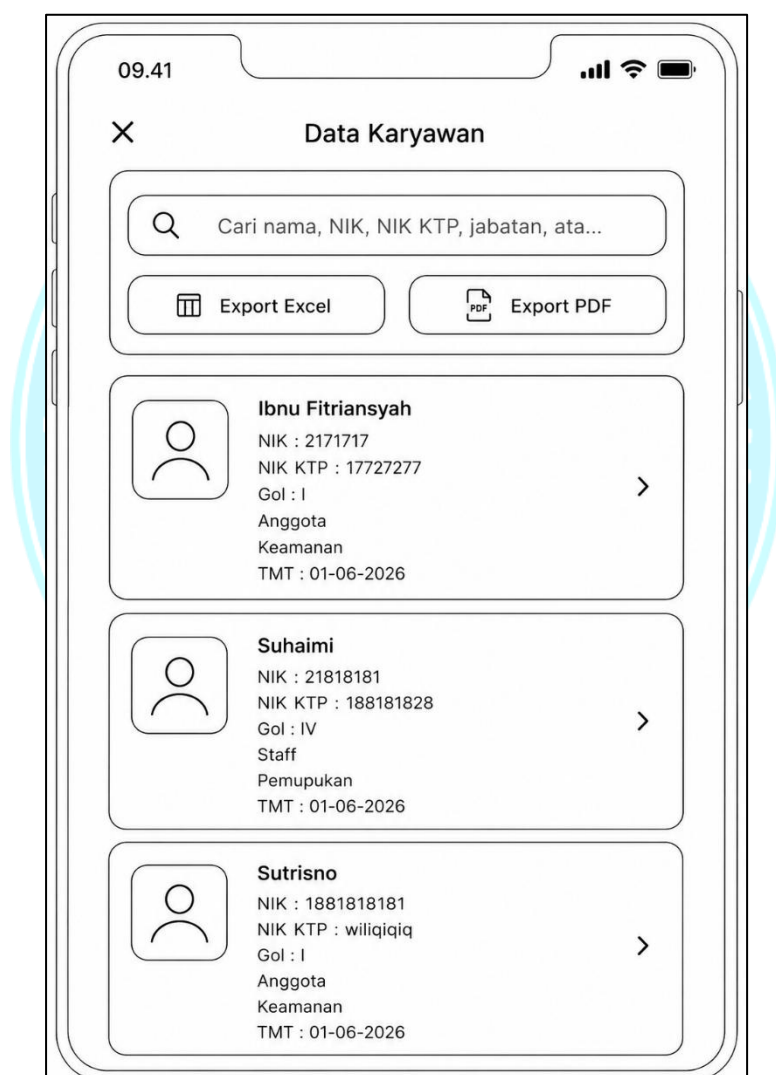
Sumber: data diolah oleh penulis, 2026

Gambar 4. 45 Halaman *Dashboard* Pimpinan

Pada gambar tampilan *dashboard* pimpinan di atas, dijelaskan bahwa halaman ini digunakan oleh pimpinan untuk mengakses informasi terkait data karyawan dan presensi perusahaan. Pada bagian atas halaman ditampilkan informasi sambutan yang menunjukkan bahwa pengguna sedang masuk sebagai pimpinan serta keterangan singkat mengenai fungsi sistem untuk memantau laporan dan data kehadiran karyawan. Selain itu, terdapat beberapa menu utama yang dapat digunakan pimpinan, yaitu Data Karyawan, Data Presensi, Persetujuan Presensi, dan Laporan Presensi. Menu Data Karyawan digunakan untuk melihat informasi data karyawan yang terdaftar dalam sistem. Menu Data Presensi digunakan untuk melihat data kehadiran karyawan yang telah tercatat. Menu Persetujuan Presensi digunakan untuk meninjau dan memberikan persetujuan atau penolakan terhadap

pengajuan sakit, izin, dan cuti yang diajukan oleh karyawan. Sementara itu, menu Laporan Presensi digunakan untuk melihat dan mengunduh laporan kehadiran karyawan. Tampilan dibuat sederhana dan informatif agar memudahkan pimpinan dalam mengakses informasi penting secara cepat tanpa fitur yang terlalu kompleks. Halaman ini berfungsi sebagai pusat navigasi dan *monitoring* bagi pimpinan dalam mengelola serta memantau data karyawan dan presensi perusahaan.

24. Halaman Data Karyawan



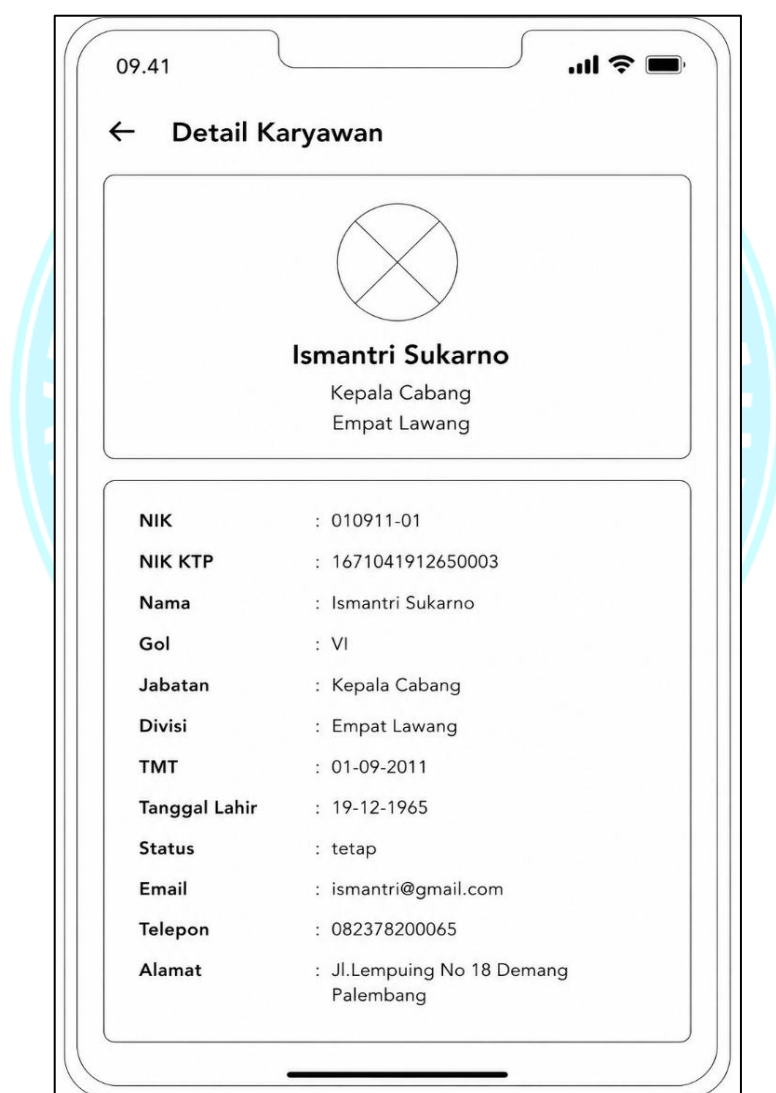
Sumber: data diolah oleh penulis, 2026

Gambar 4. 46 Halaman Data Karyawan

Halaman data karyawan merupakan halaman yang digunakan oleh pimpinan untuk melihat data seluruh karyawan yang terdaftar di dalam sistem. Pada halaman

ini, *pimpinan* dapat melihat, mencari, data karyawan sesuai kebutuhan perusahaan. Pada bagian atas halaman terdapat fitur pencarian yang digunakan untuk mencari data berdasarkan nama, nik, nik ktp, jabatan, atau data lainnya. Selain itu, tersedia tombol *export excel* dan *export pdf* yang digunakan untuk mengunduh data karyawan dalam bentuk laporan *excel* maupun *pdf*. Data karyawan ditampilkan dalam bentuk *card* yang berisi informasi seperti nama, nik, nik ktp, golongan, jabatan, divisi, dan tmt.

25. Halaman Detail Karyawan

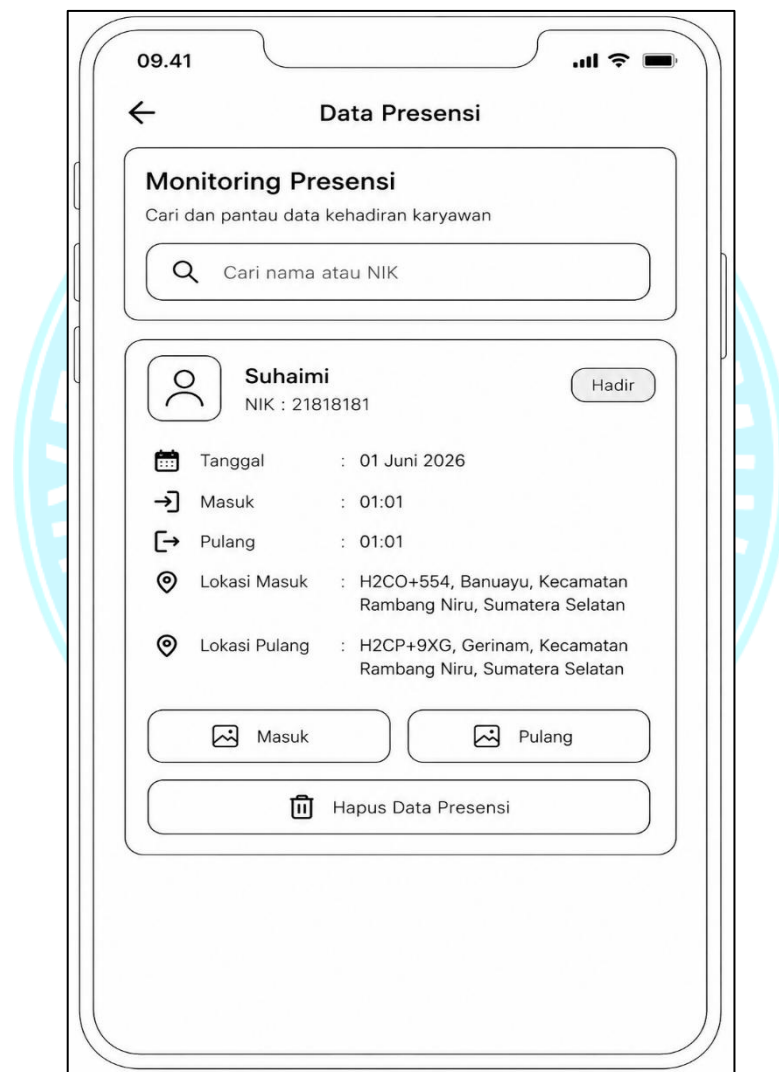


Sumber: data diolah oleh penulis, 2026

Gambar 4. 47 Halaman Detail Karyawan

Halaman detail karyawan merupakan halaman yang digunakan untuk menampilkan informasi lengkap data karyawan yang tersimpan di dalam sistem. Halaman ini membantu *HRD* maupun pimpinan dalam melihat detail identitas dan data kepegawaian setiap karyawan secara lebih lengkap. Pada bagian atas halaman terdapat *card* informasi karyawan yang menampilkan foto profil *default*, nama karyawan, jabatan, dan divisi kerja.

26. Halaman Data Presensi



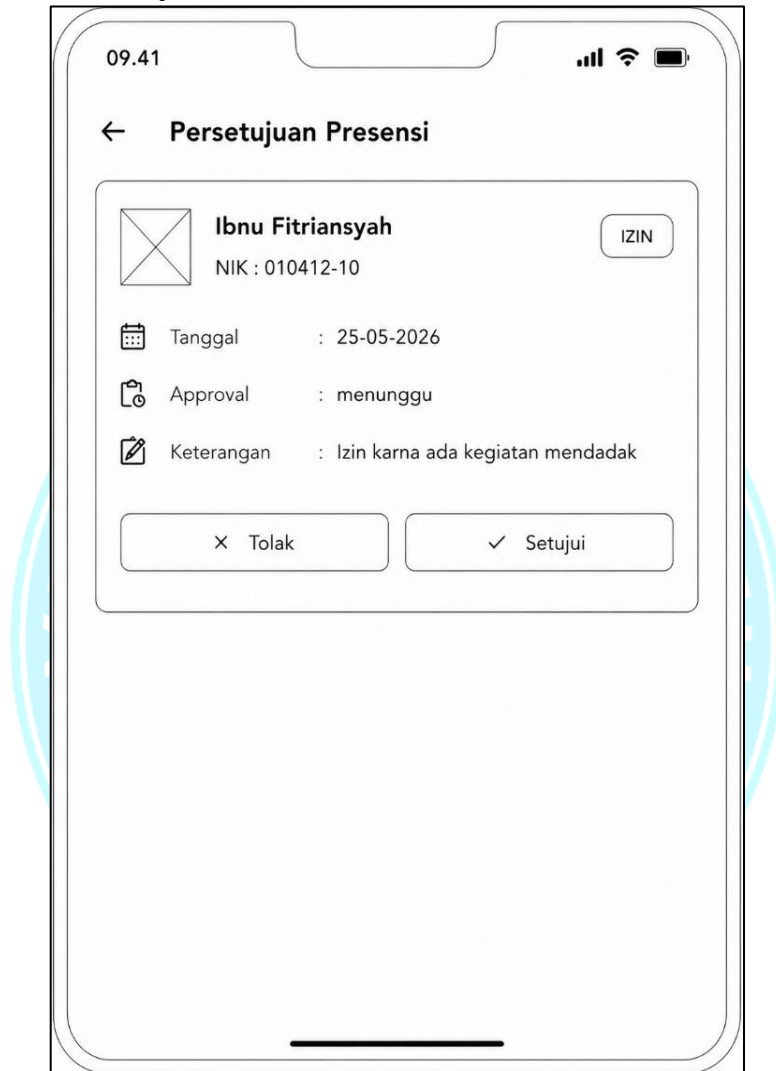
Sumber: data diolah oleh penulis, 2026

Gambar 4. 48 Halaman Data Presensi

Halaman data presensi merupakan halaman yang digunakan oleh *HRD* dan pimpinan untuk memantau data kehadiran karyawan yang tersimpan di dalam sistem. Pada halaman ini, pengguna dapat melihat informasi presensi karyawan

secara detail berdasarkan data yang telah tercatat pada database. Pada bagian atas halaman terdapat fitur pencarian yang digunakan untuk mencari data presensi berdasarkan nama atau NIK karyawan.

27. Halaman Persetujuan Presensi



Sumber: data diolah oleh penulis, 2026

Gambar 4. 49 Halaman Persetujuan Presensi

Halaman persetujuan presensi merupakan halaman yang digunakan oleh HRD dan pimpinan untuk memproses pengajuan presensi karyawan, seperti izin, sakit, dan cuti. Data pengajuan yang ditampilkan pada halaman ini berasal dari tabel pengajuan pada *database*. Pada halaman ini ditampilkan informasi pengajuan karyawan, seperti Nama, NIK, Tanggal, Status *Approval*, dan Keterangan Pengajuan.

28. Halaman Laporan Presensi

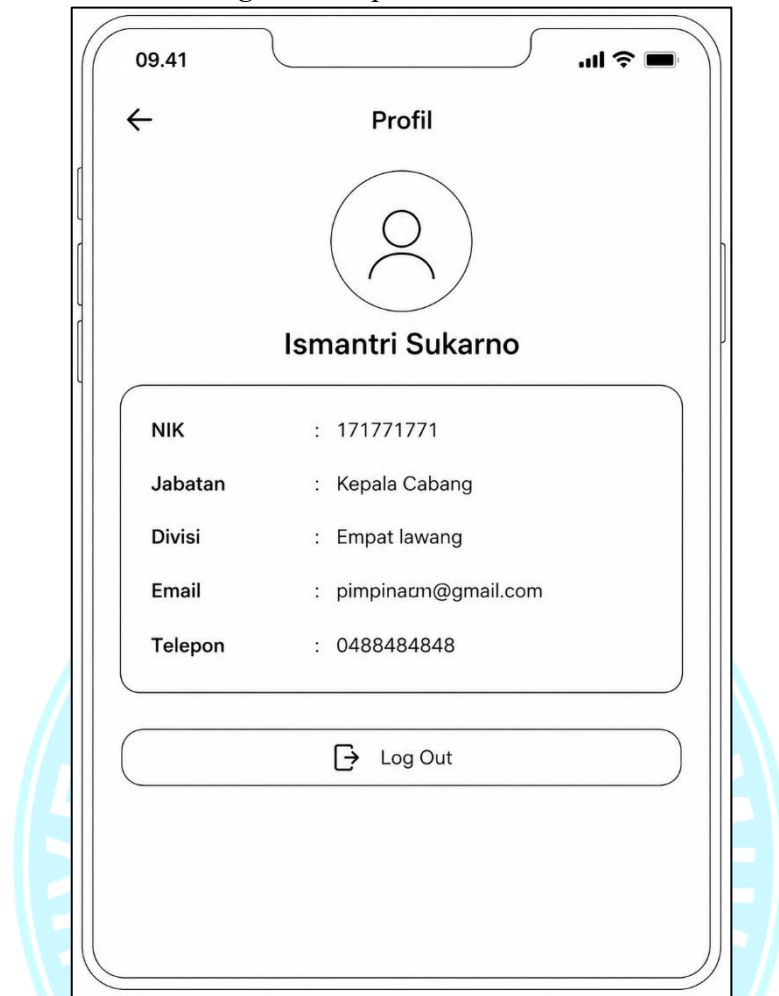
The screenshot displays a mobile application interface for generating attendance reports. At the top, the status bar shows the time as 09.41 and signal indicators. The app title is 'Laporan Presensi'. Below the title is a 'Filter Laporan' section containing several interactive elements: a dropdown for 'Pilih Karyawan' currently showing 'Semua Karyawan', a dropdown for 'Jenis Laporan' showing 'Bulanan', and two date range selectors labeled 'Bulan' and 'Tahun' with 'Pilih Bulan' and 'Pilih Tahun' respectively. At the bottom of the filter section are two buttons with 'Excel' and 'PDF' labels, each preceded by a small 'X' icon in a square.

Sumber: data diolah oleh penulis, 2026

Gambar 4. 50 Halaman Laporan Presensi

Pada gambar tampilan laporan presensi di atas, dijelaskan bahwa halaman ini digunakan oleh *HRD* dan pimpinan untuk mengunduh laporan kehadiran karyawan. Pada halaman ini, *HRD* dan pimpinan dapat menentukan rentang tanggal laporan yang ingin ditampilkan, mulai dari tanggal awal hingga tanggal akhir. Setelah memilih rentang tanggal, *HRD* dan pimpinan dapat langsung mengunduh laporan dalam *format* yang diinginkan, seperti Excel atau PDF. Tampilan dibuat sederhana agar memudahkan pengguna dalam melakukan proses pengambilan laporan tanpa fitur yang terlalu kompleks. Halaman ini berfungsi untuk mempermudah *HRD* dan pimpinan dalam memperoleh laporan presensi secara cepat dan efisien.

29. Halaman Profil dan *Log Out* Pimpinan



Sumber: data diolah oleh penulis, 2026

Gambar 4. 51 Halaman Profil dan *Logout* Pimpinan

Pada gambar tampilan profil di atas, dijelaskan bahwa halaman ini digunakan untuk menampilkan informasi data diri karyawan seperti nama, NIK, jabatan, divisi, *email* dan telepon. Selain itu, pada halaman ini juga tersedia tombol *logout* yang digunakan untuk keluar dari aplikasi. Ketika pengguna menekan tombol *logout*, maka sistem akan mengakhiri sesi pengguna dan mengarahkan kembali ke halaman *login*. Halaman ini berfungsi sebagai pusat informasi akun sekaligus tempat untuk mengelola akses pengguna dalam aplikasi.

BAB V

HASIL DAN PEMBAHASAN

5.1 Spesifikasi Perangkat

Dalam penelitian ini, pengembangan aplikasi didukung oleh pemanfaatan berbagai perangkat pada setiap tahapannya. Perangkat tersebut berperan penting mulai dari proses perancangan, penulisan kode program, hingga tahap pengujian sistem untuk memastikan aplikasi berjalan dengan baik dan sesuai dengan kebutuhan pengguna. Penggunaan perangkat yang tepat dan sesuai kebutuhan tidak hanya meningkatkan efisiensi proses pengembangan, tetapi juga membantu meminimalkan kesalahan serta mempercepat waktu penyelesaian aplikasi. Selain itu, pemilihan perangkat yang optimal turut mendukung kestabilan dan performa sistem yang dihasilkan. Adapun perangkat yang digunakan mencakup dua kategori utama, yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*), yang saling terintegrasi dan mendukung dalam keseluruhan proses pengembangan aplikasi.

5.1.1 Perangkat Keras yang Digunakan

Perangkat keras yang digunakan dalam penelitian ini merupakan perangkat yang mendukung proses perancangan, pengembangan, dan pengujian aplikasi digitalisasi sistem informasi karyawan dan presensi menggunakan *qr code* berbasis *android*. Perangkat keras tersebut digunakan untuk menjalankan aplikasi, melakukan pengujian fitur presensi *qr code*, *gps*, *selfie*, serta memastikan seluruh fungsi aplikasi dapat berjalan dengan baik pada perangkat *android*.

Kinerja perangkat keras sangat berpengaruh terhadap proses pengembangan aplikasi, kecepatan pengujian sistem, serta stabilitas aplikasi saat digunakan oleh pengguna. Oleh karena itu, pemilihan perangkat keras yang sesuai menjadi faktor penting dalam mendukung keberhasilan implementasi sistem. Adapun spesifikasi perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut:

1. *Laptop* HP 14s-fq2002au
2. *Processor*: AMD Ryzen 5 5625U

3. *RAM: 8 GB*
4. *Storage: 512 GB*
5. *Keyboard*
6. *Mouse*
7. *Handphone Android OS 16*
8. *Kabel Usb type c*

5.1.2 Perangkat Lunak yang Digunakan

Perangkat lunak dalam penelitian ini berfungsi sebagai komponen utama yang mendukung proses perancangan, pengembangan, dan implementasi sistem. Perangkat lunak tersebut digunakan dalam kegiatan penulisan kode program, pengolahan data, serta pengujian aplikasi yang telah dikembangkan. Pemilihan perangkat lunak yang tepat memungkinkan sistem dapat berjalan secara optimal dan stabil sesuai dengan kebutuhan pengembangan. Oleh karena itu, perangkat lunak memiliki peran penting dalam memastikan seluruh fitur aplikasi dapat berfungsi dengan baik. Adapun perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut.

1. *Sistem Operasi: Windows 11 Home Single Language*
2. *Bahasa Pemrograman: Dart SDK 3.7.2*
3. *Framework: Flutter 3.29.3*
4. *Firebase_core: 3.6.0*
5. *Cloud_firestore: 5.4.4*
6. *Firebase_auth: 5.3.1*
7. *Code Editor: Visual Studio Code 1.121.0*
8. *Cloudinary: 1.1*
9. *Mobile Scanner: 5.1.1*
10. *Image Picker: 1.0.7*
11. *QR Flutter: 4.1.0*

5.2 Implementasi Database

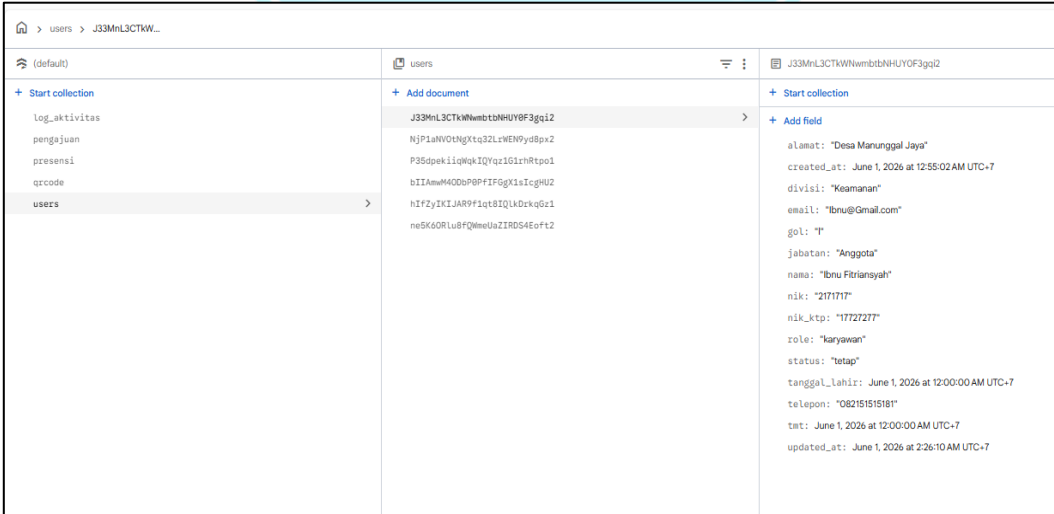
Database pada aplikasi digitalisasi sistem informasi karyawan dan presensi menggunakan *qr code* berbasis *android* digunakan untuk menyimpan, mengelola,

dan menghubungkan seluruh data yang terdapat pada sistem. *Database* ini berperan penting dalam mendukung proses pengelolaan data karyawan, presensi, pengajuan izin, serta aktivitas pengguna agar sistem dapat berjalan secara efektif dan terorganisir. Pada penelitian ini digunakan *firebase firestore* sebagai media penyimpanan *database* karena memiliki kemampuan penyimpanan data secara *realtime*.

Struktur *database* pada sistem disusun dalam beberapa *collection* utama, yaitu *users*, *presensi*, *qrcode*, *pengajuan*, dan *log_aktivitas*. *Collection users* digunakan untuk menyimpan data akun dan informasi karyawan, *collection presensi* digunakan untuk menyimpan data kehadiran karyawan, *collection qrcode* digunakan untuk menyimpan data *qr code* presensi, *collection pengajuan* digunakan untuk menyimpan data pengajuan sakit, izin, dan cuti, sedangkan *collection log_aktivitas* digunakan untuk mencatat seluruh aktivitas penting pengguna di dalam sistem.

Penggunaan *database* ini membantu sistem dalam mengurangi terjadinya duplikasi data, mempermudah proses pengelolaan informasi, serta mendukung penyajian laporan presensi secara *realtime*. Selain itu, aplikasi ini juga menggunakan *cloudinary* sebagai media penyimpanan gambar, seperti foto *selfie* presensi dan dokumentasi pendukung lainnya yang diunggah oleh pengguna.

5.2.1 Implementasi Tabel *Users*



Collection	Document ID	Fields
users	J33MnL3CTkNmbtBNHUY6F3gqI2	- alamat: "Desa Manunggal Jaya" - created_at: June 1, 2026 at 12:55:02 AM UTC+7 - divisi: "Keamanan" - email: "Ibnu@gmail.com" - gol: "T" - jabatan: "Anggota" - nama: "Ibnu Fitrianyah" - nik: "2171717" - nik_ktp: "17727277" - role: "karyawan" - status: "tetap" - tanggal_lahir: June 1, 2026 at 12:00:00 AM UTC+7 - telepon: "082151515181" - tmt: June 1, 2026 at 12:00:00 AM UTC+7 - updated_at: June 1, 2026 at 2:26:10 AM UTC+7

Sumber: data diolah oleh penulis, 2026

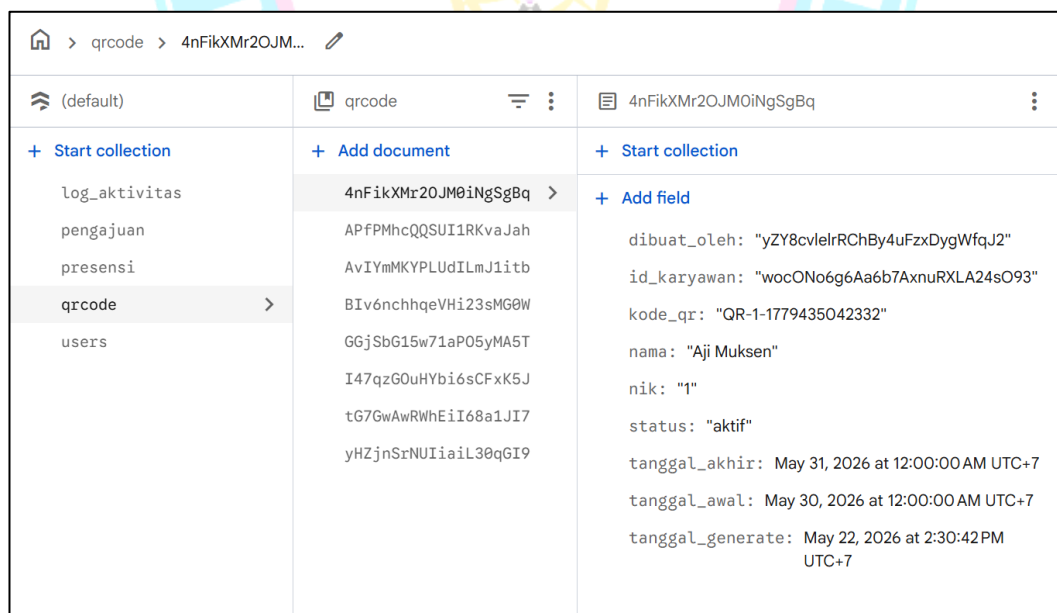
Gambar 5. 1 Tabel *Users*

Collection users pada *database firebase firestore* digunakan untuk menyimpan seluruh data pengguna dan data karyawan yang memiliki hak akses ke dalam aplikasi digitalisasi sistem informasi karyawan dan presensi menggunakan *qr code* berbasis *android*. Setiap data pengguna disimpan dalam bentuk *document* dengan *document id* yang dibuat otomatis oleh *firebase firestore* sehingga memudahkan proses pengelolaan dan pencarian data di dalam sistem.

Struktur data pada *collection users* terdiri dari beberapa *field* penting, seperti nama, nik, nik_ktp, *email*, telepon, alamat, jabatan, divisi, gol, status, tanggal_lahir, tmt, dan *role*. *Field role* digunakan untuk menentukan hak akses pengguna, seperti *hrd*, pimpinan, dan karyawan, sedangkan *field* lainnya digunakan untuk menyimpan informasi identitas dan data kepegawaian karyawan.

Selain itu, *collection users* juga memiliki *field created_at* dan *updated_at* untuk mencatat waktu pembuatan dan pembaruan data. Berdasarkan struktur tersebut, *collection users* digunakan sebagai pusat data utama dalam proses *login*, pengelolaan data karyawan, presensi *qr code*, pengajuan, dan pembuatan laporan pada aplikasi.

5.2.2 Implementasi Tabel QR Code



Collection	Document ID	Fields
qrcode	4nFikXMr20JM0iNgSgBq	- dibuat_oleh: "yZY8cvlelrRChBy4uFzxDygWfqJ2" - id_karyawan: "wocONo6g6Aa6b7AxnuRXLA24sO93" - kode_qr: "QR-1-1779435042332" - nama: "Aji Muksen" - nik: "1" - status: "aktif" - tanggal_akhir: May 31, 2026 at 12:00:00 AM UTC+7 - tanggal_awal: May 30, 2026 at 12:00:00 AM UTC+7 - tanggal_generate: May 22, 2026 at 2:30:42 PM UTC+7

Sumber: data diolah oleh penulis, 2026

Gambar 5. 2 Tabel QR Code

Collection qrcode pada *database firebase firestore* digunakan untuk menyimpan data *qr code* presensi karyawan yang digunakan dalam proses presensi masuk dan pulang pada aplikasi. Setiap data *qr code* disimpan dalam bentuk *document* dengan *document id* yang dibuat secara otomatis oleh *firebase firestore* sehingga memudahkan pengelolaan dan pencarian data *qr code* di dalam sistem.

Struktur data pada *collection qrcode* terdiri dari beberapa *field*, seperti *kode_qr*, *id_karyawan*, *nama*, *nik*, *status*, *tanggal_awal*, *tanggal_akhir*, *tanggal_generate*, dan *dibuat_oleh*. *Field kode_qr* digunakan sebagai kode utama *qr code* presensi, sedangkan *id_karyawan*, *nama*, dan *nik* digunakan untuk menghubungkan *qr code* dengan data karyawan pada *collection users*.

Field status digunakan untuk menunjukkan status *qr code*, yaitu aktif atau nonaktif. Selain itu, *field tanggal_awal* dan *tanggal_akhir* digunakan untuk menentukan masa berlaku *qr code*, sedangkan *field tanggal_generate* digunakan untuk mencatat waktu pembuatan *qr code*. Berdasarkan struktur tersebut, *collection qrcode* digunakan untuk mendukung proses presensi *qr code* agar lebih aman, teratur, dan mudah dikelola oleh pihak *hrd*.

5.2.3 Implementasi Tabel Presensi

Collection	Document ID	Fields
presensi	NPtaW0tngtq3ZLrWENydbp2	approval_status: "disetujui" foto_selfie_masuk: "https://res.cloudinary.com/difjydu/image/upload/v1780250471/presensi_selfie/cmbh3ef55mliw4.jpg" foto_selfie_pulang: "https://res.cloudinary.com/difjydu/image/upload/v1780250497/presensi_selfie/ug0g4ufycmcsam4p3u.jpg" kode_qr: "QR-21818181-1780250289581" latitude_masuk: -3.4295534 latitude_pulang: -3.4296067 lokasi_masuk: "H200-554, Barusayu, Kecamatan Rambang Niru, Sumatera Selatan" lokasi_pulang: "H200-9KG, Gerikam, Kecamatan Rambang Niru, Sumatera Selatan" longitude_masuk: 104.0375855 longitude_pulang: 104.0373354 masuk: "01:01" metode: "QR Code + Selfie + GPS" nama: "Suhaimi" nik: "21818181" pulang: "01:01" status: "Ya" tanggal: "June 1, 2026 at 10:12 AM UTC+7" tanggal_string: "2026-06-01" uid: "NPtaW0tngtq3ZLrWENydbp2"

Sumber: data diolah oleh penulis, 2026

Gambar 5. 3 Tabel Presensi

Collection presensi pada *database firebase firestore* digunakan untuk menyimpan seluruh data kehadiran karyawan pada aplikasi. Data presensi ini mencakup presensi masuk, presensi pulang, serta hasil persetujuan pengajuan sakit, izin, dan cuti yang telah diproses oleh *hrd*. Setiap data presensi disimpan dalam bentuk *document* dengan *document id* sehingga memudahkan pengelolaan dan pencarian data di dalam sistem.

Struktur data pada *collection* presensi terdiri dari beberapa *field*, seperti *uid*, *nama*, *nik*, *kode_qr*, *tanggal*, *tanggal_string*, *masuk*, *pulang*, *status*, *approval_status*, dan *metode*. *Field* *kode_qr* digunakan untuk menyimpan *qr code* yang digunakan saat presensi, sedangkan *field* *masuk* dan *pulang* digunakan untuk menyimpan waktu kehadiran karyawan. *Field* *status* digunakan untuk menunjukkan status kehadiran, seperti *hadir*, *sakit*, *izin*, *cuti*, atau *alpha*.

Selain itu, *collection* ini juga menyimpan data lokasi *gps* melalui *field* *latitude_masuk*, *longitude_masuk*, *lokasi_masuk*, *latitude_pulang*, *longitude_pulang*, dan *lokasi_pulang*. *Field* *foto_selfie_masuk* dan *foto_selfie_pulang* digunakan untuk menyimpan foto *selfie* presensi yang tersimpan pada *cloudinary*. Berdasarkan struktur tersebut, *collection* presensi digunakan untuk mendukung proses *monitoring* kehadiran karyawan secara *realtime*, akurat, dan lebih aman dengan kombinasi *qr code*, *selfie*, dan *gps*.

5.2.4 Implementasi Tabel Pengajuan

pengajuan > EjqdKINRyXB...		
(default)	pengajuan	EjqdKINRyXBLLJZE6DVE6MvaOn2_2026-05-25_sakit
+ Start collection	+ Add document	+ Start collection
log_aktivitas	EjqdKINRyXBLLJZE6DVE6MvaOn2_2026-05-25_sakit	+ Add field
pengajuan	IAkdyK4BcMn77aB2Ys2jK7yC83_2026-05-25_cuti	approval_status: "disetujui"
presensi	IAkdyK4BcMn77aB2Ys2jK7yC83_2026-05-25_cuti	dibuat_oleh: "EjqdKINRyXBLLJZE6DVE6MvaOn2"
qr_code	IAkdyK4BcMn77aB2Ys2jK7yC83_2026-05-25_cuti	diproses_oleh: "yZYBoVleRCHBy4uFzDyGwFqJ2"
users	J33MhL3CTkMwbtbNHUyF3gi2_2026-06-01_izin	jenis_pengajuan: "sakit"
		keterangan: "sakit kepala"
		nama: "Dimas kurnia"
		nik: "2"
		tanggal: May 25, 2026 at 12:51 PM UTC+7
		tanggal_pengajuan: May 25, 2026 at 12:59 PM UTC+7
		tanggal_persetujuan: May 25, 2026 at 12:50 PM UTC+7
		tanggal_string: "2026-05-25"
		uid: "EjqdKINRyXBLLJZE6DVE6MvaOn2"

Sumber: data diolah oleh penulis, 2026

Gambar 5. 4 Tabel Pengajuan

Collection pengajuan pada *database firebase firestore* digunakan untuk menyimpan data pengajuan karyawan, seperti pengajuan sakit, izin, dan cuti pada aplikasi. Setiap data pengajuan disimpan dalam bentuk *document* dengan *document id* sehingga memudahkan proses pengelolaan, pencarian, dan persetujuan data oleh pihak *hrd*.

Struktur data pada *collection* pengajuan terdiri dari beberapa *field*, seperti *uid*, *nama*, *nik*, *jenis_pengajuan*, *keterangan*, *tanggal*, *tanggal_pengajuan*, *approval_status*, *dibuat_oleh*, *diproses_oleh*, dan *tanggal_persetujuan*. *Field jenis_pengajuan* digunakan untuk menunjukkan jenis pengajuan yang dilakukan karyawan, seperti sakit, izin, atau cuti, sedangkan *field keterangan* digunakan untuk menyimpan alasan pengajuan yang diberikan oleh karyawan.

Field approval_status digunakan untuk menunjukkan status persetujuan pengajuan, seperti menunggu, disetujui, atau ditolak. Selain itu, *field dibuat_oleh* digunakan untuk menyimpan identitas pengguna yang membuat pengajuan, sedangkan *field diproses_oleh* digunakan untuk menyimpan identitas *hrd* yang memproses pengajuan tersebut. *Field tanggal_pengajuan* dan *tanggal_persetujuan* digunakan untuk mencatat waktu pengajuan dan waktu persetujuan data.

Berdasarkan struktur tersebut, *collection* pengajuan digunakan untuk mendukung proses pengajuan dan persetujuan presensi karyawan secara digital sehingga proses administrasi menjadi lebih teratur, cepat, dan mudah dipantau oleh *hrd* maupun pimpinan.

5.2.5 Implementasi Tabel *Log* Aktivitas

log_aktivitas > BVBPZwvb4Ee3...		
(default)	log_aktivitas	BVBPZwvb4Ee3DFSgAiQL
+ Start collection	+ Add document	+ Start collection
log_aktivitas >	BVBPZwvb4Ee3DFSgAiQL >	+ Add field
pengajuan	H91gKlp8bNbjvcIOtLIC	aktivitas: "Generate QR Code"
presensi	UFLbTLjeKNf4qA34KE4i	keterangan: "HRD membuat QR Code untuk Andi dengan NIK 1661616"
qrcode		nama: "Aji Muksen"
users		role: "hrd"
		uid: "bllAmwM4ODbPOPfIFGgX1slcgHU2"
		waktu: June 2, 2026 at 3:29:05 AM UTC+7

Sumber: data diolah oleh penulis, 2026

Gambar 5. 5 Tabel *Log* Aktivitas

Collection *log_aktivitas* pada *database firebase firestore* digunakan untuk menyimpan riwayat aktivitas pengguna di dalam aplikasi. *Collection* ini berfungsi untuk mencatat berbagai aktivitas penting yang dilakukan pengguna, terutama aktivitas yang dilakukan oleh *hrd* dalam proses pengelolaan sistem. Struktur data pada *collection log_aktivitas* terdiri dari beberapa *field*, seperti aktivitas, keterangan, nama, *role*, uid, dan waktu. *Field* aktivitas digunakan untuk menyimpan jenis aktivitas yang dilakukan pengguna, seperti *generate qr code*, persetujuan pengajuan, maupun pengelolaan data karyawan.

Field keterangan digunakan untuk menyimpan penjelasan detail dari aktivitas yang dilakukan. Selain itu, *field* nama, *role*, dan uid digunakan untuk menyimpan identitas pengguna yang melakukan aktivitas di dalam sistem. *field* waktu digunakan untuk mencatat waktu terjadinya aktivitas tersebut. Berdasarkan struktur tersebut, *collection log_aktivitas* digunakan untuk membantu proses *monitoring*, pengawasan, dan pencatatan aktivitas pengguna agar pengelolaan sistem menjadi lebih teratur dan mudah ditelusuri.

5.3 Tampilan Antarmuka

5.3.1 Tampilan Halaman *Splash Screen*



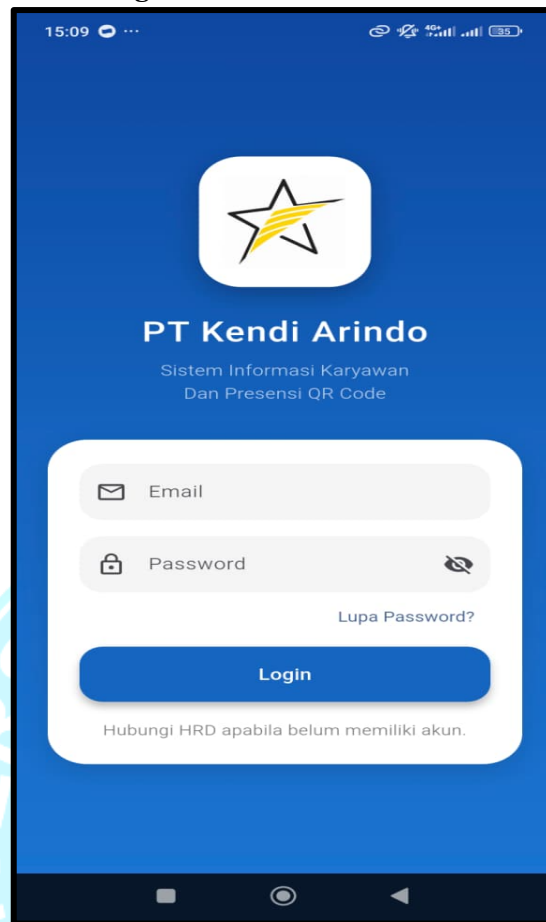
Sumber: data diolah oleh penulis, 2026

Gambar 5. 6 Halaman *Splash Screen*

Halaman *splash screen* merupakan tampilan awal yang muncul saat aplikasi “digitalisasi sistem informasi karyawan dan presensi menggunakan *qr code* berbasis *android*” dijalankan. Tampilan ini berfungsi sebagai halaman pembuka sebelum pengguna masuk ke halaman *login* atau halaman utama aplikasi. Pada halaman *splash screen* ditampilkan logo perusahaan pt kendi arindo sebagai identitas aplikasi, disertai dengan nama sistem yaitu “sistem informasi karyawan dan presensi *qr code*”.

Selain itu, terdapat indikator *loading* berbentuk lingkaran yang menunjukkan bahwa aplikasi sedang melakukan proses inisialisasi dan pengecekan data sebelum masuk ke halaman berikutnya. Tampilan *splash screen* menggunakan kombinasi warna biru dan putih yang memberikan kesan *modern*, sederhana, dan profesional sehingga nyaman dilihat oleh pengguna saat aplikasi dijalankan.

5.3.2 Tampilan Halaman *Login*



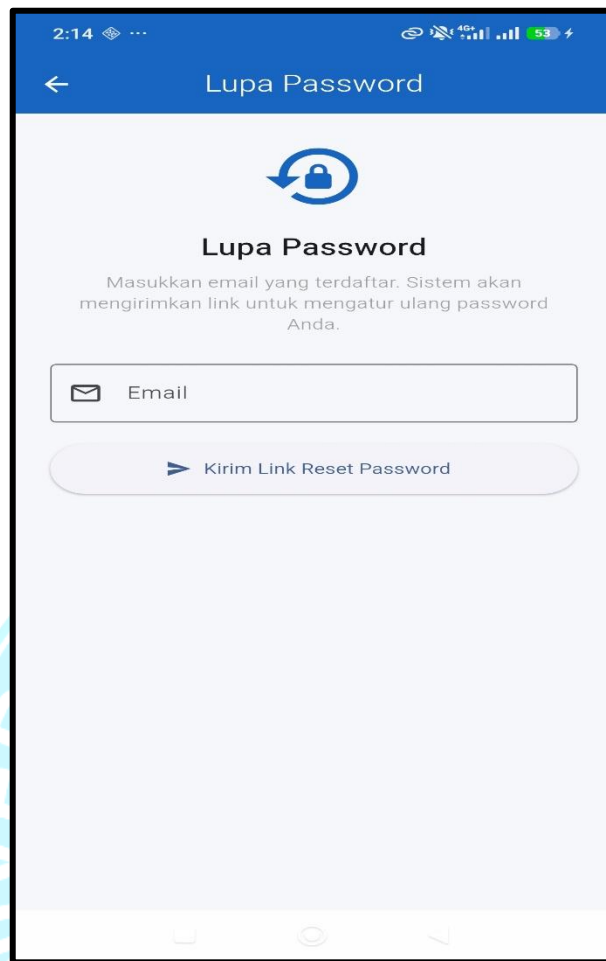
Sumber: data diolah oleh penulis, 2026

Gambar 5. 7 Halaman *Login*

Halaman pertama yang ditampilkan saat aplikasi “digitalisasi sistem informasi karyawan dan presensi menggunakan *qr code* berbasis *android*” dijalankan adalah halaman *login*. Tampilan halaman ini dirancang sederhana dan *modern* untuk memudahkan pengguna melakukan proses autentikasi sebelum masuk ke dalam sistem.

Pada bagian atas halaman ditampilkan logo pt kendi arindo sebagai identitas aplikasi, diikuti dengan nama sistem “sistem informasi karyawan dan presensi *qr code*”. Selanjutnya terdapat formulir *login* yang terdiri dari *field email* dan *password*, serta ikon untuk menampilkan atau menyembunyikan kata sandi. Di bawah formulir terdapat fitur “*lupa password?*” dan tombol *login* untuk memproses data pengguna. apabila data yang dimasukkan benar, maka pengguna akan diarahkan ke halaman utama sesuai dengan hak aksesnya, seperti *hrd*, pimpinan, atau karyawan.

5.3.3 Tampilan Halaman Lupa Password



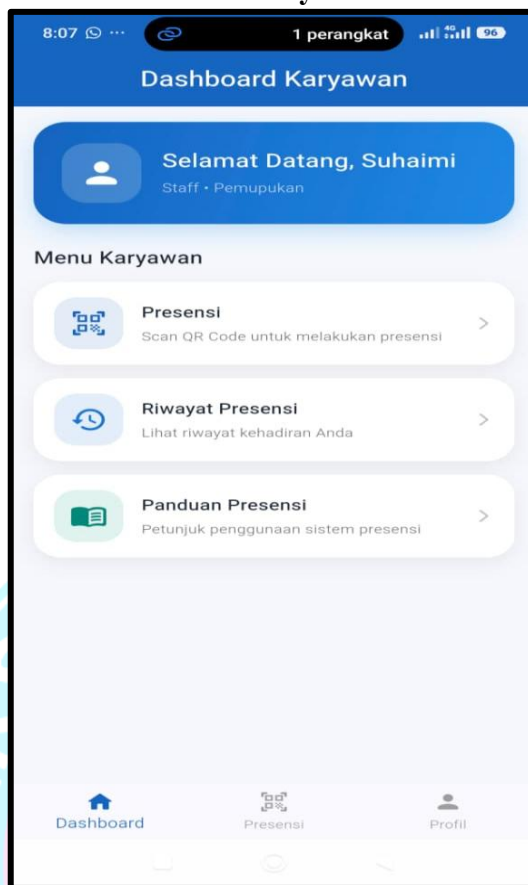
Sumber: data diolah oleh penulis, 2026

Gambar 5. 8 Halaman Lupa Password

Halaman lupa password merupakan halaman yang digunakan pengguna untuk melakukan proses *reset password* apabila lupa kata sandi akun. Pada halaman ini, pengguna dapat memasukkan *email* yang telah terdaftar pada sistem untuk menerima *link reset password* melalui *email*.

Pada bagian tengah halaman ditampilkan ikon *reset password* beserta judul *reset password* sebagai informasi fungsi halaman. Di bawahnya terdapat penjelasan singkat mengenai proses pengiriman *link reset password* oleh sistem. Selanjutnya terdapat *field email* yang digunakan pengguna untuk memasukkan alamat *email* akun yang terdaftar. Setelah *email* diisi, pengguna dapat menekan tombol kirim *link reset* untuk mengirim permintaan *reset password*.

5.3.4 Tampilan Halaman *Dashboard* Karyawan



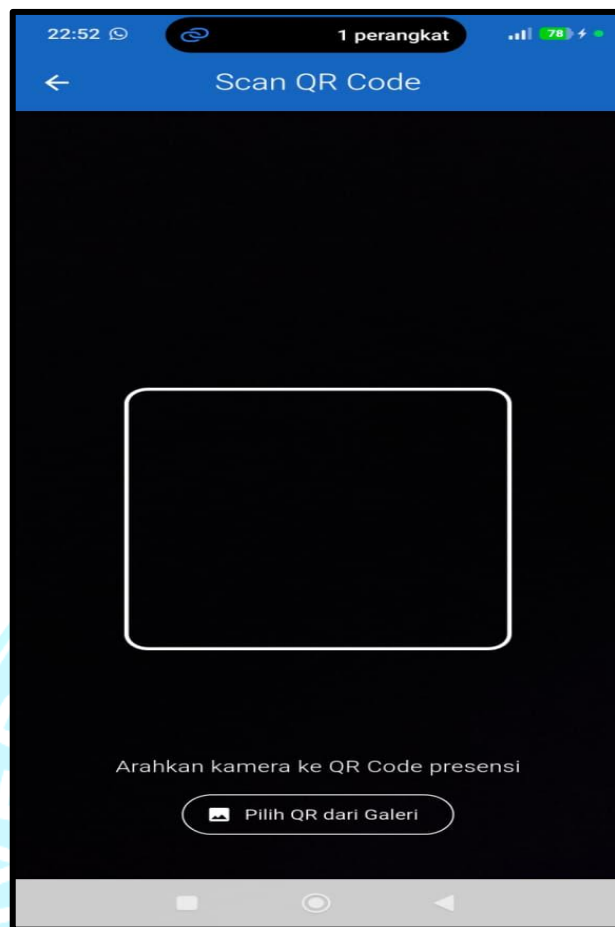
Sumber: data diolah oleh penulis, 2026

Gambar 5. 9 Halaman *Dashboard* Karyawan

Halaman *dashboard* karyawan merupakan halaman utama yang ditampilkan setelah pengguna berhasil *login* ke dalam aplikasi “digitalisasi sistem informasi karyawan dan presensi menggunakan *qr code* berbasis *android*”. Tampilan halaman ini dirancang sederhana dan *modern* untuk memudahkan pengguna dalam mengakses fitur utama aplikasi.

Pada bagian atas halaman terdapat informasi pengguna berupa nama karyawan, jabatan, dan divisi kerja yang ditampilkan dalam *card* berwarna biru sebagai identitas pengguna yang sedang *login* ke dalam sistem. selanjutnya terdapat menu utama yaitu presensi dan riwayat presensi. Menu presensi digunakan untuk melakukan *scan qr code* dalam proses presensi, menu riwayat presensi digunakan untuk melihat data dan riwayat kehadiran karyawan, sedangkan menu panduan presensi digunakan sebagai panduan tata cara dalam melakukan presensi.

5.3.5 Tampilan Halaman *Scan QR Code*



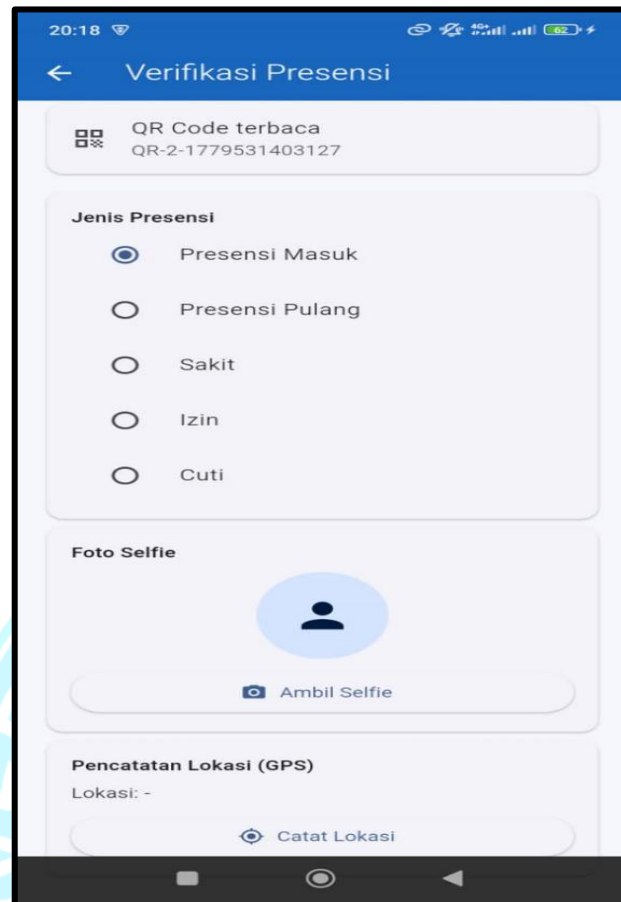
Sumber: data diolah oleh penulis, 2026

Gambar 5. 10 Halaman *Scan QR Code*

Halaman *scan qr code* merupakan halaman yang digunakan karyawan untuk melakukan proses presensi dengan cara memindai *qr code* yang telah dibuat oleh *hrd*. Tampilan halaman ini dirancang sederhana agar pengguna dapat langsung mengarahkan kamera ke *qr code* presensi. Pada bagian tengah halaman terdapat area kotak *scan* yang berfungsi sebagai panduan posisi kamera saat membaca *qr code*. pengguna cukup mengarahkan *qr code* ke dalam area tersebut agar sistem dapat melakukan proses pembacaan data.

Di bagian bawah terdapat teks instruksi “arahkan kamera ke *qr code* presensi” sebagai panduan bagi pengguna. selain itu, tersedia tombol pilih *qr* dari galeri yang dapat digunakan apabila pengguna ingin mengambil gambar *qr code* dari penyimpanan perangkat. tampilan halaman menggunakan latar gelap agar area *scan* lebih fokus dan mudah digunakan.

5.3.6 Tampilan Halaman Verifikasi Presensi (Masuk dan Pulang)



Sumber: data diolah oleh penulis, 2026

Gambar 5. 11 Halaman Verifikasi Presensi (Masuk dan Pulang)

Halaman verifikasi presensi merupakan halaman yang digunakan untuk melakukan proses verifikasi kehadiran setelah *qr code* berhasil dipindai. pada halaman ini, pengguna dapat memilih jenis presensi yang akan dilakukan, seperti presensi masuk, presensi pulang, sakit, izin, dan cuti. Apabila pengguna memilih presensi masuk atau pulang, maka sistem akan menampilkan fitur foto *selfie* dan pencatatan lokasi (*gps*) sebagai proses verifikasi tambahan.

Pengguna diwajibkan mengambil foto *selfie* dan mencatat lokasi *gps* sebelum presensi dapat disimpan ke dalam sistem. Hal ini bertujuan untuk meningkatkan keamanan dan keakuratan data presensi. Pada bagian atas halaman juga ditampilkan informasi *qr code* yang berhasil terbaca oleh sistem. Selain itu, tersedia tombol ambil *selfie* untuk membuka kamera perangkat dan tombol catat lokasi untuk mengambil lokasi *gps* pengguna secara otomatis.

5.3.7 Tampilan Halaman Verifikasi Presensi (Sakit dan Izin)

Sumber: data diolah oleh penulis, 2026

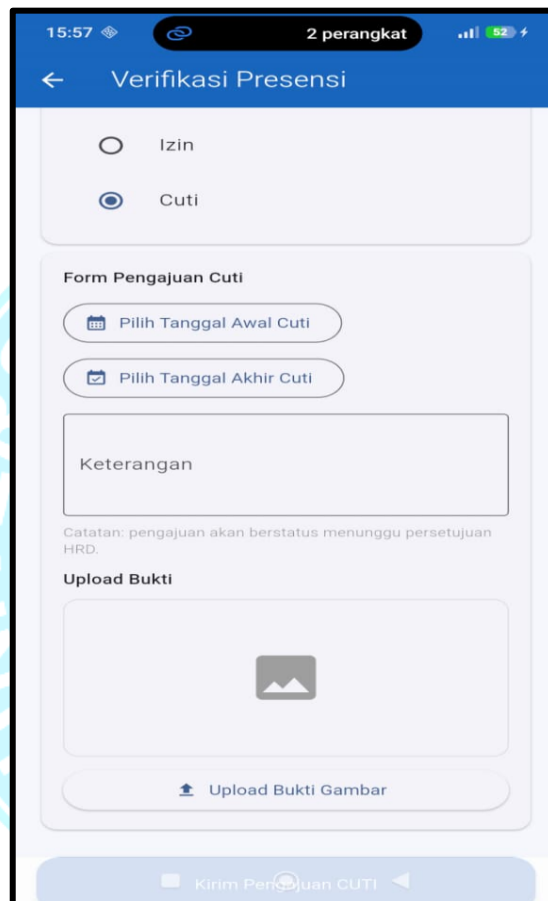
Gambar 5. 12 Halaman Verifikasi Presensi (Sakit dan Izin)

Halaman verifikasi presensi (Sakit dan Izin) merupakan halaman yang digunakan untuk melakukan proses verifikasi kehadiran setelah *qr code* berhasil dipindai. Pada halaman ini, pengguna dapat memilih jenis presensi yang akan dilakukan, seperti presensi masuk, presensi pulang, sakit, izin, dan cuti. Informasi *qr code* yang berhasil terbaca juga ditampilkan pada bagian atas halaman sebagai bukti bahwa proses pemindaian telah berhasil dilakukan oleh sistem. Apabila pengguna memilih presensi sakit atau izin, maka sistem akan menampilkan *form* pengajuan yang berisi tanggal pengajuan, keterangan, serta fitur *upload* gambar sebagai bukti pendukung.

Pengguna diwajibkan mengunggah bukti gambar, seperti surat sakit atau dokumen izin sebelum pengajuan dapat dikirim ke dalam sistem. Hal ini bertujuan untuk memastikan bahwa pengajuan yang dilakukan memiliki bukti pendukung yang *valid* dan dapat dipertanggung jawabkan.

Selain itu, tersedia tombol *upload* bukti untuk memilih gambar dari galeri perangkat pengguna. Setelah seluruh data pengajuan diisi dengan lengkap, pengguna dapat menekan tombol kirim pengajuan untuk menyimpan data ke dalam sistem dan menunggu proses persetujuan dari pihak *hrd* maupun pimpinan.

5.3.8 Tampilan Halaman Verifikasi Presensi (Cuti)



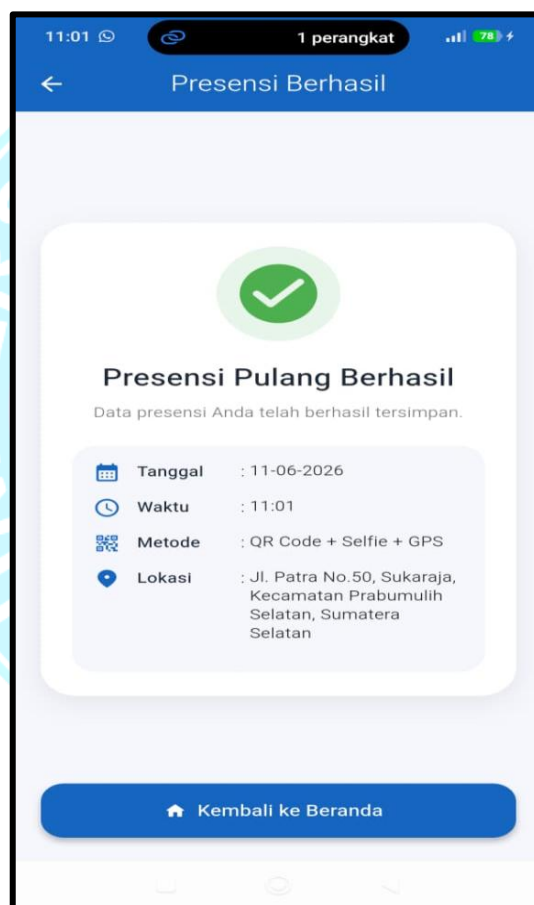
Sumber: data diolah oleh penulis, 2026

Gambar 5. 13 Halaman Verifikasi Presensi (Cuti)

Halaman verifikasi presensi merupakan halaman yang digunakan untuk melakukan proses verifikasi kehadiran setelah *qr code* berhasil dipindai. pada halaman ini, pengguna dapat memilih jenis presensi yang akan dilakukan, seperti presensi masuk, presensi pulang, sakit, cuti. Apabila pengguna memilih presensi cuti, maka sistem akan menampilkan *form* pengajuan yang berisi tanggal awal cuti sampai dengan tanggal akhir cuti, keterangan, serta fitur *upload* gambar sebagai bukti pendukung.

Pengguna diwajibkan mengunggah bukti gambar, seperti surat atau dokumen izin sebelum pengajuan dapat dikirim ke dalam sistem. Hal ini bertujuan untuk memastikan bahwa pengajuan yang dilakukan memiliki bukti pendukung yang *valid* dan dapat dipertanggung jawabkan. Setelah seluruh data pengajuan diisi dengan lengkap, pengguna dapat menekan tombol kirim pengajuan untuk menyimpan data ke dalam sistem dan menunggu proses persetujuan dari pihak *hrd* maupun pimpinan.

5.3.9 Tampilan Halaman Presensi Berhasil (Masuk dan Pulang)



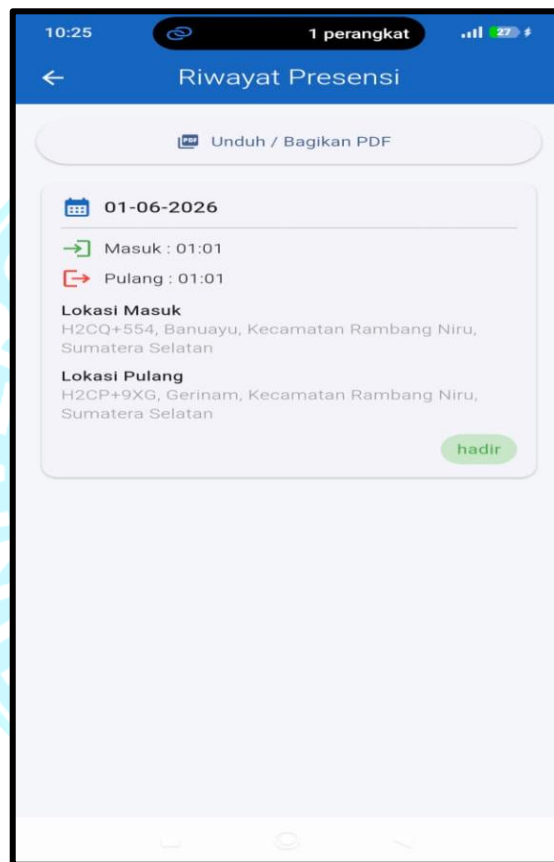
Sumber: data diolah oleh penulis, 2026

Gambar 5. 14 Halaman Presensi Berhasil (Masuk dan Pulang)

Pada gambar tampilan presensi berhasil di atas, dijelaskan bahwa setelah pengguna menyelesaikan seluruh proses presensi yang meliputi *scan qr code*, pengambilan foto *selfie*, dan pencatatan lokasi *gps*, sistem akan menampilkan halaman konfirmasi bahwa presensi telah berhasil dilakukan. Pada halaman ini ditampilkan informasi ringkasan presensi seperti tanggal, waktu, dan metode

presensi yang digunakan. Dan pencatatan data lokasi saat melakukan presensi. Selain itu, terdapat informasi bahwa data presensi telah berhasil disimpan ke dalam sistem dan dapat dilihat kembali pada riwayat presensi. Pengguna juga disediakan tombol “kembali ke beranda” untuk kembali ke halaman utama aplikasi. Halaman ini berfungsi sebagai indikator bahwa proses presensi telah selesai dan tersimpan dengan baik di dalam sistem.

5.3.10 Tampilan Halaman Riwayat Presensi



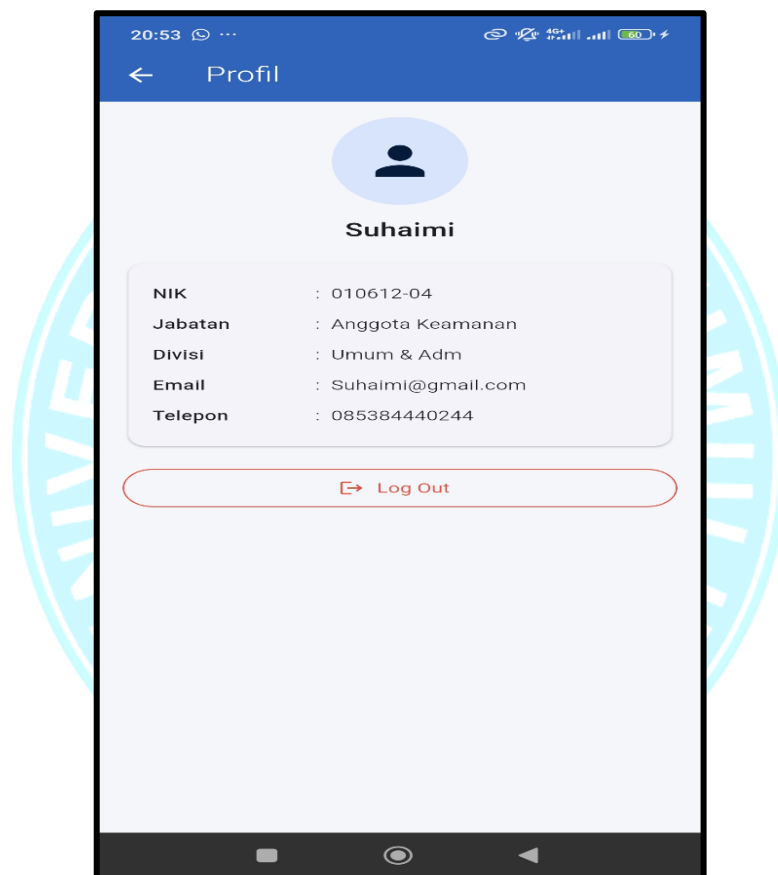
Sumber: data diolah oleh penulis, 2026

Gambar 5. 15 Halaman Riwayat Presensi

Halaman riwayat presensi merupakan halaman yang digunakan untuk menampilkan data dan riwayat kehadiran karyawan yang telah tersimpan di dalam sistem. Pada halaman ini, pengguna dapat melihat informasi presensi berdasarkan tanggal kehadiran secara detail. setiap data presensi ditampilkan dalam bentuk *card* yang berisi informasi tanggal, jam masuk, jam pulang, lokasi masuk, lokasi pulang, serta status kehadiran.

Status kehadiran dapat berupa hadir, sakit, izin, cuti, maupun *alpha* sesuai data yang tersimpan pada *database firebase firestore*. Halaman ini membantu pengguna dalam memantau riwayat kehadiran dan memastikan data presensi yang telah dilakukan tercatat dengan baik di dalam sistem. Dan juga tersedia fitur unduh/bagikan *pdf* berfungsi sebagai bukti laporan kehadiran karyawan tersebut apabila diperlukan.

5.3.11 Tampilan Halaman Profil dan *Log Out* Karyawan



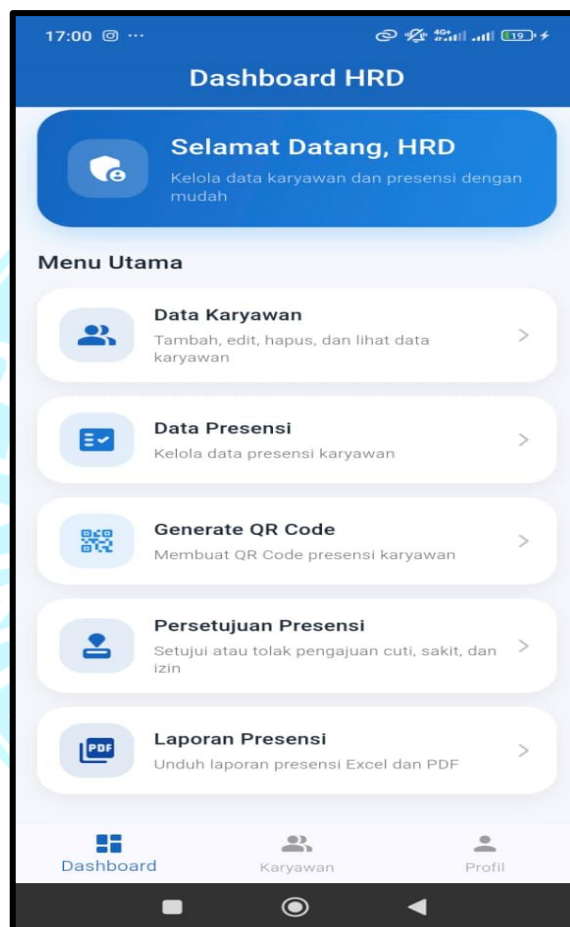
Sumber: data diolah oleh penulis, 2026

Gambar 5. 16 Halaman Profil dan *Log Out* Karyawan

Halaman profil merupakan halaman yang digunakan untuk menampilkan informasi data pribadi pengguna yang sedang *login* ke dalam aplikasi. Pada halaman ini, pengguna dapat melihat identitas dan data karyawan yang tersimpan pada *database firebase firestore*. Pada bagian atas halaman ditampilkan foto profil *default* dan nama pengguna sebagai identitas akun yang sedang digunakan.

Selanjutnya terdapat informasi data karyawan yang ditampilkan dalam bentuk *card*, seperti nik, jabatan, divisi, *email*, dan telepon. Informasi tersebut digunakan untuk memudahkan pengguna dalam melihat data diri yang telah terdaftar di dalam sistem. Pada bagian bawah halaman terdapat tombol *log out* yang digunakan pengguna untuk keluar dari aplikasi dan kembali ke halaman *login*.

5.3.12 Tampilan Halaman Dashboard *HRD*



Sumber: data diolah oleh penulis, 2026

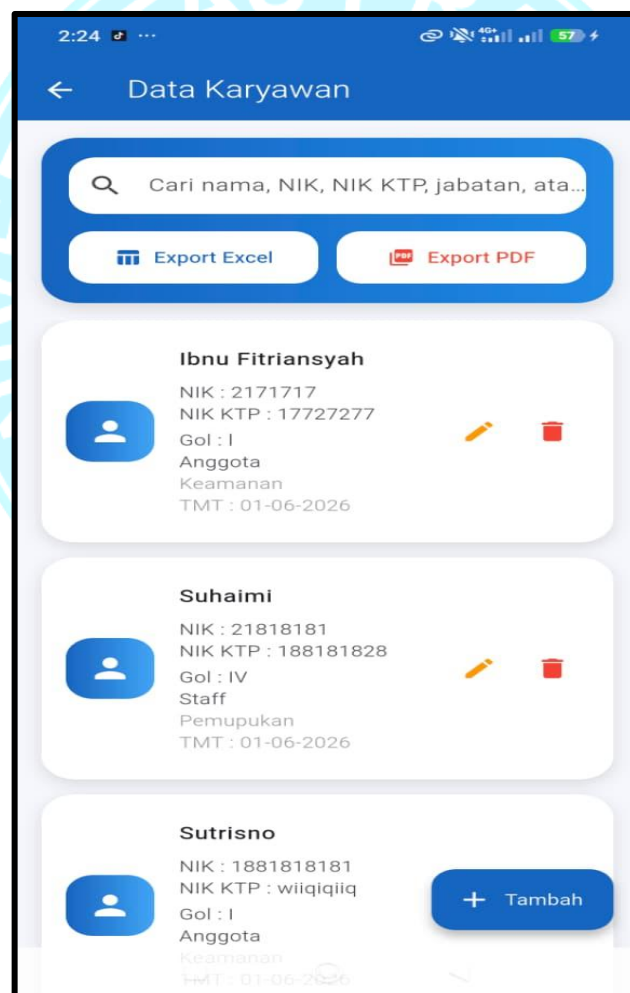
Gambar 5. 17 Halaman *Dashboard HRD*

Halaman *dashboard hrd* merupakan halaman utama yang ditampilkan setelah pengguna dengan *role hrd* berhasil *login* ke dalam aplikasi “digitalisasi sistem informasi karyawan dan presensi menggunakan *qr code* berbasis *android*”. Halaman ini digunakan untuk mengelola data karyawan, presensi, dan berbagai proses administrasi pada sistem. Pada bagian atas halaman terdapat *card* informasi

pengguna *hrd* sebagai identitas dan penanda hak akses pengelolaan data pada aplikasi.

Selanjutnya terdapat beberapa menu utama, yaitu data karyawan, data presensi, *generate qr code*, persetujuan presensi, dan laporan presensi. menu-menu tersebut digunakan untuk mengelola data karyawan, memantau presensi, membuat *qr code*, memproses pengajuan sakit/izin/cuti, serta mengunduh laporan presensi dalam bentuk *pdf* dan *excel*. Pada bagian bawah halaman terdapat *navigation bar* yang terdiri dari menu *dashboard*, karyawan, dan profil untuk memudahkan perpindahan halaman.

5.3.13 Tampilan Halaman Data Karyawan



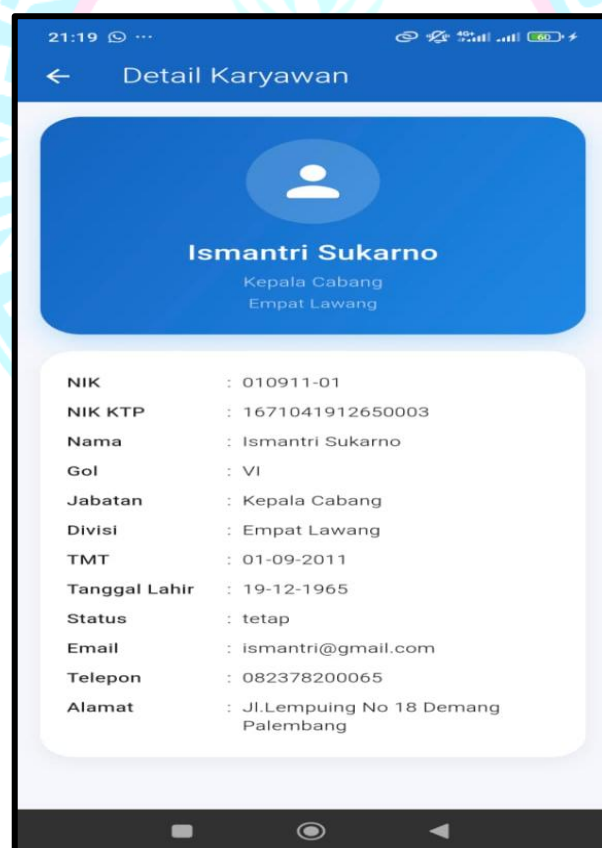
Sumber: data diolah oleh penulis, 2026

Gambar 5. 18 Halaman Data Karyawan

Halaman data karyawan merupakan halaman yang digunakan oleh *hrd* untuk mengelola data seluruh karyawan yang terdaftar di dalam sistem. Pada halaman ini, *hrd* dapat melihat, mencari, menambah, mengubah, dan menghapus data karyawan sesuai kebutuhan perusahaan. Pada bagian atas halaman terdapat fitur pencarian yang digunakan untuk mencari data berdasarkan nama, nik, nik ktp, jabatan, atau data lainnya. Selain itu, tersedia tombol *export excel* dan *export pdf* yang digunakan untuk mengunduh data karyawan dalam bentuk laporan *excel* maupun *pdf*.

Data karyawan ditampilkan dalam bentuk *card* yang berisi informasi seperti nama, nik, nik ktp, golongan, jabatan, divisi, dan tmt. Pada setiap *card* juga tersedia tombol *edit* dan hapus untuk memudahkan *hrd* dalam melakukan pengelolaan data. Di bagian bawah halaman terdapat tombol tambah yang digunakan untuk menambahkan data karyawan baru ke dalam sistem.

5.3.14 Tampilan Halaman Detail Karyawan



Sumber: data diolah oleh penulis, 2026

Gambar 5. 19 Halaman Detail Karyawan

Halaman detail karyawan merupakan halaman yang digunakan untuk menampilkan informasi lengkap data karyawan yang tersimpan di dalam sistem. Halaman ini membantu *hrd* maupun pimpinan dalam melihat detail identitas dan data kepegawaian setiap karyawan secara lebih lengkap. Pada bagian atas halaman terdapat *card* informasi karyawan yang menampilkan foto profil *default*, nama karyawan, jabatan, dan divisi kerja.

Tampilan ini berfungsi sebagai identitas utama karyawan yang sedang dilihat datanya. Selanjutnya ditampilkan informasi detail karyawan dalam bentuk *card*, seperti nik, nik ktp, nama, golongan, jabatan, divisi, tmt, tanggal lahir, status, *email*, telepon, dan alamat. Informasi tersebut digunakan untuk memudahkan proses pengelolaan dan pencarian data karyawan di dalam perusahaan.

5.3.15 Tampilan Halaman Tambah Karyawan

Sumber: data diolah oleh penulis, 2026

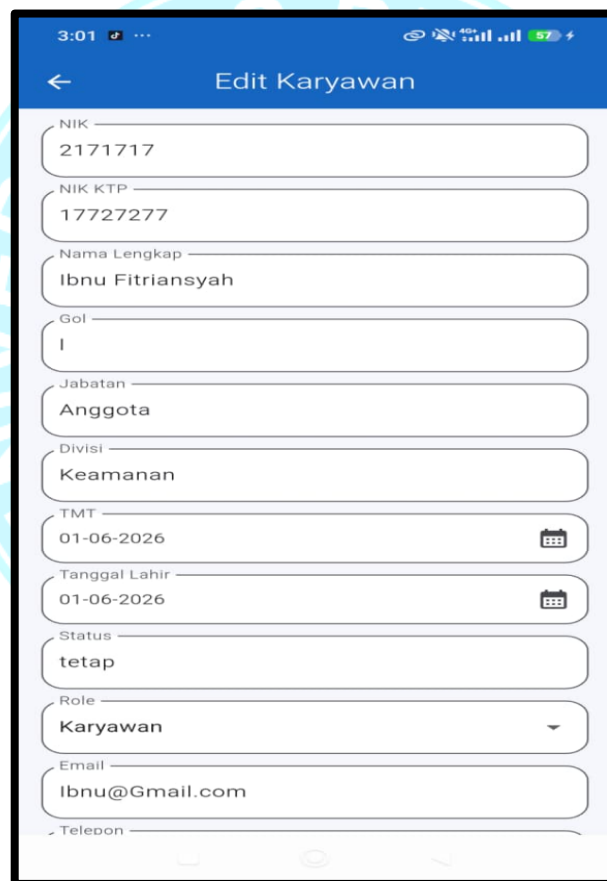
Gambar 5. 20 Halaman Tambah Karyawan

Halaman tambah karyawan merupakan halaman yang digunakan oleh *hrd* untuk menambahkan data karyawan baru ke dalam sistem. Pada halaman ini, *hrd*

dapat mengisi berbagai informasi identitas dan data kepegawaian yang nantinya akan disimpan pada *database firebase firestore*.

Form input pada halaman ini terdiri dari beberapa *field*, seperti *nik*, *nik ktp*, *nama lengkap*, *golongan*, *jabatan*, *divisi*, *tmt*, *tanggal lahir*, *status*, *role*, *email*, *password* awal, dan *telepon*. Selain itu, tersedia fitur pemilihan tanggal menggunakan *date picker* pada bagian *tmt* dan *tanggal lahir* untuk mempermudah proses pengisian data. Halaman ini digunakan untuk memastikan seluruh data karyawan dapat tersimpan secara lengkap dan terorganisir di dalam sistem.

5.3.16 Tampilan Halaman *Edit Karyawan*



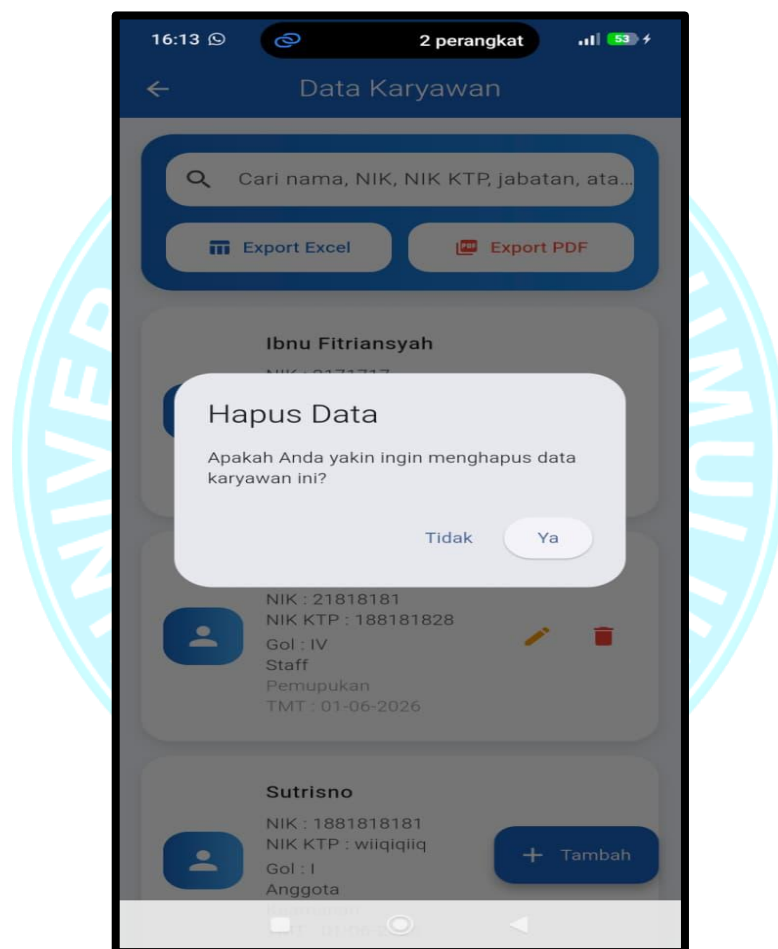
Sumber: data diolah oleh penulis, 2026

Gambar 5. 21 Halaman *Edit Karyawan*

Halaman *edit* karyawan merupakan halaman yang digunakan oleh *hrd* untuk mengubah dan memperbarui data karyawan yang telah tersimpan di dalam sistem. Pada halaman ini, *hrd* dapat melakukan perubahan data apabila terdapat informasi karyawan yang berubah atau perlu diperbaiki.

Form pada halaman ini menampilkan data karyawan yang sudah tersimpan sebelumnya, seperti nik, nik ktp, nama lengkap, golongan, jabatan, divisi, tmt, tanggal lahir, status, *email*, telepon, dan alamat. Selain itu, tersedia fitur *date picker* pada bagian tmt dan tanggal lahir untuk mempermudah proses perubahan tanggal. Halaman ini membantu *hrd* dalam menjaga data karyawan agar tetap lengkap, akurat, dan sesuai dengan kondisi terbaru di perusahaan.

5.3.17 Tampilan Halaman Hapus Data Karyawan



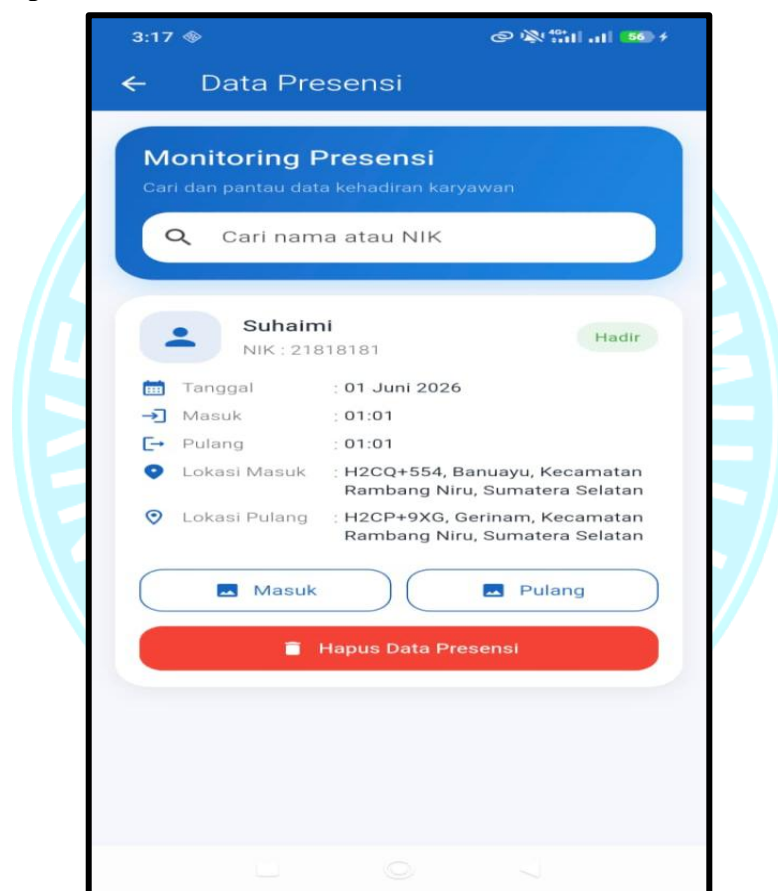
Sumber: data diolah oleh penulis, 2026

Gambar 5. 22 Halaman Hapus Data Karyawan

Halaman hapus data karyawan merupakan halaman yang digunakan oleh *hrd* untuk menghapus data karyawan yang telah tersimpan di dalam sistem. Pada halaman ini, *hrd* dapat menghapus data karyawan apabila data tersebut sudah tidak digunakan lagi, seperti karyawan yang telah *resign* atau tidak lagi bekerja di perusahaan.

Halaman ini menampilkan informasi data karyawan yang akan dihapus, seperti nik, nik ktp, nama lengkap, golongan, jabatan, divisi, status, *email*, telepon, dan alamat. Sebelum proses penghapusan dilakukan, sistem akan menampilkan konfirmasi penghapusan untuk memastikan bahwa data yang dipilih benar-benar akan dihapus dari sistem. Halaman ini membantu *hrd* dalam mengelola data karyawan agar data yang tersimpan tetap rapi, *valid*, dan sesuai dengan kondisi terbaru di perusahaan.

5.3.18 Tampilan Halaman Data Presensi



Sumber: data diolah oleh penulis, 2026

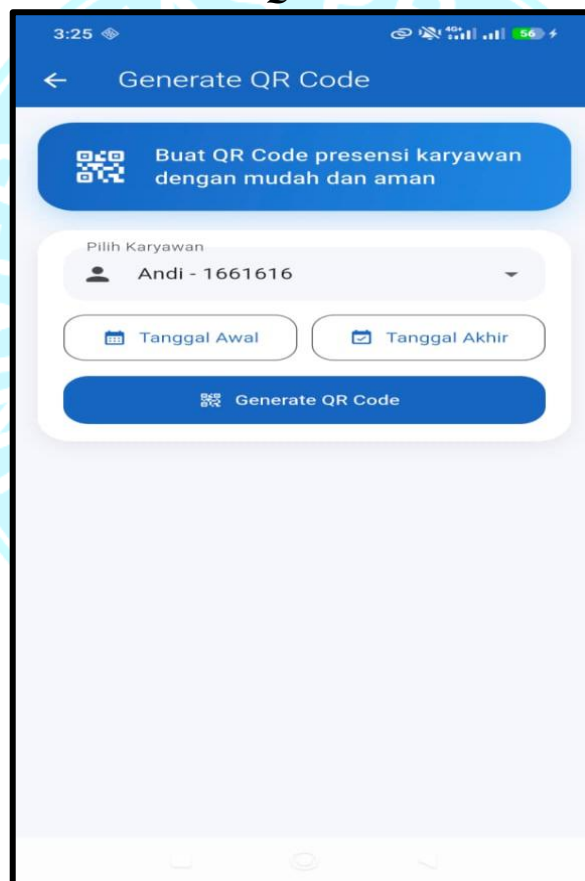
Gambar 5. 23 Halaman Data Presensi

Halaman data presensi merupakan halaman yang digunakan oleh *hrd* dan pimpinan untuk memantau data kehadiran karyawan yang tersimpan di dalam sistem. Pada halaman ini, pengguna dapat melihat informasi presensi karyawan secara detail berdasarkan data yang telah tercatat pada *database firebase*

firestore. Pada bagian atas halaman terdapat fitur pencarian yang digunakan untuk mencari data presensi berdasarkan nama atau nik karyawan.

Selanjutnya data presensi ditampilkan dalam bentuk *card* yang berisi informasi seperti nama, nik, tanggal, jam masuk, jam pulang, lokasi masuk, lokasi pulang, dan status kehadiran. Selain itu, tersedia tombol untuk melihat foto *selfie* presensi masuk dan pulang sebagai bukti verifikasi kehadiran. Khusus pada *role hrd*, tersedia fitur hapus data presensi yang digunakan untuk menghapus data presensi apabila diperlukan, sedangkan pimpinan hanya dapat melihat dan memantau data presensi tanpa melakukan perubahan data.

5.3.19 Tampilan Halaman *Generate QR Code*



Sumber: data diolah oleh penulis, 2026

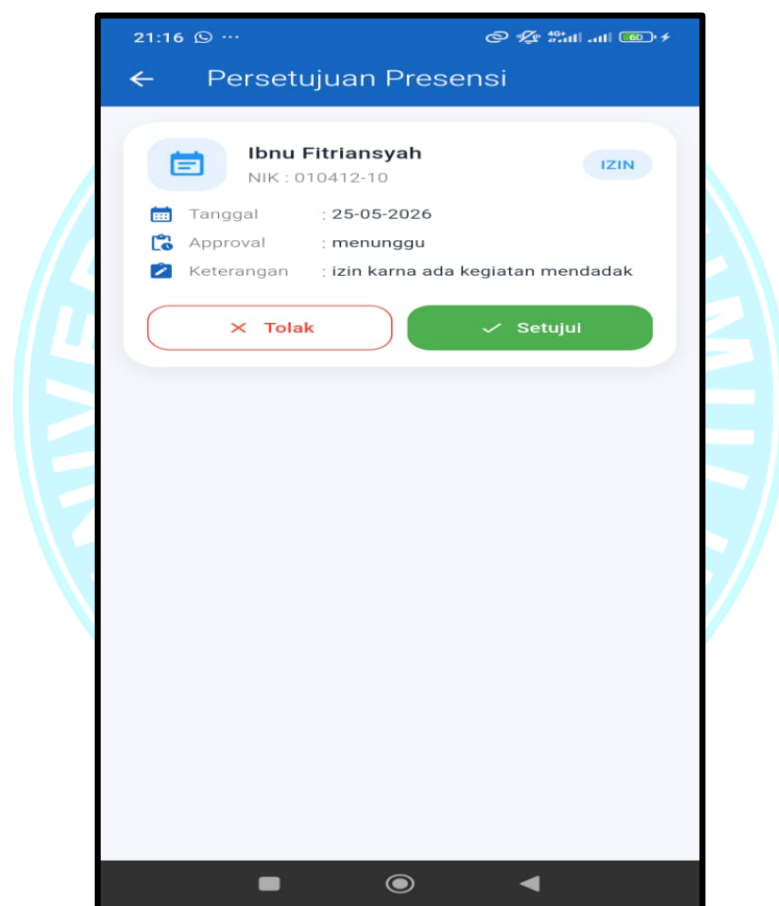
Gambar 5. 24 Halaman *Generate QR Code*

Halaman *generate qr code* merupakan halaman yang digunakan oleh *hrd* untuk membuat *qr code* presensi karyawan. *qr code* ini digunakan sebagai media utama dalam proses presensi masuk maupun pulang pada aplikasi. Pada bagian atas halaman terdapat informasi dan *form* pembuatan *qr code* yang terdiri dari pilihan

karyawan serta tanggal mulai dan tanggal akhir masa berlaku *qr code*. Setelah data dipilih, *hrd* dapat menekan tombol *generate qr code* untuk membuat *qr code* presensi secara otomatis.

Selanjutnya sistem akan menampilkan informasi jumlah *qr code* yang masih aktif beserta hasil *qr code* yang berhasil dibuat. pada bagian tersebut juga ditampilkan kode *qr* dan masa berlaku *qr code* agar dapat digunakan sesuai periode yang telah ditentukan.

5.3.20 Tampilan Halaman Persetujuan Presensi



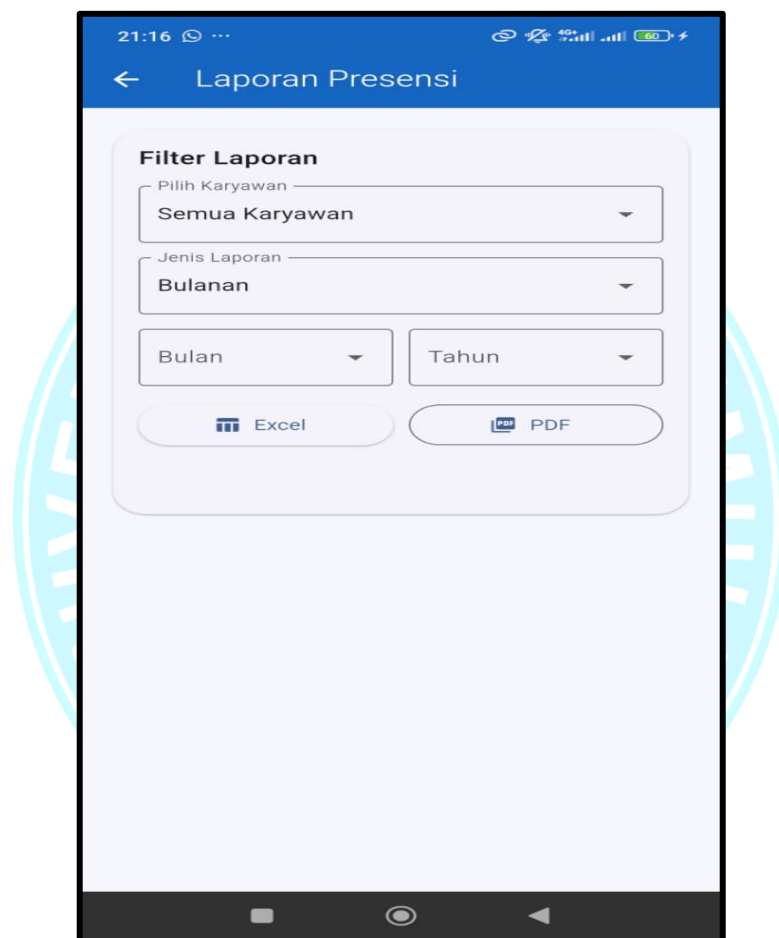
Sumber: data diolah oleh penulis, 2026

Gambar 5. 25 Halaman Persetujuan Presensi

Halaman persetujuan presensi merupakan halaman yang digunakan oleh *hrd* untuk memproses pengajuan presensi karyawan, seperti izin, sakit, dan cuti. Data pengajuan yang ditampilkan pada halaman ini berasal dari *collection* pengajuan pada *database firebase firestore*. Pada halaman ini ditampilkan informasi pengajuan karyawan, seperti nama, nik, tanggal, status *approval*, dan keterangan pengajuan.

Selain itu, pada bagian kanan atas *card* ditampilkan jenis pengajuan, misalnya izin, sakit, atau cuti. *Hrd* dapat melakukan proses persetujuan melalui tombol setuju dan tolak. Apabila pengajuan disetujui, maka data akan masuk ke *collection* presensi dengan status sesuai pengajuan. Sedangkan apabila pengajuan ditolak, maka sistem akan mencatat status kehadiran sebagai *alpha*.

5.3.21 Tampilan Halaman Laporan Presensi



Sumber: data diolah oleh penulis, 2026

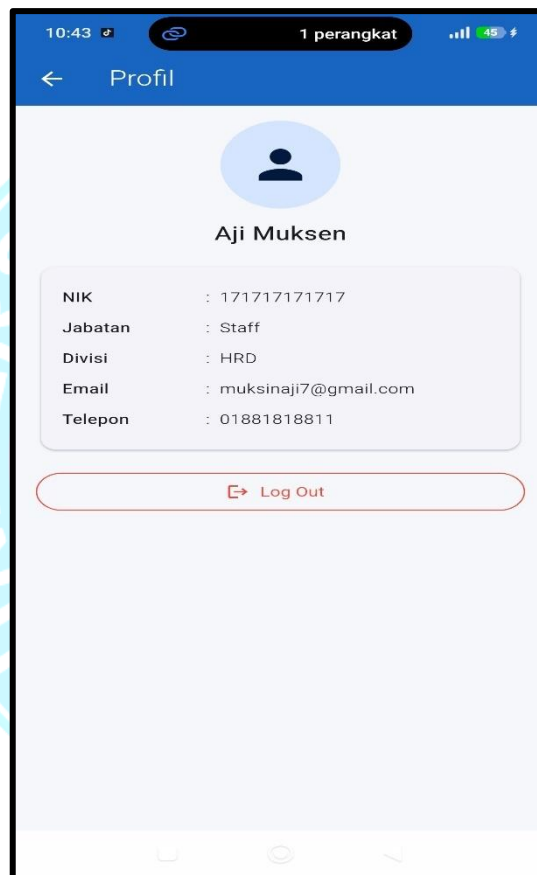
Gambar 5. 26 Halaman Laporan Presensi

Halaman laporan presensi merupakan halaman yang digunakan oleh *hrd* dan pimpinan untuk membuat serta mengunduh laporan data kehadiran karyawan. Pada halaman ini, pengguna dapat melakukan filter laporan berdasarkan karyawan, jenis laporan, bulan, dan tahun sesuai kebutuhan. Pada bagian filter laporan tersedia pilihan semua karyawan atau karyawan tertentu untuk menampilkan data presensi yang diinginkan. Selain itu, pengguna juga dapat memilih jenis laporan, seperti

laporan bulanan maupun tahunan. Sistem kemudian akan menampilkan data sesuai filter yang dipilih.

Di bagian bawah halaman terdapat tombol *excel* dan *pdf* yang digunakan untuk mengunduh laporan presensi dalam *format excel* maupun *pdf*. Fitur ini memudahkan *hrd* dan pimpinan dalam melakukan rekap, penyimpanan, dan pencetakan laporan kehadiran karyawan.

5.3.22 Tampilan Halaman Profil dan Log Out HRD



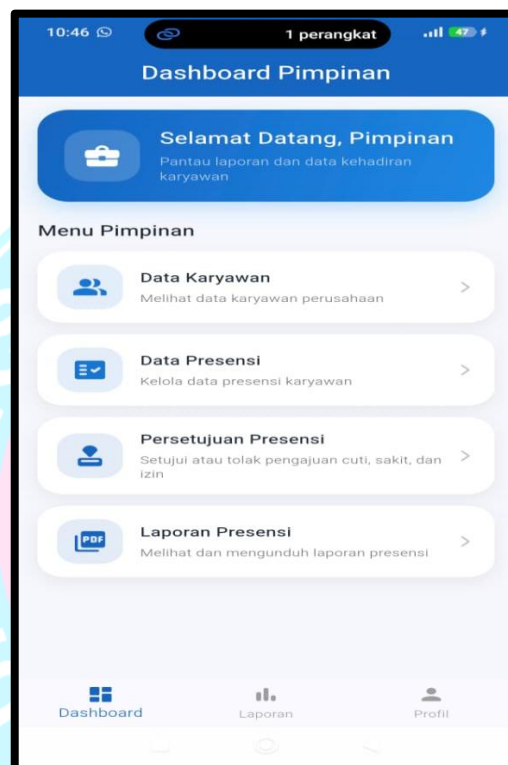
Sumber: data diolah oleh penulis, 2026

Gambar 5. 27 Halaman Profil dan Log Out HRD

Halaman profil merupakan halaman yang digunakan untuk menampilkan informasi data pribadi pengguna yang sedang *login* ke dalam aplikasi. Pada halaman ini, pengguna dapat melihat identitas dan data *hrd* yang tersimpan pada *database firebase firestore*. Pada bagian atas halaman ditampilkan foto profil *default* dan nama pengguna sebagai identitas akun yang sedang digunakan.

Selanjutnya terdapat informasi data karyawan yang ditampilkan dalam bentuk *card*, seperti nik, jabatan, divisi, *email*, dan telepon. Informasi tersebut digunakan untuk memudahkan pengguna dalam melihat data diri yang telah terdaftar di dalam sistem. Pada bagian bawah halaman terdapat tombol *log out* yang digunakan pengguna untuk keluar dari aplikasi dan kembali ke halaman *login*.

5.3.23 Tampilan Halaman *Dashboard Pimpinan*



Sumber: data diolah oleh penulis, 2026

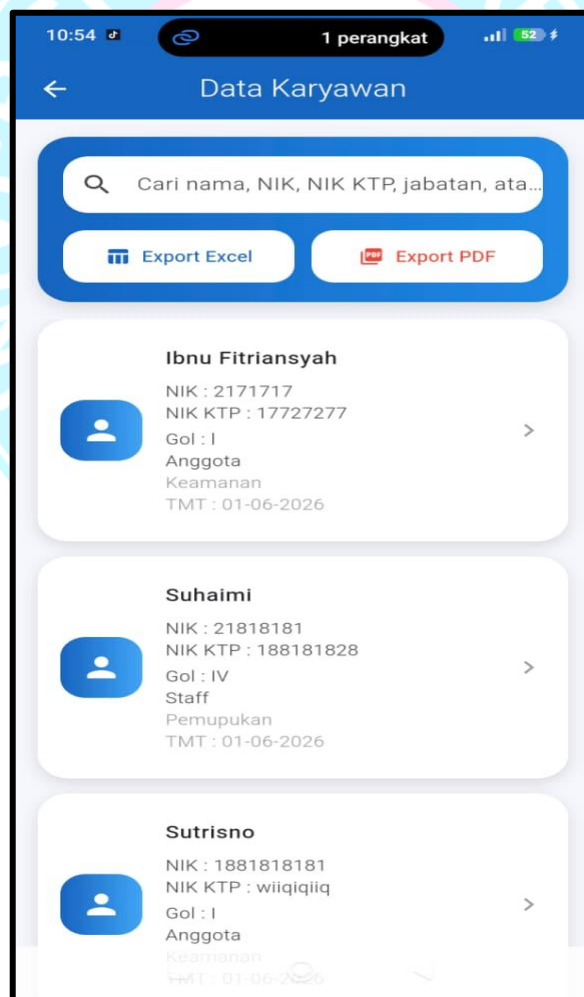
Gambar 5. 28 Halaman *Dashboard Pimpinan*

Halaman *dashboard* pimpinan merupakan halaman utama yang ditampilkan setelah pengguna dengan *role* pimpinan berhasil *login* ke dalam aplikasi “digitalisasi sistem informasi karyawan dan presensi menggunakan *qr code* berbasis *android*”. Halaman ini digunakan untuk memantau data karyawan, data presensi, menolak atau menyetujui persetujuan presensi (cuti) dan laporan kehadiran karyawan pada perusahaan. Pada bagian atas halaman terdapat *card* informasi pengguna dengan tampilan berwarna biru yang menampilkan sambutan untuk pimpinan. tampilan ini berfungsi sebagai identitas pengguna sekaligus

memberikan informasi bahwa pimpinan memiliki hak akses untuk melihat data dan laporan pada sistem.

Selanjutnya terdapat beberapa menu utama, yaitu menu data karyawan, data presensi, persetujuan presensi, laporan presensi, menu data karyawan digunakan untuk melihat data karyawan perusahaan, menu data presensi digunakan untuk melihat data kehadiran karyawan, menu persetujuan presensi untuk menolak atau menyetujui persetujuan presensi (cuti), menu laporan presensi digunakan untuk melihat dan mengunduh laporan presensi. Pada bagian bawah halaman terdapat *navigation bar* yang terdiri dari menu *dashboard*, laporan, dan profil untuk memudahkan perpindahan halaman pada aplikasi.

5.3.24 Tampilan Halaman Data Karyawan

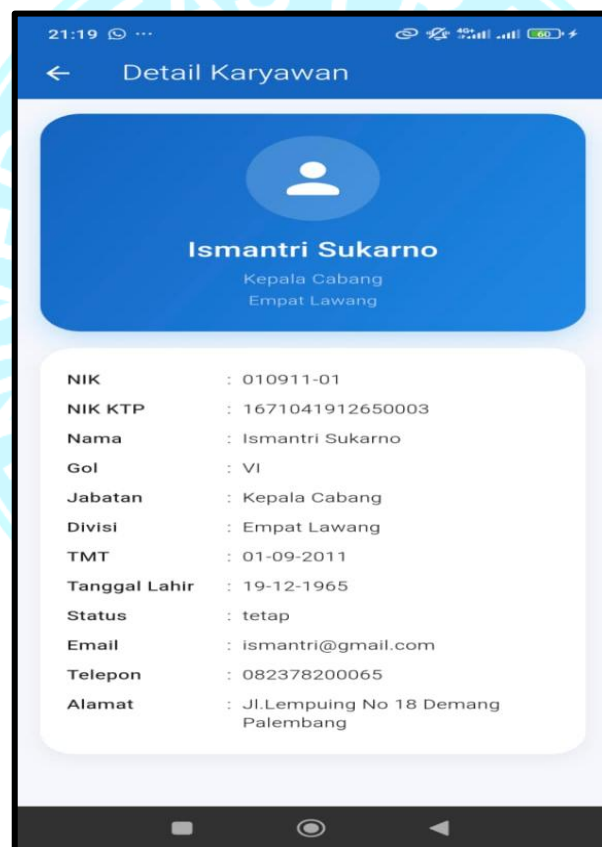


Sumber: data diolah oleh penulis, 2026

Gambar 5. 29 Halaman Data Karyawan

Halaman data karyawan merupakan halaman yang digunakan oleh pimpinan untuk melihat data seluruh karyawan yang terdaftar di dalam sistem. Pada halaman ini, *pimpinan* dapat melihat, mencari, data karyawan sesuai kebutuhan perusahaan. Pada bagian atas halaman terdapat fitur pencarian yang digunakan untuk mencari data berdasarkan nama, nik, nik ktp, jabatan, atau data lainnya. Selain itu, tersedia tombol *export excel* dan *export pdf* yang digunakan untuk mengunduh data karyawan dalam bentuk laporan *excel* maupun *pdf*. Data karyawan ditampilkan dalam bentuk *card* yang berisi informasi seperti nama, nik, nik ktp, golongan, jabatan, divisi, dan tmt.

5.3.25 Tampilan Halaman Detail Karyawan



Sumber: data diolah oleh penulis, 2026

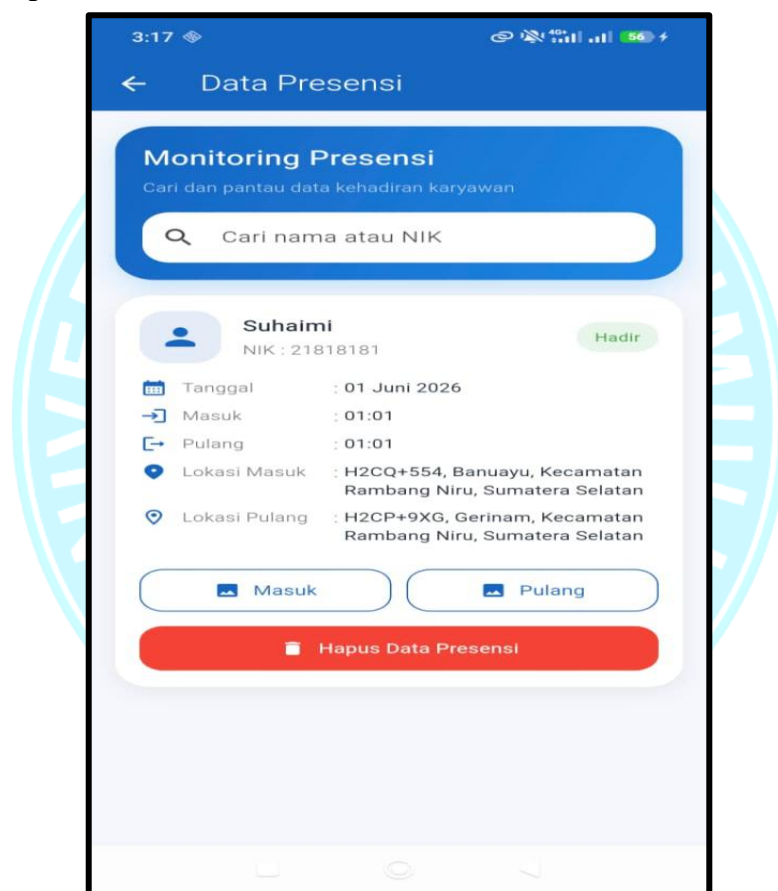
Gambar 5. 30 Halaman Detail Karyawan

Halaman detail karyawan merupakan halaman yang digunakan untuk menampilkan informasi lengkap data karyawan yang tersimpan di dalam sistem. Halaman ini membantu *hrd* maupun pimpinan dalam melihat detail identitas dan data kepegawaian setiap karyawan secara lebih lengkap. Pada bagian atas halaman

terdapat *card* informasi karyawan yang menampilkan foto profil *default*, nama karyawan, jabatan, dan divisi kerja.

Tampilan ini berfungsi sebagai identitas utama karyawan yang sedang dilihat datanya. Selanjutnya ditampilkan informasi detail karyawan dalam bentuk *card*, seperti nik, nik ktp, nama, golongan, jabatan, divisi, tmt, tanggal lahir, status, *email*, telepon, dan alamat. Informasi tersebut digunakan untuk memudahkan proses pengelolaan dan pencarian data karyawan di dalam perusahaan.

5.3.26 Tampilan Halaman Data Presensi



Sumber: data diolah oleh penulis, 2026

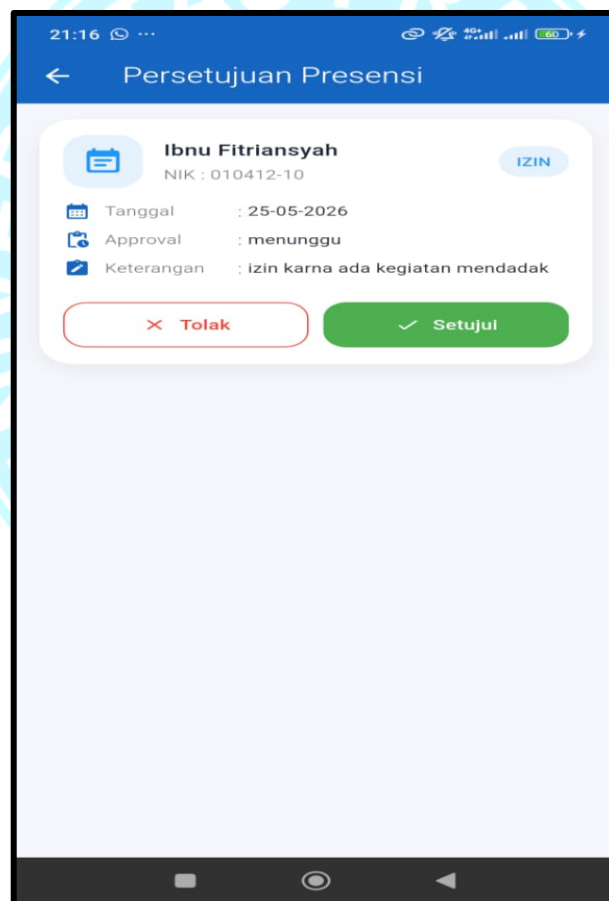
Gambar 5. 31 Halaman Data Presensi

Halaman data presensi merupakan halaman yang digunakan oleh *hrd* dan pimpinan untuk memantau data kehadiran karyawan yang tersimpan di dalam sistem. Pada halaman ini, pengguna dapat melihat informasi presensi karyawan secara detail berdasarkan data yang telah tercatat pada *database firebase*

firestore. Pada bagian atas halaman terdapat fitur pencarian yang digunakan untuk mencari data presensi berdasarkan nama atau nik karyawan.

Selanjutnya data presensi ditampilkan dalam bentuk *card* yang berisi informasi seperti nama, nik, tanggal, jam masuk, jam pulang, lokasi masuk, lokasi pulang, dan status kehadiran. Selain itu, tersedia tombol untuk melihat foto *selfie* presensi masuk dan pulang sebagai bukti verifikasi kehadiran. Khusus pada *role hrd*, tersedia fitur hapus data presensi yang digunakan untuk menghapus data presensi apabila diperlukan, sedangkan pimpinan hanya dapat melihat dan memantau data presensi tanpa melakukan perubahan data.

5.3.27 Tampilan Halaman Persetujuan Presensi



Sumber: data diolah oleh penulis, 2026

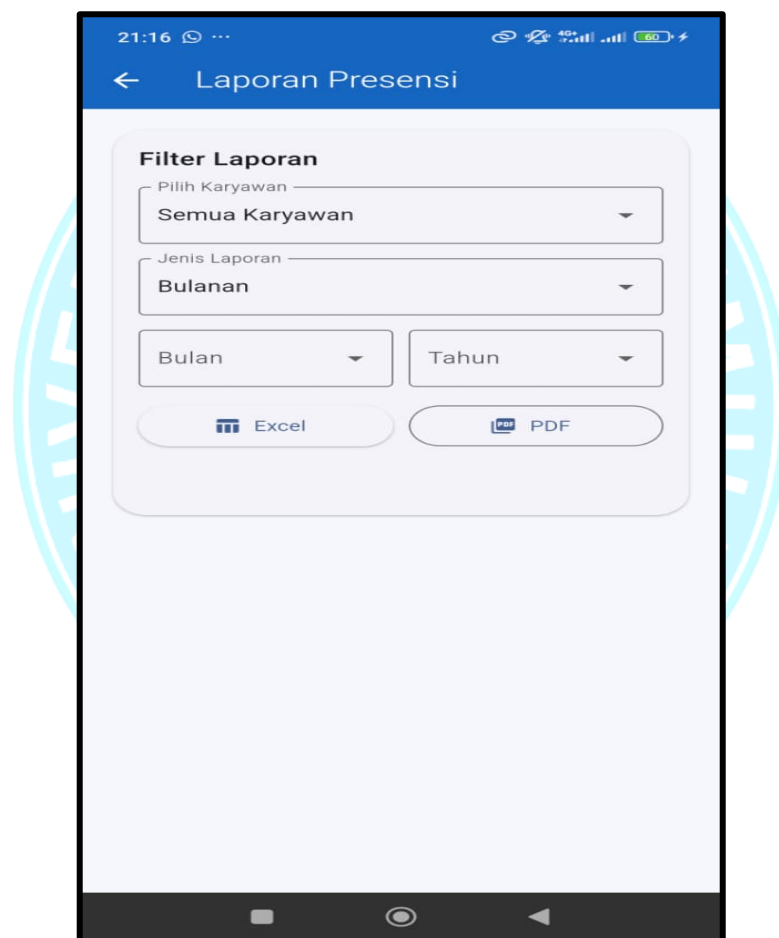
Gambar 5. 32 Halaman Persetujuan Presensi

Halaman persetujuan presensi merupakan halaman yang digunakan oleh pimpinan untuk memproses pengajuan presensi karyawan terkhusus cuti. Data pengajuan yang ditampilkan pada halaman ini berasal dari *collection* pengajuan

pada *database firebase firestore*. Pada halaman ini ditampilkan informasi pengajuan karyawan, seperti nama, nik, tanggal, status *approval*, dan keterangan pengajuan.

Selain itu, pada bagian kanan atas *card* ditampilkan jenis pengajuan cuti. Pimpinan dapat melakukan proses persetujuan melalui tombol setuju dan tolak. Apabila pengajuan disetujui, maka data akan masuk ke *collection* presensi dengan status sesuai pengajuan. Sedangkan apabila pengajuan ditolak, maka sistem tidak akan mengkonfirmasi cuti karyawan tersebut.

5.3.28 Tampilan Halaman Laporan Presensi



Sumber: data diolah oleh penulis, 2026

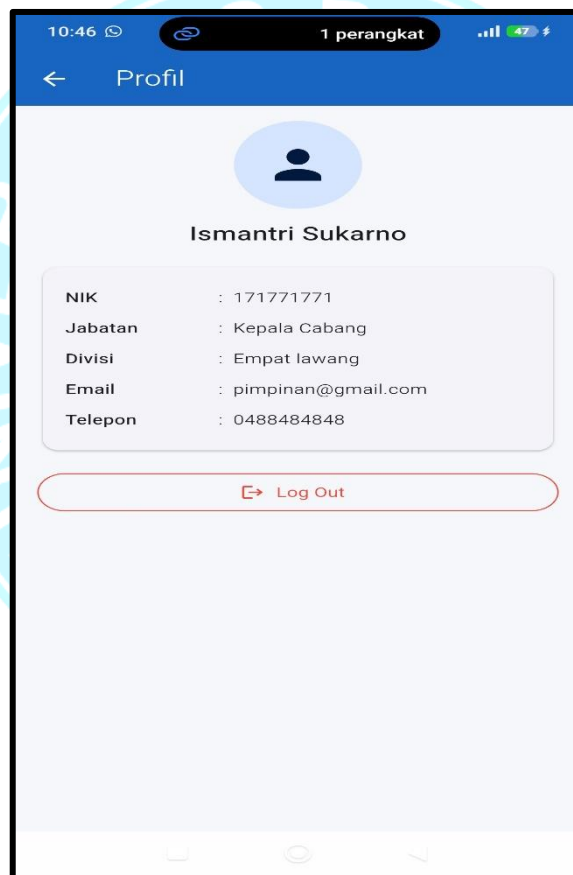
Gambar 5. 33 Halaman Laporan Presensi

Halaman laporan presensi merupakan halaman yang digunakan oleh *hrd* dan pimpinan untuk membuat serta mengunduh laporan data kehadiran karyawan. Pada halaman ini, pengguna dapat melakukan filter laporan berdasarkan karyawan, jenis laporan, bulan, dan tahun sesuai kebutuhan. Pada bagian filter laporan tersedia

pilihan semua karyawan atau karyawan tertentu untuk menampilkan data presensi yang diinginkan. Selain itu, pengguna juga dapat memilih jenis laporan, seperti laporan bulanan maupun tahunan. Sistem kemudian akan menampilkan data sesuai filter yang dipilih.

Di bagian bawah halaman terdapat tombol *excel* dan *pdf* yang digunakan untuk mengunduh laporan presensi dalam *format excel* maupun *pdf*. Fitur ini memudahkan *hrd* dan pimpinan dalam melakukan rekap, penyimpanan, dan pencetakan laporan kehadiran karyawan.

5.3.29 Tampilan Halaman Profil dan *Log Out* Pimpinan



Sumber: data diolah oleh penulis, 2026

Gambar 5. 34 Halaman Profil dan *Log Out* Pimpinan

Halaman profil merupakan halaman yang digunakan untuk menampilkan informasi data pribadi pengguna yang sedang *login* ke dalam aplikasi. Pada halaman ini, pengguna dapat melihat identitas dan data pimpinan yang tersimpan

pada *database firebase firestore*. Pada bagian atas halaman ditampilkan foto profil *default* dan nama pengguna sebagai identitas akun yang sedang digunakan.

Selanjutnya terdapat informasi data karyawan yang ditampilkan dalam bentuk *card*, seperti *nik*, jabatan, divisi, *email*, dan telepon. Informasi tersebut digunakan untuk memudahkan pengguna dalam melihat data diri yang telah terdaftar di dalam sistem. Pada bagian bawah halaman terdapat tombol *log out* yang digunakan pengguna untuk keluar dari aplikasi dan kembali ke halaman *login*.

5.4 Hasil Pengujian Sistem

Pengujian sistem pada penelitian ini dilakukan menggunakan metode *white box testing* untuk menganalisis alur logika program pada aplikasi sistem informasi karyawan dan presensi menggunakan *QR Code* berbasis *android*. Pengujian dilakukan dengan menganalisis *source code* pada beberapa proses utama sistem, seperti proses *login*, lupa *password*, pengelolaan data karyawan, presensi masuk dan pulang, pengajuan presensi, persetujuan pengajuan, hingga proses *logout*. Analisis dimulai dari identifikasi *source code* pada setiap fungsi, kemudian dilakukan identifikasi *statement* untuk menentukan bagian-bagian logika program yang memengaruhi alur eksekusi sistem. *Statement* tersebut selanjutnya dijadikan dasar dalam pembentukan *flowchart* dan *flowgraph* untuk menggambarkan aliran proses program secara terstruktur.

Setelah *flowgraph* terbentuk, dilakukan perhitungan *cyclomatic complexity* untuk mengetahui jumlah jalur independen minimum yang perlu diuji pada setiap fungsi program. Nilai kompleksitas tersebut digunakan sebagai dasar dalam menentukan *independent path* yang merepresentasikan jalur eksekusi berbeda pada sistem. Berdasarkan jalur independen tersebut, kemudian disusun *test case* untuk memastikan seluruh alur logika program berjalan sesuai dengan kondisi yang diharapkan. Tahap akhir pengujian dilakukan menggunakan *statement coverage* untuk mengevaluasi apakah seluruh *statement* pada *source code* yang diuji telah berhasil dieksekusi selama proses pengujian, sehingga fungsi aplikasi dapat berjalan sesuai dengan kebutuhan pengguna.

5.4.1 Pengujian Fungsi *Login*

a. Potongan *Source Code* Fungsi *Login*

Fungsi *login* pada aplikasi sistem informasi karyawan dan presensi menggunakan *qr code* berbasis *android* digunakan untuk melakukan proses autentikasi pengguna sebelum masuk ke dalam sistem. Proses *login* dilakukan dengan memvalidasi *input email* dan *password*, melakukan autentikasi menggunakan *firebase authentication*, mengambil data pengguna dari *firebase firestore*, memeriksa *role* pengguna, mencatat aktivitas *login* ke dalam *collection log_aktivitas*, serta mengarahkan pengguna ke halaman *dashboard* sesuai *role*, yaitu *hrd*, pimpinan, atau karyawan. Berikut adalah potongan *source code* fungsi *login* yang digunakan dalam pengujian:

```
Future<void> _login() async {
  if (emailC.text.trim().isEmpty || passC.text.trim().isEmpty) {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('Email dan password wajib diisi')),
    );
    return;
  }
  setState(() => isLoading = true);
  try {
    final credential = await FirebaseAuth.instance.signInWithEmailAndPassword(
      email: emailC.text.trim(),
      password: passC.text.trim(),
    );
    final uid = credential.user!.uid;
    final userDoc =
      await FirebaseFirestore.instance.collection('users').doc(uid).get();
    if (!userDoc.exists) {
      await _simpanLogAktivitas(
        uid: uid,
        nama: '-',
        role: '-',

```

```

    aktivitas: 'Login',
    keterangan: 'Login gagal, data user tidak ditemukan di Firestore',
  );
  throw Exception('Data user tidak ditemukan di Firestore');
}
final data = userDoc.data!;
final role = (data['role'] ?? '').toString().toLowerCase();
final nama =
  (data['nama'] ?? data['name'] ?? emailC.text.trim()).toString();
Widget page;
if (role == 'hrd') {
  page = const HrdDashboardPage();
} else if (role == 'pimpinan') {
  page = const PimpinanDashboardPage();
} else if (role == 'karyawan') {
  page = const KaryawanDashboardPage();
} else {
  await _simpanLogAktivitas(
    uid: uid,
    nama: nama,
    role: role,
    aktivitas: 'Login',
    keterangan: 'Login gagal, role tidak dikenali',
  );
  throw Exception('Role tidak dikenali: $role');
}
await _simpanLogAktivitas(
  uid: uid,
  nama: nama,
  role: role,
  aktivitas: 'Login',
  keterangan: 'Berhasil login sebagai $role',

```

```

);
if (!mounted) return;
Navigator.pushReplacement(
  context,
  MaterialPageRoute(builder: (_) => page),
);
} on FirebaseAuthException catch (e) {
  String message = 'Login gagal';
  if (e.code == 'user-not-found') {
    message = 'Email tidak ditemukan';
  } else if (e.code == 'wrong-password') {
    message = 'Password salah';
  } else if (e.code == 'invalid-email') {
    message = 'Format email tidak valid';
  } else if (e.code == 'invalid-credential') {
    message = 'Email atau password salah';
  }
  await FirebaseFirestore.instance.collection('log_aktivitas').add({
    'uid': '-',
    'nama': emailC.text.trim(),
    'role': '-',
    'aktivitas': 'Login',
    'keterangan': message,
    'waktu': FieldValue.serverTimestamp(),
  });
  if (!mounted) return;
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text(message)),
  );
} catch (e) {
  if (!mounted) return;

```

```

ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(content: Text(e.toString())),
);
} finally {
  if (mounted) {
    setState(() => isLoading = false);
  }
}
}

```

b. Identifikasi *Statement* Fungsi *Login*

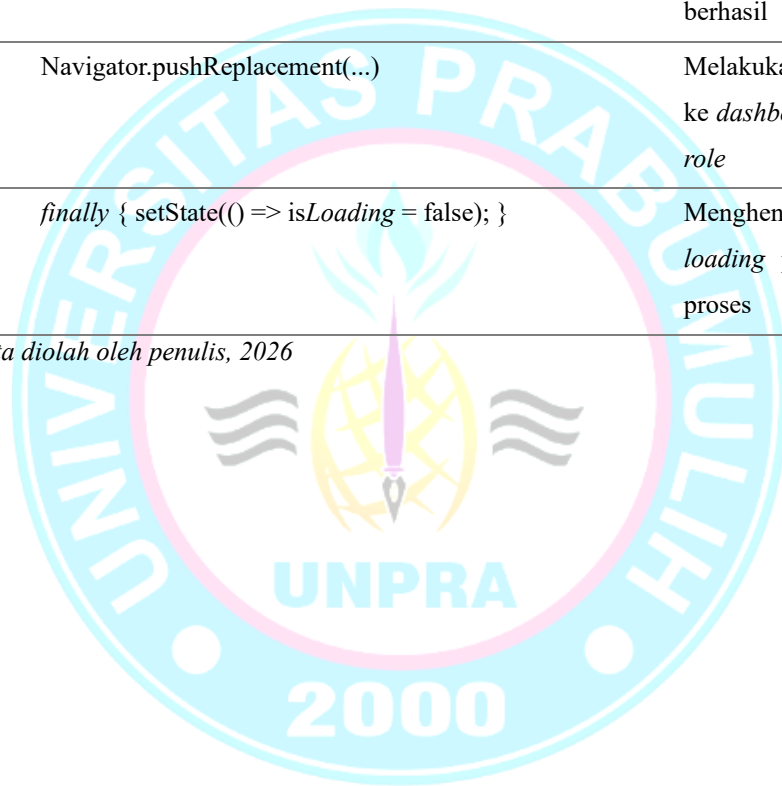
Tabel 5. 1 Identifikasi *Statement* Fungsi *Login*

<i>Statement</i>	<i>Source Code</i>	<i>Keterangan</i>
S1	<i>Input email dan password</i>	Mengambil <i>input email</i> dan <i>password</i> dari pengguna
S2	<i>if (emailC.text.trim().isEmpty passC.text.trim().isEmpty)</i>	Memeriksa apakah <i>email</i> atau <i>password</i> kosong
S3	<i>SnackBar('Email dan password wajib diisi')</i>	Menampilkan pesan <i>input</i> wajib diisi
S4	<i>setState(() => isLoading = true)</i>	Mengaktifkan status <i>loading</i>
S5	<i>FirebaseAuth.instance.signInWithEmailAndPassword(...)</i>	Melakukan autentikasi pengguna melalui <i>Firebase Authentication</i>
S6	Autentikasi berhasil?	Memeriksa apakah proses autentikasi berhasil atau gagal
S7	<i>on FirebaseAuthException catch (e)</i>	Menangkap <i>error</i> dari <i>Firebase Authentication</i>

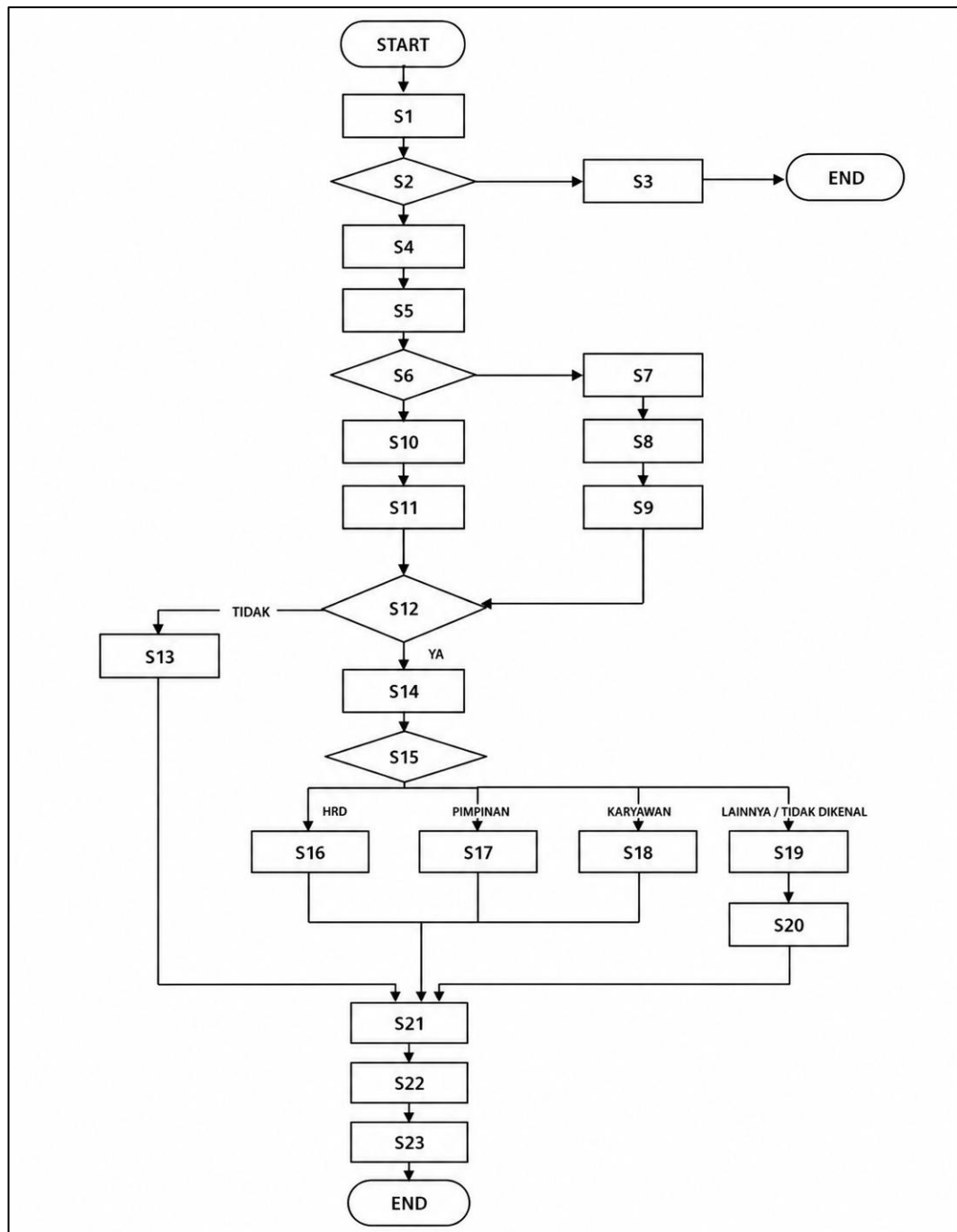
S8	<code>collection('log_aktivitas').add(...)</code>	Menyimpan log aktivitas <i>login</i> gagal
S9	<code>SnackBar(message)</code>	Menampilkan pesan <i>error login</i> kepada pengguna
S10	<code>final uid = credential.user!.uid</code>	Mengambil UID pengguna yang berhasil <i>login</i>
S11	<code>FirebaseFirestore.instance.collection('users').doc(uid).get()</code>	Mengambil data pengguna dari <i>collection users</i> di <i>Firestore</i>
S12	<code>if (!userDoc.exists)</code>	Memeriksa apakah data <i>user</i> ditemukan di <i>Firestore</i>
S13	<code>_simpanLogAktivitas(...)</code>	Menyimpan log <i>login</i> gagal karena data <i>user</i> tidak ditemukan
S14	<code>final role = (data['role'] ?? "").toString().toLowerCase()</code>	Mengambil <i>role</i> pengguna
S15	Pemeriksaan <i>role user</i>	Memeriksa <i>role</i> pengguna untuk menentukan <i>dashboard</i> tujuan
S16	<code>page = const HrdDashboardPage()</code>	Mengarahkan pengguna ke <i>dashboard HRD</i>
S17	<code>page = const PimpinanDashboardPage()</code>	Mengarahkan pengguna ke <i>dashboard Pimpinan</i>
S18	<code>page = const KaryawanDashboardPage()</code>	Mengarahkan pengguna ke

		<i>dashboard</i> Karyawan
S19	<code>_simpanLogAktivitas(...)</code> pada <i>role</i> tidak dikenali	Menyimpan log <i>login</i> gagal karena <i>role</i> tidak dikenali
S20	<code>throw Exception('Role tidak dikenali')</code>	Menampilkan atau menghasilkan <i>error</i> <i>role</i> tidak dikenali
S21	<code>_simpanLogAktivitas(...)</code> <i>login</i> berhasil	Menyimpan log aktivitas <i>login</i> berhasil
S22	<code>Navigator.pushReplacement(...)</code>	Melakukan navigasi ke <i>dashboard</i> sesuai <i>role</i>
S23	<code>finally { setState(() => isLoading = false); }</code>	Menghentikan <i>loading</i> pada akhir proses

Sumber: data diolah oleh penulis, 2026



c. *Flowchart Fungsi Login*



Sumber: data diolah oleh penulis, 2026

Gambar 5. 35 *Flowchart Fungsi Login*

Berdasarkan diagram alir pada gambar *flowchart* fungsi *login* di atas, urutan logika kerja fungsi *login* dapat dijabarkan sebagai berikut:

1. S1 dan S2 (*Input dan Validasi Email/Password*)

Sistem memulai proses *login* dengan mengambil *input email* dan *password* dari pengguna. Selanjutnya, sistem melakukan pengecekan apakah *email* atau *password* masih kosong.

2. S3 (*Input Kosong*)

Jika *email* atau *password* kosong, maka sistem menjalankan S3 untuk menampilkan pesan “*email dan password wajib diisi*”, kemudian proses *login* dihentikan.

3. S4 dan S5 (*Proses Autentikasi Firebase*)

Jika *email* dan *password* telah diisi, sistem menjalankan S4 untuk mengaktifkan *loading*. Selanjutnya, sistem menjalankan S5 untuk melakukan autentikasi menggunakan *firebase authentication*.

4. S7, S8, dan S9 (*Login Gagal Firebase*)

Jika proses autentikasi *firebase* gagal, maka sistem menjalankan S7 untuk menangkap *firebase auth exception* dan menentukan pesan *error* sesuai jenis kesalahan. Setelah itu, sistem menjalankan S8 untuk menyimpan *log* aktivitas *login* gagal, kemudian S9 untuk menampilkan pesan *error* kepada pengguna.

5. S10 dan S11 (*Pengambilan Data User*)

Jika autentikasi berhasil, sistem menjalankan S10 untuk mengambil *uid* pengguna. Selanjutnya, sistem menjalankan S11 untuk mengambil data pengguna dari *collection users* di *firebase firestore*.

6. S12 dan S13 (*Data User Tidak Ditemukan*)

Pada S12, sistem memeriksa apakah data *user* ditemukan. Jika data *user* tidak ditemukan, maka sistem menjalankan S13 untuk menyimpan *log* aktivitas bahwa data *user* tidak ditemukan, kemudian proses *login* dihentikan.

7. S14 dan S15 (*Pengecekan Role User*)

Jika data *user* ditemukan, sistem menjalankan S14 untuk mengambil *role* pengguna. Selanjutnya, sistem menjalankan S15 untuk memeriksa *role* pengguna sebagai dasar penentuan halaman *dashboard*.

8. S16, S17, dan S18 (Navigasi Berdasarkan *Role*)

Jika *role* pengguna adalah *hrd*, maka sistem menjalankan S16 untuk mengarahkan pengguna ke *hrd dashboard page*. Jika *role* adalah pimpinan, sistem menjalankan S17 untuk mengarahkan pengguna ke pimpinan *dashboard page*. Jika *role* adalah karyawan, sistem menjalankan S18 untuk mengarahkan pengguna ke karyawan *dashboard page*.

9. S19 dan S20 (*Role* Tidak Dikenali)

Jika *role* pengguna tidak sesuai dengan *hrd*, pimpinan, atau karyawan, maka sistem menjalankan S19 untuk menyimpan *log* aktivitas bahwa *role* tidak dikenali. Setelah itu, sistem menjalankan S20 untuk menghentikan proses dengan pesan *error role* tidak dikenali.

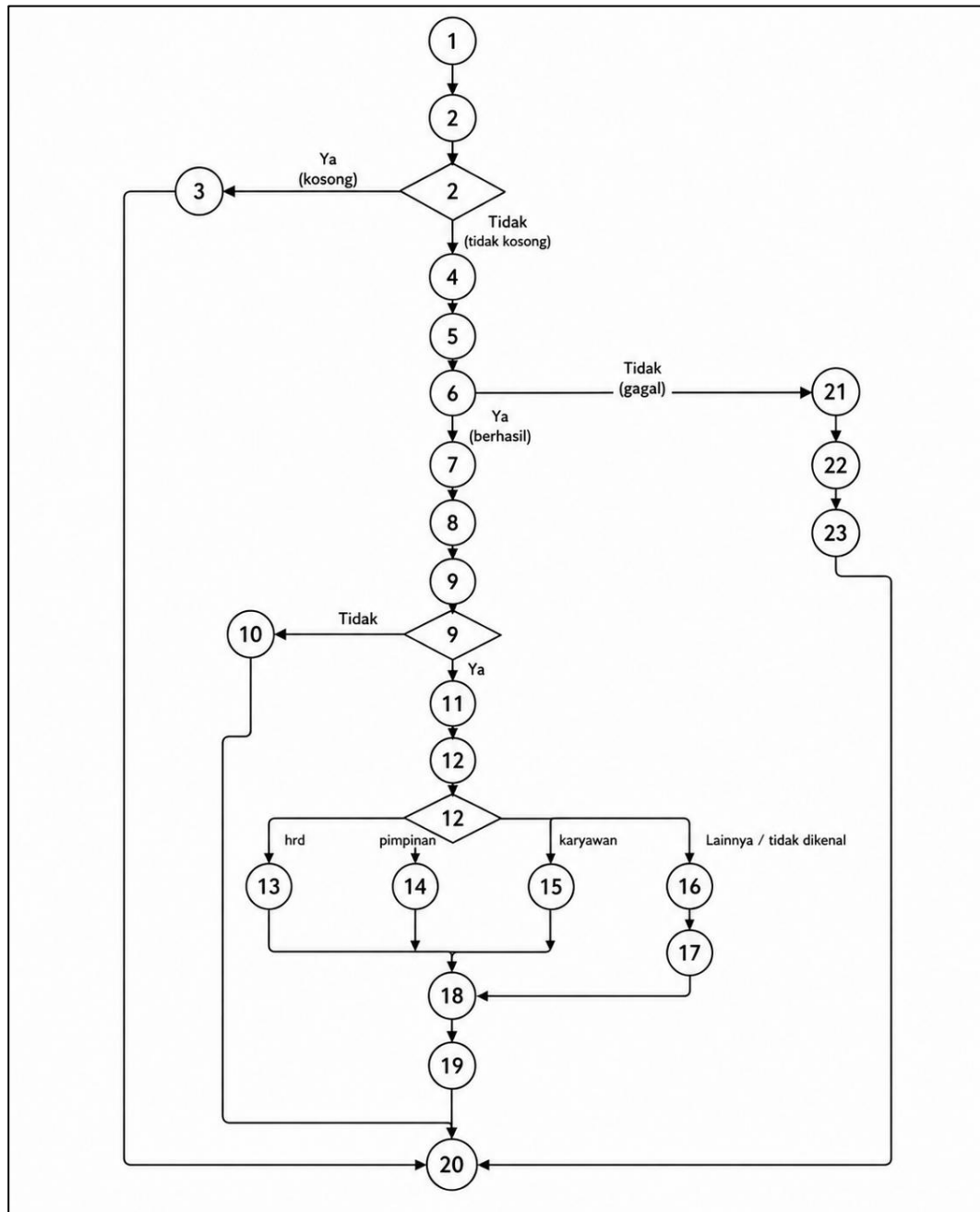
10. S21 dan S22 (*Login* Berhasil)

Jika *role* dikenali dan halaman *dashboard* telah ditentukan, sistem menjalankan S21 untuk menyimpan *log* aktivitas *login* berhasil. Selanjutnya, sistem menjalankan S22 untuk melakukan navigasi ke *dashboard* sesuai *role* pengguna.

11. S23 (Akhir Proses)

Pada akhir proses, sistem menjalankan S23 untuk menghentikan *loading* melalui blok *finally*, baik ketika *login* berhasil maupun ketika terjadi kegagalan.

d. *Flowgraph Fungsi Login*



Sumber: data diolah oleh penulis 2026

Gambar 5. 36 *Flowgraph Fungsi Login*

Flowgraph fungsi *login* menggambarkan alur eksekusi program berdasarkan *node* dan jalur yang telah diidentifikasi. Proses dimulai dari *node* 1, yaitu pengguna memasukkan *email* dan *password*. Selanjutnya sistem menuju *node* 2 untuk melakukan pengecekan apakah *email* atau *password* masih kosong. Jika kondisi pada *node* 2 bernilai benar atau *input* masih kosong, maka alur menuju *node* 3 untuk

menampilkan pesan bahwa *email* dan *password* wajib diisi. Setelah pesan ditampilkan, proses *login* langsung berakhir pada *node 20*.

Jika *email* dan *password* telah diisi, maka alur dilanjutkan menuju *node 4* untuk mengaktifkan *loading*. Selanjutnya pada *node 5* sistem menjalankan proses autentikasi menggunakan *firebase authentication*. Pada *node 6* sistem memeriksa apakah autentikasi berhasil. Jika autentikasi gagal, maka alur menuju *node 21* untuk menentukan pesan *error* sesuai jenis kesalahan *firebase authentication*. Setelah itu sistem menjalankan *node 22* untuk menyimpan log aktivitas *login* gagal, kemudian *node 23* untuk menampilkan pesan *error* kepada pengguna. Setelah proses selesai, alur berakhir pada *node 20* untuk menghentikan *loading*.

Jika autentikasi berhasil, maka alur dilanjutkan menuju *node 7* untuk mengambil uid pengguna. Selanjutnya pada *node 8* sistem mencari data pengguna pada *collection users* di *firebase firestore*. Pada *node 9* sistem memeriksa apakah data *user* ditemukan. Jika data *user* tidak ditemukan, maka alur menuju *node 10* untuk menyimpan log aktivitas bahwa data *user* tidak ditemukan. Setelah itu proses berakhir pada *node 20*. Jika data *user* ditemukan, sistem melanjutkan proses ke *node 11* untuk mengambil *role* pengguna. Selanjutnya pada *node 12* sistem memeriksa jenis *role* pengguna.

Jika *role* pengguna adalah *hrd*, maka alur menuju *node 13* untuk mengarahkan pengguna ke halaman *hrd dashboard page*. Jika *role* pengguna adalah pimpinan, maka alur menuju *node 14* untuk mengarahkan pengguna ke halaman pimpinan *dashboard page*. Jika *role* pengguna adalah karyawan, maka alur menuju *node 15* untuk mengarahkan pengguna ke halaman karyawan *dashboard page*. Namun jika *role* pengguna tidak dikenali, maka alur menuju *node 16* untuk menyimpan log aktivitas bahwa *role* tidak dikenali. Setelah itu sistem menjalankan *node 17* untuk menampilkan pesan *error role* tidak dikenali, kemudian proses berakhir pada *node 20*.

Jika *role* valid dan pengguna berhasil diarahkan ke *dashboard* sesuai *role*, maka sistem menjalankan *node 18* untuk menyimpan log aktivitas *login* berhasil. Selanjutnya pada *node 19* sistem melakukan navigasi ke *dashboard* sesuai *role* pengguna. Pada akhir proses, seluruh alur berakhir pada *node 20* untuk menghentikan *loading* melalui blok *finally*.

e. *Cyclomatic Complexity Fungsi Login*

Berdasarkan *flowgraph* yang telah dibentuk, terdapat 7 *predicate node* (P) yang menjadi titik keputusan pada fungsi *login*. *Predicate node* tersebut yaitu *node* 2 yang melakukan pengecekan apakah *email* atau *password* kosong, *node* 6 yang melakukan pengecekan keberhasilan autentikasi *firebase authentication*, *node* 9 yang melakukan pengecekan keberadaan data *user* pada *collection users*, *node* 12 yang melakukan pengecekan *role hrd*, *node* 14 yang melakukan pengecekan *role pimpinan*, *node* 16 yang melakukan pengecekan *role karyawan*, serta *node* 21 yang menangani kondisi *error* dari *firebase authentication*. Oleh karena itu, nilai *cyclomatic complexity* dapat dihitung menggunakan rumus berikut:

$$V(G) = P + 1$$

$$V(G) = 7 + 1$$

$$V(G) = 8$$

Dari perhitungan tersebut diperoleh nilai *cyclomatic complexity* sebesar 8. Nilai tersebut menunjukkan bahwa terdapat delapan jalur independen dalam proses eksekusi fungsi *login* yang merepresentasikan seluruh kemungkinan alur logika, mulai dari *input* kosong, autentikasi gagal, data *user* tidak ditemukan, *role* tidak dikenali, hingga *login* berhasil sebagai *hrd*, *pimpinan*, atau *karyawan*. Nilai *cyclomatic complexity* tersebut kemudian digunakan sebagai dasar untuk menentukan jumlah *independent path* yang akan diuji pada tahap pengujian berikutnya, sehingga setiap jalur eksekusi dapat diverifikasi untuk memastikan seluruh kondisi pada fungsi *login* telah ditangani dengan benar.

f. *Independent Path Fungsi Login*

Berdasarkan hasil perhitungan *cyclomatic complexity* yang diperoleh dari *flowgraph* fungsi *login*, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses *white box testing*. Setiap jalur independen merepresentasikan kemungkinan alur eksekusi program yang berbeda berdasarkan kondisi validasi *input*, proses autentikasi *firebase authentication*, pengecekan data pengguna pada *firebase firestore*, pengecekan *role* pengguna, serta penanganan *error* pada sistem *login*. Dengan demikian, setiap kemungkinan skenario pada fungsi *login* dapat dianalisis dan diuji secara

menyeluruh untuk memastikan seluruh logika program berjalan dengan benar. Berdasarkan *flowgraph* fungsi *login*, diperoleh 8 jalur independen (*independent path*) sebagai berikut:

1. *Path 1:*

$1 \rightarrow 2 \rightarrow 3 \rightarrow 23$

Email atau *password* kosong sehingga sistem menampilkan pesan bahwa *email* dan *password* wajib diisi.

2. *Path 2:*

$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 23$

Proses autentikasi *firebase* gagal sehingga sistem menyimpan *log login* gagal dan menampilkan pesan *error login*.

3. *Path 3:*

$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 23$

Autentikasi berhasil, tetapi data *user* tidak ditemukan pada *collection users* di *firestore* sehingga proses *login* dihentikan.

4. *Path 4:*

$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 14 \rightarrow 15 \rightarrow 16 \rightarrow 21 \rightarrow 22 \rightarrow 23$

Login berhasil dengan *role hrd* sehingga pengguna diarahkan ke *hrd dashboard page*.

5. *Path 5:*

$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 14 \rightarrow 15 \rightarrow 17 \rightarrow 21 \rightarrow 22 \rightarrow 23$

Login berhasil dengan *role pimpinan* sehingga pengguna diarahkan ke *pimpinan dashboard page*.

6. *Path 6:*

$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 14 \rightarrow 15 \rightarrow 18 \rightarrow 21 \rightarrow 22 \rightarrow 23$

Login berhasil dengan *role karyawan* sehingga pengguna diarahkan ke *karyawan dashboard page*.

7. *Path 7:*

$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 14 \rightarrow 15 \rightarrow 19 \rightarrow 20 \rightarrow 23$

Role pengguna tidak dikenali sehingga sistem menyimpan *log* gagal dan menampilkan pesan *error role* tidak dikenali.

8. *Path* 8:

1 → 2 → 4 → 5 → 10 → 11 → 12 → 14 → 15 → 19 → 20 → 23

Terjadi *error* umum pada proses *login* sehingga sistem menampilkan pesan *error* kepada pengguna sebelum proses berakhir.

g. *Test Case Fungsi Login*

Pengujian dilakukan berdasarkan jalur independen (*independent path*) yang telah diperoleh dari hasil analisis *flowgraph* fungsi *login*. Setiap jalur independen direpresentasikan ke dalam beberapa *test case* dengan kombinasi *input email* dan *password* yang berbeda untuk menguji seluruh kemungkinan alur eksekusi program. Jalur eksekusi pada masing-masing *test case* diperoleh dari *node-node* yang dilalui pada *flowgraph* sehingga setiap proses pengujian dapat memverifikasi fungsi *login* secara menyeluruh.

Tabel 5. 2 *Test Case Fungsi Login*

<i>Test Case</i>	<i>Email</i>	<i>Password</i>	Jalur Eksekusi	<i>Output</i>
TC1	Kosong	Kosong	1 → 2 → 3 → 23	Muncul pesan “ <i>Email</i> dan <i>password</i> wajib diisi”
TC2	<i>Email</i> tidak terdaftar	<i>Password</i> valid	1 → 2 → 4 → 5 → 7 → 8 → 9 → 23	<i>Login</i> gagal dan muncul pesan “ <i>Email</i> tidak ditemukan”
TC3	<i>Format email</i> salah	<i>Password</i> valid	1 → 2 → 4 → 5 → 7 → 8 → 9 → 23	<i>Login</i> gagal dan muncul pesan “ <i>Format email</i> tidak valid”
TC4	<i>Email</i> valid	<i>Password</i> salah	1 → 2 → 4 → 5 → 7 → 8 → 9 → 23	<i>Login</i> gagal dan muncul pesan “ <i>Password</i> salah”
TC5	<i>Email</i> valid <i>role HRD</i>	<i>Password</i> benar	1 → 2 → 4 → 5 → 10 → 11 → 12 → 14 → 15 → 16 → 21 → 22 → 23	<i>Login</i> berhasil dan diarahkan ke <i>HrdDashboardPage</i>
TC6	<i>Email</i> valid <i>role Pimpinan</i>	<i>Password</i> benar	1 → 2 → 4 → 5 → 10 → 11 → 12 → 14 → 15 → 17 → 21 → 22 → 23	<i>Login</i> berhasil dan diarahkan ke <i>PimpinanDashboardPage</i>

TC7	Email role Karyawan	valid	Password benar	1 → 2 → 4 → 5 → 10 → 11 → 12 → 14 → 15 → 18 → 21 → 22 → 23	Login berhasil dan diarahkan ke KaryawanDashboardPage
TC8	Email tetapi tidak dikenali	valid role benar	Password benar	1 → 2 → 4 → 5 → 10 → 11 → 12 → 14 → 15 → 19 → 20 → 23	Login gagal dan muncul pesan “Role tidak dikenali”

Sumber : data diolah oleh penulis, 2026

h. Statement Coverage Fungsi Login

Statement coverage digunakan untuk mengukur sejauh mana seluruh *statement* pada fungsi *login* telah dieksekusi melalui *test case* yang telah dirancang. Pengujian ini bertujuan untuk memastikan bahwa setiap *statement* pada program telah dijalankan minimal satu kali sehingga seluruh logika program dapat diverifikasi dengan baik. *Statement* yang tereksekusi pada masing-masing *test case* adalah sebagai berikut:

1. TC1:
S1, S2, S3, S23
2. TC2:
S1, S2, S4, S5, S7, S8, S9, S23
3. TC3:
S1, S2, S4, S5, S7, S8, S9, S23
4. TC4:
S1, S2, S4, S5, S7, S8, S9, S23
5. TC5:
S1, S2, S4, S5, S10, S11, S12, S14, S15, S16, S21, S22, S23
6. TC6:
S1, S2, S4, S5, S10, S11, S12, S14, S15, S17, S21, S22, S23
7. TC7:
S1, S2, S4, S5, S10, S11, S12, S14, S15, S18, S21, S22, S23
8. TC8:
S1, S2, S4, S5, S10, S11, S12, S14, S15, S19, S20, S23

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *statement* pada fungsi *login* yaitu S1 hingga S23 berhasil dieksekusi melalui kombinasi *test case* TC1 sampai TC8. Dengan demikian diperoleh nilai *statement coverage* sebesar:
 $(23 / 23) \times 100\% = 100\%$

Nilai tersebut menunjukkan bahwa seluruh *statement* pada fungsi *login* telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses pengujian *white box testing*.

5.4.2 Pengujian Fungsi Log Out

a. Potongan Source Code Fungsi Log Out

Fungsi *log out* digunakan untuk mengeluarkan pengguna dari aplikasi. Pada fungsi ini, sistem menjalankan proses keluar akun menggunakan *firebase authentication*. Setelah proses *log out* berhasil, sistem memeriksa apakah halaman masih aktif. Jika halaman masih aktif, pengguna diarahkan kembali ke halaman *login* menggunakan *navigator push and remove until ()* agar pengguna tidak dapat kembali ke halaman sebelumnya setelah keluar. Berikut adalah potongan *source code* fungsi *log out* yang digunakan dalam pengujian:

```
Future<void> _logout(BuildContext context) async {
  await FirebaseAuth.instance.signOut();
  if (!context.mounted) return;
  Navigator.pushAndRemoveUntil(
    context,
    MaterialPageRoute(builder: (_) => const LoginPage()),
    (_) => false,
  );
}
```

b. Identifikasi Statement Fungsi Log Out

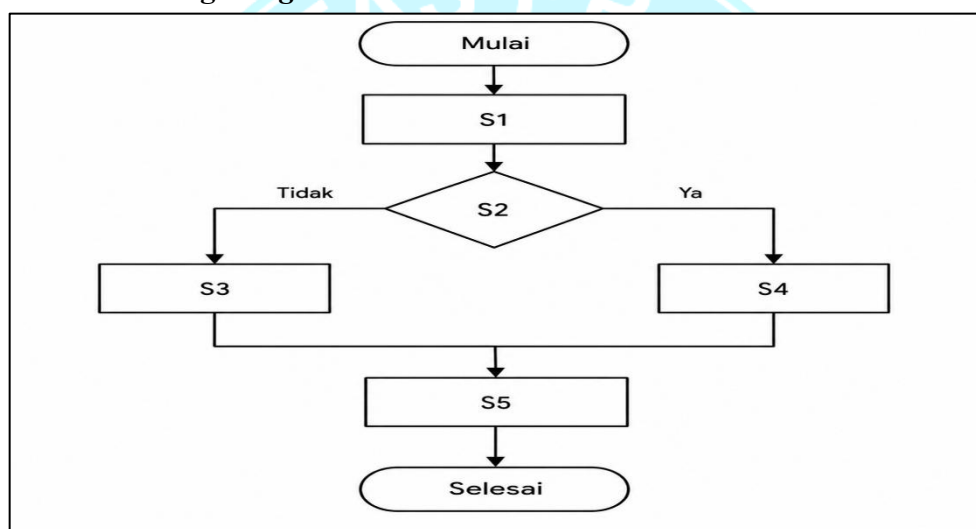
Tabel 5. 3 Identifikasi *Statement* Fungsi Log Out

<i>Statement</i>	<i>Source Code</i>	<i>Keterangan</i>
S1	<i>FirebaseAuth.instance.signOut()</i>	Menghapus sesi <i>login</i> pengguna dari <i>Firebase Authentication</i>

S2	<code>if (!context.mounted)</code>	Memeriksa apakah halaman masih aktif
S3	<code>return</code>	Menghentikan proses jika halaman sudah tidak aktif
S4	<code>Navigator.pushAndRemoveUntil(...)</code>	Mengarahkan pengguna ke halaman <i>Login</i>
S5	<code>() => false</code>	Menghapus seluruh riwayat halaman sebelumnya sehingga pengguna tidak dapat kembali ke halaman aplikasi setelah log out

Sumber : data diolah oleh penulis, 2026

c. Flowchart Fungsi Log Out



Sumber: data diolah oleh penulis, 2026

Gambar 5. 37 Flowchart Fungsi Log Out

Berdasarkan diagram alir pada gambar *flowchart* fungsi *log out*, urutan logika kerja fungsi *log out* dapat dijabarkan sebagai berikut:

1. S1 (Proses *Sign Out*)

Sistem menjalankan proses keluar akun menggunakan *firebase auth*. *Instance sign out ()* untuk menghapus sesi *login* pengguna dari *firebase authentication*.

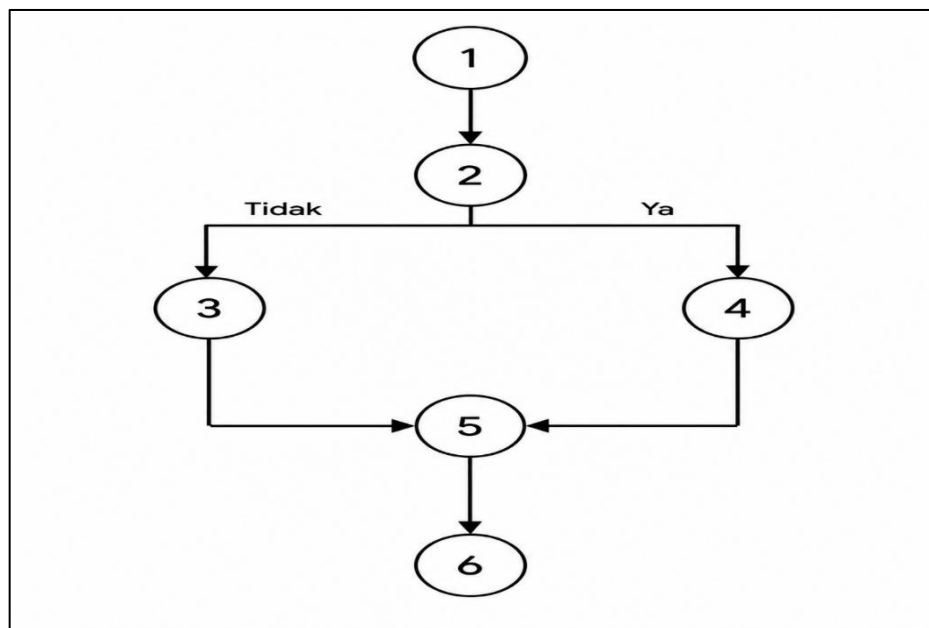
2. S2 dan S3 (Pengecekan Halaman Aktif)

Setelah proses *sign out* selesai, sistem memeriksa apakah halaman masih aktif menggunakan *context mounted*. Jika halaman sudah tidak aktif, maka proses dihentikan.

3. S4 dan S5 (Navigasi ke Halaman *Login*)

Jika halaman masih aktif, sistem mengarahkan pengguna ke halaman *login page* menggunakan *navigator push and remove until ()*. Seluruh riwayat halaman sebelumnya dihapus agar pengguna tidak dapat kembali ke halaman aplikasi setelah *log out*.

d. *Flowgraph Fungsi Log Out*



Sumber: data diolah oleh penulis, 2026

Gambar 5. 38 *Flowgraph Fungsi Log Out*

Berdasarkan gambar *flowgraph* fungsi *log out*, alur logika program fungsi *log out* dapat dijelaskan sebagai berikut. Proses dimulai pada *node* 1, yaitu sistem menjalankan *firebase auth instance.sign out ()* untuk menghapus sesi *login* pengguna dari *firebase authentication*. Setelah proses keluar akun selesai, sistem melanjutkan ke *node* 2 untuk memeriksa apakah halaman masih aktif menggunakan *context mounted*.

Jika kondisi pada *node* 2 bernilai tidak, maka alur menuju *node* 3 untuk menghentikan proses dengan *return*. Jika kondisi bernilai ya, maka sistem menjalankan *node* 4 untuk mengarahkan pengguna ke halaman *login page* menggunakan *navigator push and remove until ()*. Pada proses tersebut, *node* 5

digunakan untuk menghapus seluruh riwayat halaman sebelumnya agar pengguna tidak dapat kembali ke halaman aplikasi setelah melakukan *log out*.

Dengan demikian, *flowgraph* fungsi *log out* memiliki dua kemungkinan alur utama, yaitu proses dihentikan ketika halaman sudah tidak aktif, atau pengguna berhasil diarahkan kembali ke halaman *login* setelah sesi *login* dihapus.

e. Cyclomatic Complexity Fungsi Log Out

Berdasarkan *flowgraph* yang telah dibentuk, terdapat 1 *predicate node* (P) pada fungsi *log out*, yaitu *node 2* yang memeriksa kondisi *context.mounted*. Maka nilai *cyclomatic complexity* dapat dihitung menggunakan rumus berikut:

$$V(G) = P + 1$$

$$V(G) = 1 + 1$$

$$V(G) = 2$$

Dari hasil perhitungan tersebut diperoleh nilai *cyclomatic complexity* sebesar 2. Nilai tersebut menunjukkan bahwa terdapat dua jalur independen yang dapat dilalui selama proses *log out* berlangsung. Nilai kompleksitas yang rendah menunjukkan bahwa fungsi *log out* memiliki logika yang sederhana karena hanya terdiri dari proses keluar akun dan satu percabangan untuk memeriksa apakah halaman masih aktif sebelum melakukan navigasi ke halaman *login*.

f. Independent Path Fungsi Log Out

Berdasarkan hasil perhitungan *cyclomatic complexity* fungsi *log out*, diperoleh nilai $V(G) = 2$. Nilai tersebut menunjukkan bahwa terdapat dua jalur independen yang harus diuji dalam proses *white box testing*. Adapun jalur independen fungsi *log out* adalah sebagai berikut:

1. *Path 1:*

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 6$$

Halaman sudah tidak aktif sehingga proses dihentikan dengan *return*.

2. *Path 2:*

$$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6$$

Halaman masih aktif sehingga pengguna diarahkan ke halaman *login* dan seluruh riwayat halaman sebelumnya dihapus.

g. Test Case Pengujian Fungsi Log Out

Pengujian fungsi *log out* dilakukan berdasarkan jalur independen yang telah diperoleh dari hasil analisis *flowgraph*. Setiap *test case* digunakan untuk menguji proses keluar akun dan perpindahan halaman setelah sesi *login* dihapus.

Tabel 5. 4 Test Case Fungsi Log Out

Test Case	Kondisi	Jalur Eksekusi	Output
TC1	Halaman sudah tidak aktif (<i>context.mounted</i> = false)	1 → 2 → 3 → 6	Proses dihentikan dengan <i>return</i> dan tidak terjadi perpindahan halaman
TC2	Halaman masih aktif (<i>context.mounted</i> = true)	1 → 2 → 4 → 5 → 6	Pengguna berhasil keluar dari akun, diarahkan ke halaman <i>login</i> , dan seluruh riwayat halaman sebelumnya dihapus

Sumber: data diolah oleh penulis, 2026

h. Statement Coverage Fungsi Log Out

Statement coverage digunakan untuk mengukur sejauh mana seluruh *statement* pada fungsi *log out* telah dieksekusi melalui *test case* yang telah dirancang. Pengujian ini bertujuan untuk memastikan setiap *statement* pada proses keluar akun, pengecekan halaman aktif, dan perpindahan ke halaman *login* telah dijalankan minimal satu kali. *Statement* yang tereksekusi pada masing-masing *test case* adalah sebagai berikut:

1. TC1:
S1, S2, S3
2. TC2:
S1, S2, S4, S5

Berdasarkan hasil pengujian, seluruh *statement* pada fungsi *log out* yaitu S1 sampai S5 berhasil dieksekusi melalui kombinasi *test case* TC1 dan TC2. Dengan demikian diperoleh nilai *statement coverage* sebesar:

$$(5 / 5) \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *statement* pada fungsi *log out* telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam

proses *white box testing*. Dengan demikian, fungsi *log out* dapat dinyatakan telah memenuhi kriteria pengujian *white box* berdasarkan metode *statement coverage*.

5.4.3 Pengujian Fungsi Lupa Password

a. Potongan Source Code Fungsi Lupa Password

Fungsi lupa *password* digunakan untuk membantu pengguna mengatur ulang *password* akun apabila pengguna lupa kata sandi yang digunakan untuk *login*. Pada aplikasi ini, proses *reset password* dilakukan menggunakan *firebase authentication* dengan cara mengirimkan *link reset password* ke *email* pengguna yang telah terdaftar. alur fungsi dimulai dengan memeriksa apakah *field email* sudah diisi atau belum.

Jika *email* masih kosong, sistem akan menampilkan pesan bahwa *email* wajib diisi. Jika *email* telah diisi, sistem mengaktifkan *loading* dan mengirimkan *link reset password* melalui *firebase authentication*. Apabila proses berhasil, sistem menampilkan pesan bahwa *link reset password* telah dikirim dan pengguna diarahkan kembali ke halaman sebelumnya. Namun, jika terjadi kesalahan seperti *email* tidak terdaftar atau *format email* tidak valid, sistem akan menampilkan pesan *error* sesuai kondisi yang terjadi. Berikut adalah potongan *source code* fungsi lupa *password* yang digunakan dalam pengujian:

```
Future<void> _ kirimResetPassword() async {
  if (emailC.text.trim().isEmpty) {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('Email wajib diisi')),
    );
    return;
  }
  setState(() => isLoading = true);
  try {
    await FirebaseAuth.instance.sendPasswordResetEmail(
      email: emailC.text.trim(),
    );
    if (!mounted) return;
```

```

ScaffoldMessenger.of(context).showSnackBar(
  const SnackBar(
    content: Text(
      'Link reset password telah dikirim. Cek Inbox, Spam, atau Promotions.',
    ),
  ),
);
Navigator.pop(context);
} on FirebaseAuthException catch (e) {
  String message = 'Gagal mengirim link reset password';
  if (e.code == 'user-not-found') {
    message = 'Email tidak terdaftar';
  } else if (e.code == 'invalid-email') {
    message = 'Format email tidak valid';
  }
  if (!mounted) return;
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text(message)),
  );
} finally {
  if (mounted) setState(() => isLoading = false);
}
}

```

b. Identifikasi *Statement* Fungsi Lupa Password

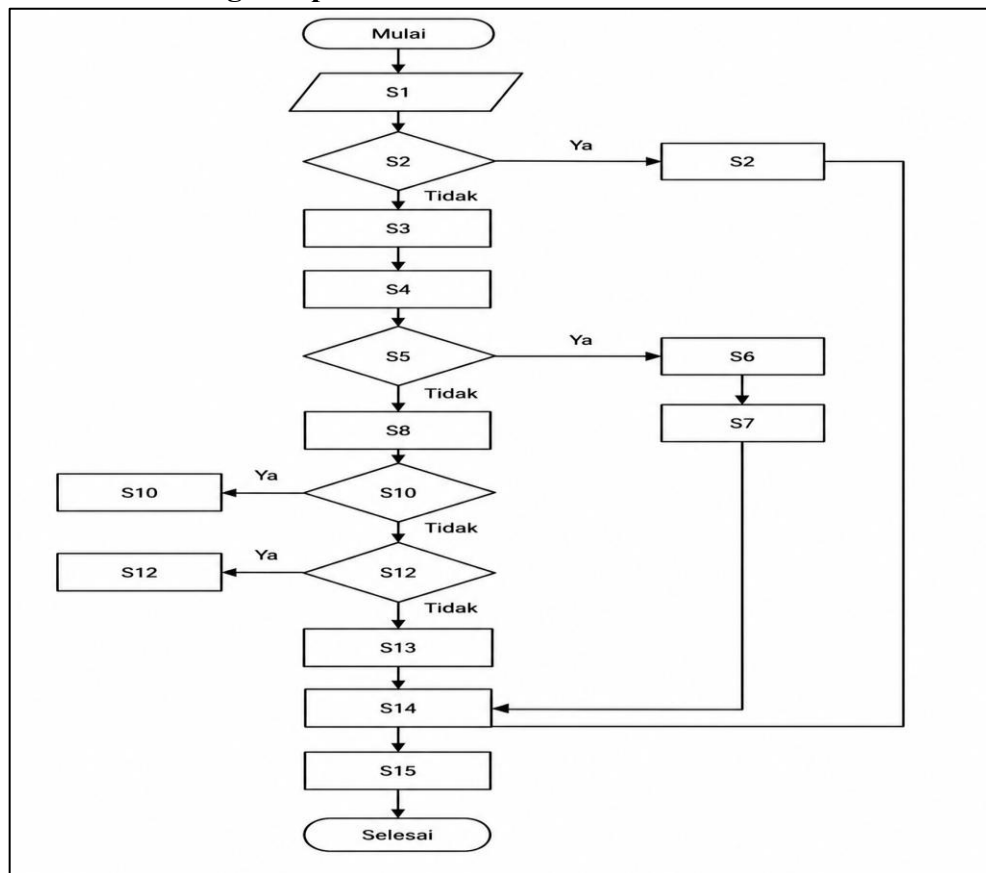
Tabel 5. 5 Identifikasi *Statement* Fungsi Lupa Password

<i>Statement</i>	<i>Source Code</i>	<i>Keterangan</i>
S1	if (emailC.text.trim().isEmpty)	Memeriksa apakah email kosong
S2	SnackBar('Email wajib diisi')	Menampilkan pesan email wajib diisi

S3	<code>setState(() => isLoading = true)</code>	Mengaktifkan status <i>loading</i>
S4	<code>FirebaseAuth.instance.sendPasswordResetEmail (...)</code>	Mengirim <i>link reset password</i> melalui <i>Firebase Authentication</i>
S5	<code>if (!mounted) return</code>	Memeriksa apakah halaman masih aktif
S6	<code>SnackBar('Link reset password telah dikirim...')</code>	Menampilkan pesan berhasil mengirim <i>link reset password</i>
S7	<code>Navigator.pop(context)</code>	Kembali ke halaman sebelumnya
S8	<code>on FirebaseAuthException catch (e)</code>	Menangkap <i>error FirebaseAuth</i>
S9	<code>if (e.code == 'user-not-found')</code>	Memeriksa <i>email</i> tidak terdaftar
S10	<code>message = 'Email tidak terdaftar'</code>	Menentukan pesan <i>email</i> tidak terdaftar
S11	<code>else if (e.code == 'invalid-email')</code>	Memeriksa <i>format email</i> tidak valid
S12	<code>message = 'Format email tidak valid'</code>	Menentukan pesan <i>format email</i> tidak valid
S13	<code>SnackBar(content: Text(message))</code>	Menampilkan pesan <i>error</i>
S14	<code>finally</code>	Menjalankan proses akhir
S15	<code>setState(() => isLoading = false)</code>	Menonaktifkan <i>loading</i>

Sumber : data diolah oleh penulis, 2026

c. *Flowchart Fungsi Lupa Password*



Sumber : data diolah oleh penulis, 2026

Gambar 5. 39 *Flowchart Fungsi Lupa Password*

Berdasarkan diagram alir pada gambar *flowchart* fungsi lupa *password*, urutan logika kerja fungsi lupa *password* dapat dijabarkan sebagai berikut:

1. S1 dan S2 (Validasi *Email Kosong*)

Sistem memulai proses lupa *password* dengan memeriksa apakah *email* telah diisi oleh pengguna. Jika *email* kosong, maka sistem menjalankan S2 untuk menampilkan pesan “*email* wajib diisi”, kemudian proses dihentikan.

2. S3 dan S4 (Proses *Reset Password*)

Jika *email* telah diisi, sistem menjalankan S3 untuk mengaktifkan *loading*. Selanjutnya, sistem menjalankan S4 untuk mengirim *link reset password* menggunakan *firebase authentication*.

3. S5, S6, dan S7 (*Reset Password Berhasil*)

Jika proses pengiriman *link* berhasil, sistem menjalankan S5 untuk memastikan halaman masih aktif. Setelah itu, sistem menjalankan S6 untuk

menampilkan pesan bahwa *link reset password* telah dikirim. Selanjutnya, sistem menjalankan S7 untuk kembali ke halaman sebelumnya.

4. S8, S9, dan S10 (*Email Tidak Terdaftar*)

Jika terjadi *firebase auth exception* dengan kode *user-not-found*, maka sistem menjalankan S8 untuk menangkap *error firebase*. Selanjutnya, sistem menjalankan S9 dan S10 untuk menentukan pesan bahwa *email* tidak terdaftar.

5. S11 dan S12 (*Format Email Tidak Valid*)

Jika terjadi *firebase auth exception* dengan kode *invalid-email*, maka sistem menjalankan S11 dan S12 untuk menentukan pesan bahwa *format email* tidak valid.

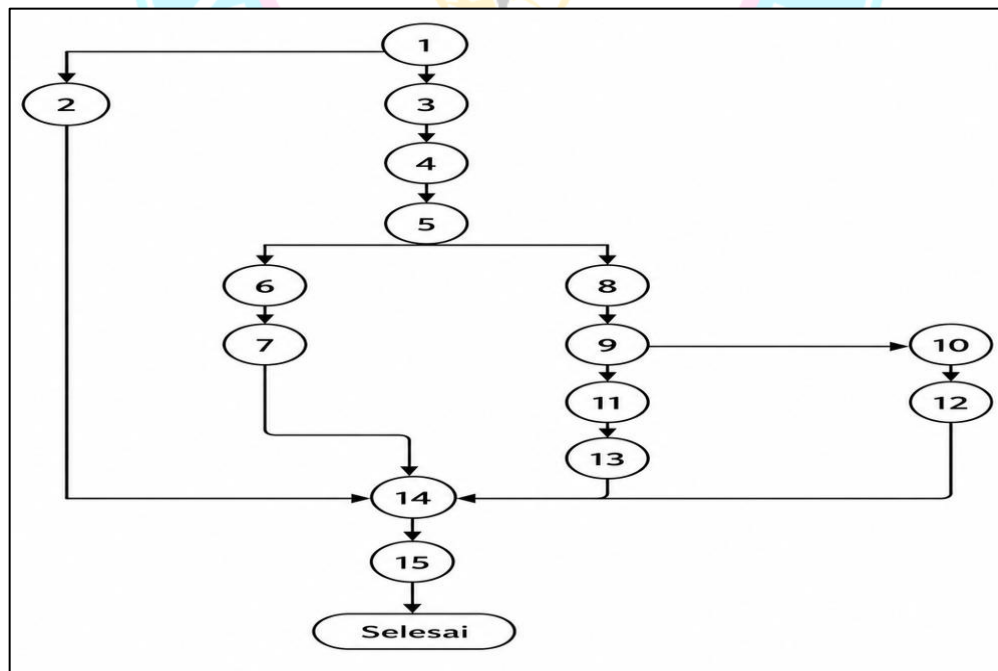
6. S13 (Menampilkan Pesan *Error*)

Setelah pesan *error* ditentukan, sistem menjalankan S13 untuk menampilkan pesan *error* kepada pengguna.

7. S14 dan S15 (Akhir Proses)

Pada akhir proses, sistem menjalankan S14 dan S15 untuk menonaktifkan *loading* melalui blok *finally*.

d. **Flowgraph Fungsi Lupa Password**



Sumber: data diolah oleh penulis, 2026

Gambar 5. 40 Flowgraph Fungsi Lupa Password

Flowgraph fungsi lupa *password* menggambarkan alur eksekusi program berdasarkan *statement* yang telah diidentifikasi. Alur dimulai dari validasi *email* kosong, proses pengiriman *link reset password* menggunakan *firebase authentication*, penanganan *error firebase*, hingga proses akhir untuk menghentikan *loading*. *Node* 1 digunakan untuk memeriksa apakah *email* kosong. Jika kondisi bernilai benar, maka alur menuju *node* 2 untuk menampilkan pesan *email* wajib diisi, kemudian proses berakhir pada *node* 15.

Jika *email* telah diisi, alur dilanjutkan ke *node* 3 untuk mengaktifkan *loading* dan *node* 4 untuk mengirim *link reset password* menggunakan *firebase authentication*. Jika proses berhasil, sistem melanjutkan ke *node* 5 untuk memeriksa apakah halaman masih aktif. Selanjutnya, sistem menjalankan *node* 6 untuk menampilkan pesan berhasil mengirim *link reset password* dan *node* 7 untuk kembali ke halaman sebelumnya.

Jika terjadi *firebase auth exception*, alur menuju *node* 8 untuk menangkap *error firebase*. Jika *error* berupa *user-not-found*, maka alur menuju *node* 9 dan *node* 10 untuk menentukan pesan *email* tidak terdaftar. Jika *error* berupa *invalid-email*, maka alur menuju *node* 11 dan *node* 12 untuk menentukan pesan *format email* tidak valid. Setelah pesan *error* ditentukan, sistem menjalankan *node* 13 untuk menampilkan pesan *error* kepada pengguna. Seluruh jalur proses berakhir pada *node* 14 dan *node* 15 untuk menghentikan *loading*.

e. Cyclomatic Complexity Fungsi Lupa Password

Berdasarkan *flowgraph* yang telah dibentuk, terdapat 4 *predicate node* (P) pada fungsi lupa *password*, yaitu *node* 1 yang melakukan pengecekan *email* kosong, *node* 5 yang memeriksa apakah halaman masih aktif, *node* 9 yang memeriksa *error user-not-found*, dan *node* 11 yang memeriksa *error invalid-email*. Oleh karena itu, nilai *cyclomatic complexity* dapat dihitung menggunakan rumus berikut:

$$V(G) = P + 1$$

$$V(G) = 4 + 1$$

$$V(G) = 5$$

Dari perhitungan tersebut diperoleh nilai *cyclomatic complexity* sebesar 5. Nilai tersebut menunjukkan bahwa terdapat lima jalur independen dalam proses eksekusi fungsi lupa *password*.

f. Independent Path Fungsi Lupa Password

Berdasarkan hasil perhitungan *cyclomatic complexity*, diperoleh 5 jalur independen sebagai berikut:

1. *Path 1:*

$1 \rightarrow 2 \rightarrow 15$

Email kosong sehingga sistem menampilkan pesan *email* wajib diisi.

2. *Path 2:*

$1 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 14 \rightarrow 15$

Link reset password berhasil dikirim dan pengguna kembali ke halaman sebelumnya.

3. *Path 3:*

$1 \rightarrow 3 \rightarrow 4 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 13 \rightarrow 14 \rightarrow 15$

Email tidak terdaftar sehingga sistem menampilkan pesan *email* tidak terdaftar.

4. *Path 4:*

$1 \rightarrow 3 \rightarrow 4 \rightarrow 8 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15$

Format email tidak valid sehingga sistem menampilkan pesan *format email* tidak valid.

5. *Path 5:*

$1 \rightarrow 3 \rightarrow 4 \rightarrow 8 \rightarrow 13 \rightarrow 14 \rightarrow 15$

Terjadi *error firebase* lainnya sehingga sistem menampilkan pesan *login* gagal.

g. Test Case Fungsi Lupa Password

Pengujian fungsi lupa *password* dilakukan berdasarkan jalur independen (*independent path*) yang telah diperoleh dari hasil analisis *flowgraph* fungsi lupa *password*. Setiap jalur independen direpresentasikan ke dalam beberapa *test case* dengan kondisi *email* yang berbeda untuk menguji seluruh kemungkinan alur eksekusi program, mulai dari validasi *email* kosong, pengiriman *link reset password* berhasil, hingga penanganan *error firebase authentication*. Jalur eksekusi pada masing-masing *test case* diperoleh dari *node-node* yang dilalui pada *flowgraph* sehingga setiap proses pengujian dapat memverifikasi fungsi lupa *password* secara menyeluruh.

Tabel 5. 6 Test Case Fungsi Lupa Password

Test Case	Email	Jalur Eksekusi	Output
TC1	Kosong	1 → 2 → 15	Muncul pesan “Email wajib diisi”
TC2	Email valid terdftar	1 → 3 → 4 → 5 → 6 → 7 → 14 → 15	Link reset password berhasil dikirim
TC3	Email tidak terdftar	1 → 3 → 4 → 8 → 9 → 10 → 13 → 14 → 15	Muncul pesan “Email tidak terdftar”
TC4	Format email salah	1 → 3 → 4 → 8 → 11 → 12 → 13 → 14 → 15	Muncul pesan “Format email tidak valid”
TC5	Error Firebase lainnya	1 → 3 → 4 → 8 → 13 → 14 → 15	Muncul pesan “Gagal mengirim link reset password”

Sumber: data diolah oleh penulis, 2026

h. Statement Coverage Fungsi Lupa Password

Statement Coverage digunakan untuk memastikan seluruh *statement* pada fungsi lupa *password* telah dieksekusi minimal satu kali melalui *test case* yang dirancang. *Statement* yang tereksekusi pada masing-masing *test case* adalah sebagai berikut:

1. TC1:
S1, S2, S15
2. TC2:
S1, S3, S4, S5, S6, S7, S14, S15

3. TC3:
S1, S3, S4, S8, S9, S10, S13, S14, S15
4. TC4:
S1, S3, S4, S8, S11, S12, S13, S14, S15
5. TC5:
S1, S3, S4, S8, S13, S14, S15

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *statement* pada fungsi lupa *password* yaitu S1 hingga S15 berhasil dieksekusi melalui kombinasi *test case* TC1 sampai TC5.

$$(15 / 15) \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *statement* pada fungsi lupa *password* telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses *white box testing*.

5.4.4 Pengujian Fungsi Presensi Masuk dan Pulang

a. Potongan *Source Code* Fungsi Presensi Masuk dan Pulang

Fungsi presensi hadir digunakan untuk mencatat kehadiran karyawan melalui *qr code*, *selfie*, dan lokasi *gps*. Pada fungsi ini, pengguna dapat melakukan presensi masuk dan presensi pulang. Data presensi masuk akan disimpan ke *collection* presensi, sedangkan presensi pulang akan memperbarui data presensi yang sudah dibuat sebelumnya. Berikut adalah potongan *source code* fungsi presensi masuk dan pulang yang digunakan dalam pengujian:

```
Future<void> _simpan() async {
  if (isHadir) {
    await _simpanPresensiHadir();
  } else {
    await _simpanTidakMasuk();
  }
}

Future<void> _simpanPresensiHadir() async {
  if (!selfieDone || !gpsDone) {
```

```

ScaffoldMessenger.of(context).showSnackBar(
  const SnackBar(
    content: Text(
      'Lengkapi selfie dan pencatatan lokasi terlebih dahulu.',
    ),
  ),
);
return;
}
setState(() => isLoading = true);
try {
  final uid = FirebaseAuth.instance.currentUser!.uid;
  final now = DateTime.now();
  final tanggalString = DateFormat('yyyy-MM-dd').format(now);
  final jam = DateFormat('HH:mm').format(now);
  final docId = '${uid}_${tanggalString}';
  final userDoc =
    await FirebaseFirestore.instance.collection('users').doc(uid).get();
  final user = userDoc.data() as Map<String, dynamic>;
  final docRef =
    FirebaseFirestore.instance.collection('presensi').doc(docId);
  final presensiDoc = await docRef.get();
  if (jenisPresensi == 'masuk') {
    if (presensiDoc.exists &&
      presensiDoc.data()?['masuk'] != null &&
      presensiDoc.data()?['masuk'] != '-') {
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(
          content: Text('Anda sudah melakukan presensi masuk hari ini.'),
        ),
      );
    }
    return;
  }
}

```

```

}
selfieUrl = await _uploadSelfieCloudinary();
await docRef.set({
  'uid': uid,
  'nama': user['nama'],
  'nik': user['nik'],
  'tanggal': Timestamp.now(),
  'tanggal_string': tanggalString,
  'masuk': jam,
  'pulang': '-',
  'status': 'hadir',
  'approval_status': 'disetujui',
  'kode_qr': widget.qrData,
  'metode': 'QR Code + Selfie + GPS',
  'lokasi_masuk': lokasi,
  'latitude_masuk': latitude,
  'longitude_masuk': longitude,
  'foto_selfie_masuk': selfieUrl,
});
} else {
  if (!presensiDoc.exists) {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(
        content: Text('Anda belum melakukan presensi masuk hari ini.'),
      ),
    );
    return;
  }
  final dataPresensi = presensiDoc.data();
  if (dataPresensi?['status'] != 'hadir') {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(

```

```

        content: Text('Presensi pulang hanya untuk status hadir.'),
      ),
    );
    return;
  }
  if (dataPresensi?['pulang'] != null &&
    dataPresensi?['pulang'] != '-') {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(
        content: Text('Anda sudah melakukan presensi pulang hari ini.'),
      ),
    );
    return;
  }
  selfieUrl = await _uploadSelfieCloudinary();
  await docRef.update({
    'pulang': jam,
    'lokasi_pulang': lokasi,
    'latitude_pulang': latitude,
    'longitude_pulang': longitude,
    'foto_selfie_pulang': selfieUrl,
  });
}
if (!mounted) return;
Navigator.pushReplacement(
  context,
  MaterialPageRoute(
    builder: (_) => PresensiBerhasilPage(
      lokasi: lokasi,
      jenisPresensi: jenisPresensi,
    ),
  ),
);

```

```

);
} catch (e) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text('Gagal menyimpan presensi: $e')),
  );
} finally {
  if (mounted) setState(() => isLoading = false);
}
}
}

```

b. Identifikasi *Statement* Fungsi Presensi Masuk dan Pulang

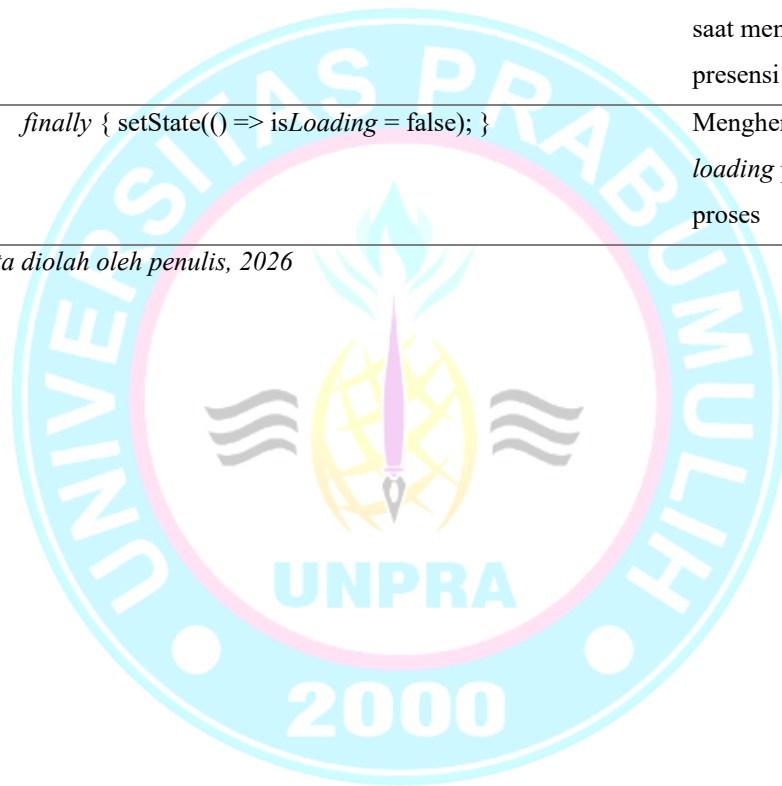
Tabel 5. 7 Identifikasi *Statement* Fungsi Presensi Masuk dan Pulang

<i>Statement</i>	<i>Source Code</i>	<i>Keterangan</i>
S1	if (isHadir)	Memeriksa apakah jenis presensi adalah masuk atau pulang
S2	_simpanPresensiHadir()	Menjalankan proses presensi hadir
S3	`if (!selfieDone	
S4	SnackBar('Lengkapi selfie...')	Menampilkan pesan jika selfie atau GPS belum lengkap
S5	setState(() => isLoading = true)	Mengaktifkan status loading
S6	<i>FirebaseAuth.instance.currentUser!.uid</i>	Mengambil UID pengguna yang sedang login
S7	<i>DateFormat('yyyy-MM-dd').format(now)</i>	Membentuk tanggal presensi
S8	<i>Firestore.instance.collection('users').doc(uid).get()</i>	Mengambil data karyawan dari collection users
S9	<i>Firestore.instance.collection('presensi').doc(docId)</i>	Membuat referensi dokumen presensi berdasarkan UID dan tanggal

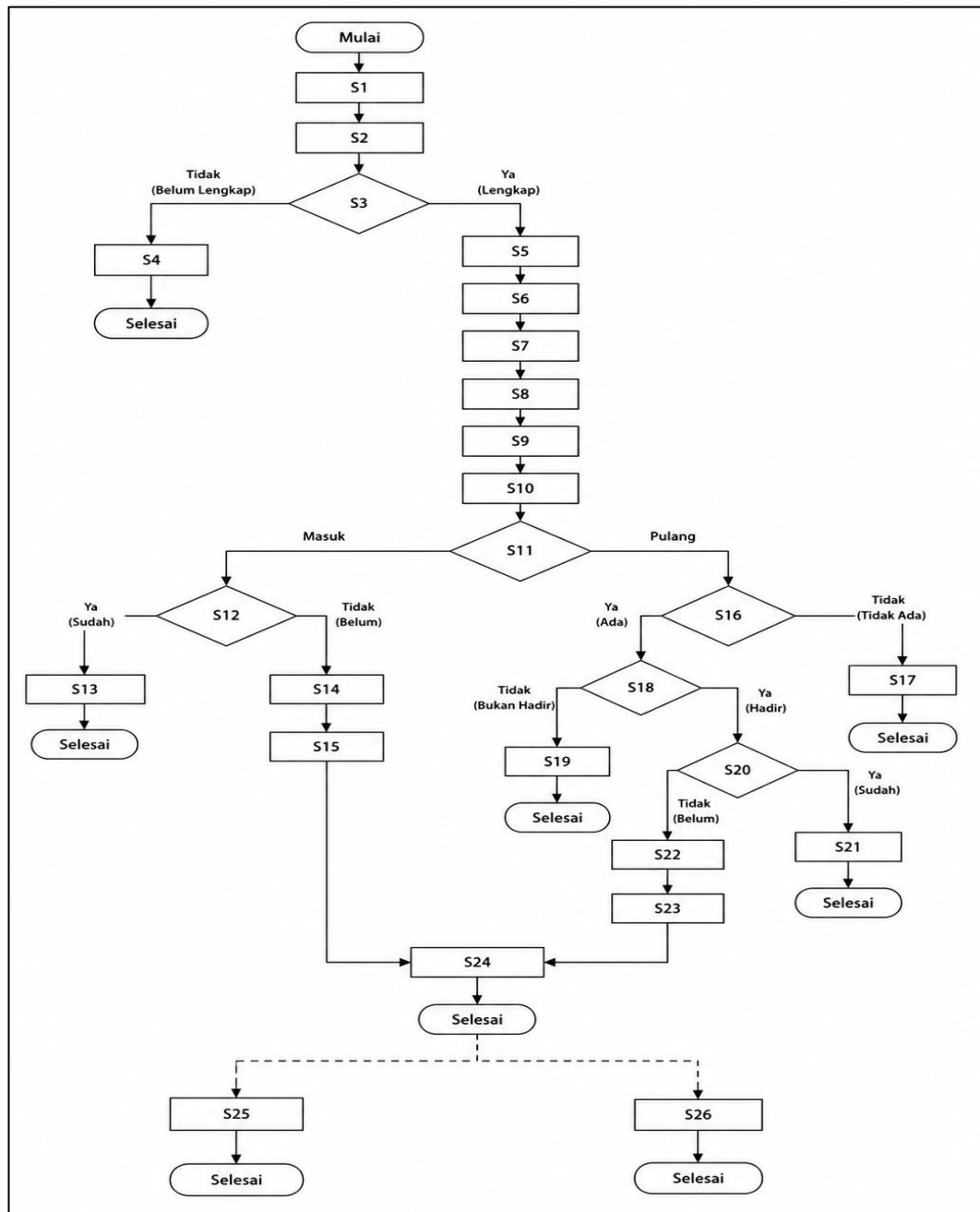
S10	<code>presensiDoc = await docRef.get()</code>	Mengambil data presensi hari ini
S11	<code>if (jenisPresensi == 'masuk')</code>	Memeriksa apakah jenis presensi adalah masuk
S12	<code>if (presensiDoc.exists && presensiDoc.data()['masuk'] != '-')</code>	Memeriksa apakah sudah presensi masuk hari ini
S13	<code>SnackBar('Anda sudah melakukan presensi masuk hari ini')</code>	Menampilkan pesan presensi masuk sudah dilakukan
S14	<code>_uploadSelfieCloudinary()</code>	Mengunggah foto <i>selfie</i> masuk ke Cloudinary
S15	<code>docRef.set({...})</code>	Menyimpan data presensi masuk ke <i>collection</i> presensi
S16	<code>if (!presensiDoc.exists)</code>	Memeriksa apakah data presensi masuk belum ada saat presensi pulang
S17	<code>SnackBar('Anda belum melakukan presensi masuk hari ini')</code>	Menampilkan pesan belum presensi masuk
S18	<code>if (dataPresensi?['status'] != 'hadir')</code>	Memeriksa apakah status presensi bukan hadir
S19	<code>SnackBar('Presensi pulang hanya untuk status hadir')</code>	Menampilkan pesan presensi pulang hanya untuk status hadir
S20	<code>if (dataPresensi?['pulang'] != null && dataPresensi?['pulang'] != '-')</code>	Memeriksa apakah sudah presensi pulang hari ini
S21	<code>SnackBar('Anda sudah melakukan presensi pulang hari ini')</code>	Menampilkan pesan presensi pulang sudah dilakukan

S22	<code>_uploadSelfieCloudinary()</code>	Mengunggah foto <i>selfie</i> pulang ke Cloudinary
S23	<code>docRef.update({...})</code>	Memperbarui data presensi pulang di <i>collection</i> presensi
S24	<code>Navigator.pushReplacement(...)</code>	Mengarahkan pengguna ke halaman presensi berhasil
S25	<code>catch (e)</code>	Menangani <i>error</i> saat menyimpan presensi
S26	<code>finally { setState(() => isLoading = false); }</code>	Menghentikan <i>loading</i> pada akhir proses

Sumber: data diolah oleh penulis, 2026



c. *Flowchart fungsi Presensi Masuk dan Pulang*



Sumber: data diolah oleh penulis, 2026

Gambar 5. 41 *Flowchart Fungsi Presensi Masuk dan Pulang*

Berdasarkan diagram alir pada gambar *flowchart* fungsi presensi masuk dan pulang, urutan logika kerja fungsi presensi masuk dan pulang dapat dijabarkan sebagai berikut:

1. S1 dan S2 (Pemilihan Presensi Masuk dan Pulang)

Sistem memeriksa apakah jenis presensi yang dipilih pengguna termasuk presensi hadir, yaitu presensi masuk atau presensi pulang. Jika termasuk presensi hadir, sistem menjalankan proses presensi hadir.

2. S3 dan S4 (Validasi *Selfie* dan *GPS*)

Sistem memeriksa apakah *selfie* dan pencatatan lokasi *gps* sudah lengkap. Jika belum lengkap, sistem menampilkan pesan agar pengguna melengkapi *selfie* dan lokasi terlebih dahulu, kemudian proses dihentikan.

3. S5 sampai S10 (Persiapan Data Presensi)

Jika *selfie* dan *gps* sudah lengkap, sistem mengaktifkan *loading*, mengambil uid pengguna, membentuk tanggal presensi, mengambil data *user* dari *firestore*, membuat referensi dokumen presensi, dan mengambil data presensi hari ini.

4. S11 sampai S15 (Presensi Masuk)

Sistem memeriksa apakah jenis presensi adalah masuk. Jika pengguna sudah melakukan presensi masuk pada hari tersebut, sistem menampilkan pesan bahwa presensi masuk sudah dilakukan. Jika belum, sistem mengunggah *selfie* ke *cloudinary*, lalu menyimpan data presensi masuk ke *firestore*.

5. S16 sampai S23 (Presensi Pulang)

Jika jenis presensi adalah pulang, sistem memeriksa apakah data presensi masuk sudah ada. Jika belum ada, sistem menampilkan pesan bahwa pengguna belum melakukan presensi masuk. Jika data ada, sistem memeriksa apakah status presensi adalah hadir. Jika bukan hadir, sistem menampilkan pesan bahwa presensi pulang hanya untuk status hadir. Jika status valid, sistem memeriksa apakah pengguna sudah presensi pulang. Jika sudah, sistem menampilkan pesan bahwa presensi pulang sudah dilakukan. Jika belum, sistem mengunggah *selfie* pulang ke *cloudinary* dan memperbarui data presensi pulang di *firestore*.

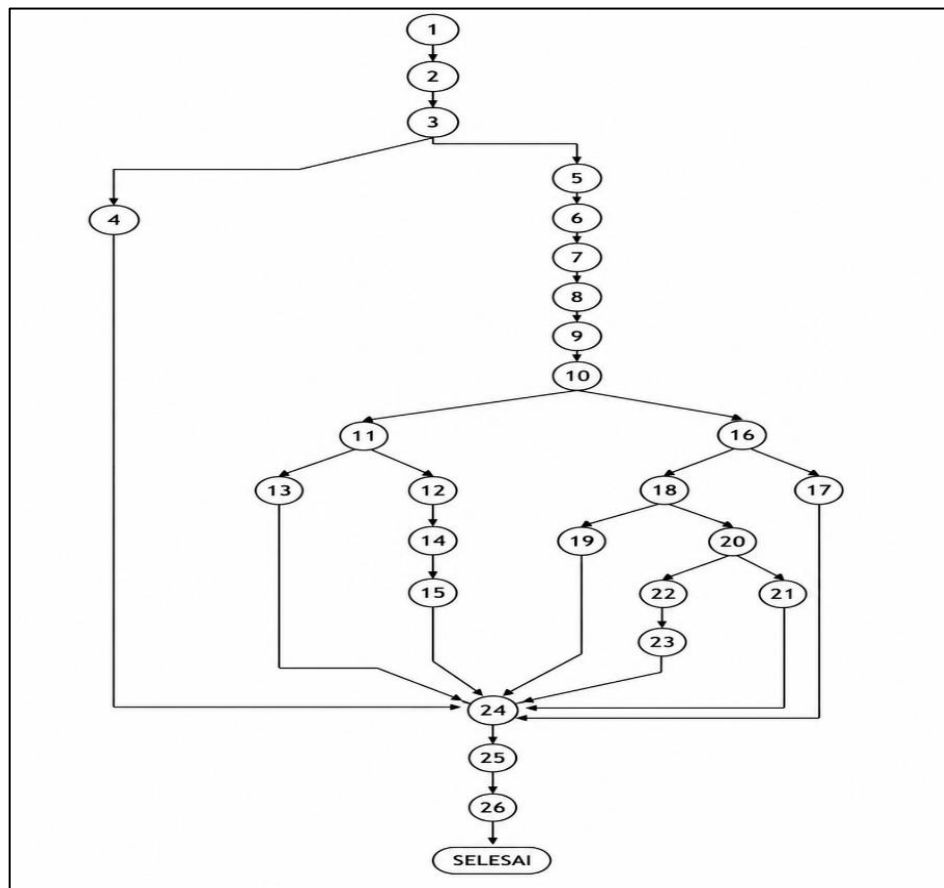
6. S24 (Presensi Berhasil)

Jika presensi masuk atau presensi pulang berhasil disimpan, sistem menampilkan halaman presensi berhasil.

7. S25 dan S26 (*Error* dan Akhir Proses)

Jika terjadi kesalahan saat proses presensi, sistem menjalankan bagian *catch* untuk menampilkan pesan *error*. Pada akhir proses, sistem menjalankan bagian *finally* untuk menghentikan *loading*.

d. *Flowgraph* Fungsi Presensi Masuk dan Pulang



Sumber: data diolah oleh penulis 2026

Gambar 5. 42 *Flowgraph* Fungsi Presensi Masuk dan Pulang

Berdasarkan gambar *flowgraph* fungsi presensi masuk dan pulang, alur logika program fungsi presensi masuk dan pulang dapat dijelaskan sebagai berikut. Proses dimulai pada *node* 1 dan *node* 2, yaitu sistem memeriksa jenis presensi yang dipilih pengguna serta menjalankan proses presensi masuk dan pulang. Selanjutnya pada *node* 3, sistem melakukan pengecekan apakah *selfie* dan lokasi *gps* telah lengkap. Jika *selfie* atau *gps* belum lengkap, maka alur menuju *node* 4 untuk menampilkan pesan bahwa *selfie* dan *gps* harus dilengkapi terlebih dahulu, kemudian proses berakhir.

Jika *selfie* dan *gps* telah lengkap, sistem melanjutkan ke *node* 5 untuk mengaktifkan *loading*. Setelah itu, *node* 6 digunakan untuk mengambil uid pengguna yang sedang *login*. Pada *node* 7 sistem membentuk *format* tanggal presensi, kemudian *node* 8 digunakan untuk mengambil data pengguna dari *firebase firestore*. Selanjutnya, *node* 9 membuat referensi dokumen presensi dan *node* 10 mengambil data presensi pengguna pada hari tersebut.

Pada *node* 10, alur bercabang menjadi presensi masuk dan presensi pulang. Jika pengguna melakukan presensi masuk, sistem menuju *node* 11 untuk memeriksa apakah pengguna sudah melakukan presensi masuk sebelumnya. Jika sudah melakukan presensi masuk, maka sistem menuju *node* 13 untuk menampilkan pesan bahwa presensi masuk telah dilakukan, kemudian proses selesai. Jika belum melakukan presensi masuk, maka alur menuju *node* 12 dan *node* 14 untuk melakukan *upload selfie* ke *cloudinary*. Setelah *upload* berhasil, sistem menjalankan *node* 15 untuk menyimpan data presensi masuk ke *firebase firestore*.

Jika pengguna melakukan presensi pulang, sistem menuju *node* 16 untuk memeriksa apakah data presensi masuk sudah tersedia. Jika data presensi masuk belum ada, maka sistem menuju *node* 17 untuk menampilkan pesan bahwa pengguna belum melakukan presensi masuk, kemudian proses selesai. Jika data presensi masuk tersedia, sistem melanjutkan ke *node* 18 untuk memeriksa status presensi. Jika status bukan “hadir”, maka sistem menampilkan pesan bahwa presensi pulang hanya dapat dilakukan pada status hadir.

Namun jika status valid, sistem menuju *node* 20 untuk memeriksa apakah pengguna sudah melakukan presensi pulang. Jika sudah melakukan presensi pulang, maka sistem menuju *node* 21 untuk menampilkan pesan bahwa presensi pulang telah dilakukan sebelumnya. Jika belum melakukan presensi pulang, sistem menjalankan *node* 22 untuk mengunggah *selfie* pulang ke *cloudinary*, kemudian *node* 23 untuk memperbarui data presensi pulang pada *firebase firestore*.

Setelah proses presensi masuk maupun presensi pulang berhasil dilakukan, alur menuju *node* 24 untuk menampilkan halaman presensi berhasil. Jika selama proses terjadi kesalahan, maka sistem menjalankan *node* 25 untuk menangani *error* dan menampilkan pesan kesalahan kepada pengguna. Pada akhir proses, sistem

menjalankan *node* 26 untuk menghentikan *loading* melalui blok *finally*, kemudian proses berakhir pada terminator selesai.

e. *Cyclomatic Complexity* Fungsi Presensi Masuk dan Pulang

Berdasarkan *flowgraph* yang telah dibentuk, terdapat 7 *predicate node* (P) yang menjadi titik keputusan pada fungsi presensi masuk dan pulang *predicate node* tersebut yaitu *node* 3 yang melakukan pengecekan kelengkapan *selfie* dan *gps*, *Node* 10 yang menentukan jenis presensi masuk atau pulang, *node* 11 yang memeriksa apakah pengguna sudah melakukan presensi masuk, *node* 16 yang memeriksa apakah data presensi masuk sudah tersedia saat presensi pulang, *node* 18 yang memeriksa status presensi, *node* 20 yang memeriksa apakah pengguna sudah melakukan presensi pulang, serta *node* 25 yang menangani *error* pada proses presensi. Oleh karena itu, nilai *cyclomatic complexity* dapat dihitung menggunakan rumus berikut:

$$V(G) = P + 1$$

$$V(G) = 7 + 1$$

$$V(G) = 8$$

Dari perhitungan tersebut diperoleh nilai *cyclomatic complexity* sebesar 8. Nilai tersebut menunjukkan bahwa terdapat delapan jalur independen dalam proses eksekusi fungsi presensi hadir.

f. *Independent Path* Fungsi Presensi Masuk dan Pulang

Berdasarkan hasil perhitungan *cyclomatic complexity* fungsi presensi masuk dan pulang, diperoleh nilai $V(G) = 8$. Nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses *white box testing*. Sehingga didapatkan 8 jalur independen sebagai berikut:

1. *Path* 1:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 26$

Selfie atau *gps* belum lengkap.

2. *Path* 2:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 13 \rightarrow 26$

Pengguna sudah melakukan presensi masuk.

3. *Path 3:*

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 14 \rightarrow 15 \rightarrow 24 \rightarrow 26$

Presensi masuk berhasil disimpan.

4. *Path 4:*

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 16 \rightarrow 17 \rightarrow 26$

Presensi pulang gagal karena belum melakukan presensi masuk.

5. *Path 5:*

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 16 \rightarrow 18 \rightarrow 19 \rightarrow 26$

Presensi pulang ditolak karena status bukan hadir.

6. *Path 6:*

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 16 \rightarrow 18 \rightarrow 20 \rightarrow 21 \rightarrow 26$

Pengguna sudah melakukan presensi pulang.

7. *Path 7:*

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 16 \rightarrow 18 \rightarrow 20 \rightarrow 22 \rightarrow 23 \rightarrow 24 \rightarrow 26$

Presensi pulang berhasil diperbarui.

8. *Path 8:*

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 25 \rightarrow 26$

Terjadi *error* saat proses presensi berlangsung.

g. *Test Case Fungsi Presensi Masuk dan Pulang*

Pengujian fungsi presensi masuk dan pulang dilakukan berdasarkan jalur independen yang telah diperoleh dari hasil analisis *flowgraph*. Setiap *test case* digunakan untuk menguji kemungkinan alur pada proses presensi masuk dan presensi pulang.

Tabel 5. 8 *Test Case* Fungsi Presensi Masuk dan Pulang

<i>Test Case</i>	Kondisi	Jalur Eksekusi	Output
TC1	<i>Selfie</i> atau <i>GPS</i> belum lengkap	1 → 2 → 3 → 4 → 26	Muncul pesan lengkapi <i>selfie</i> dan lokasi terlebih dahulu
TC2	Sudah presensi masuk	1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 13 → 26	Muncul pesan sudah melakukan presensi masuk hari ini
TC3	Presensi masuk valid	1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12 → 14 → 15 → 24 → 26	Data presensi masuk berhasil disimpan
TC4	Presensi pulang tanpa presensi masuk	1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 10 → 16 → 17 → 26	Muncul pesan belum melakukan presensi masuk hari ini
TC5	Presensi pulang dengan status bukan hadir	1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 10 → 16 → 18 → 19 → 26	Muncul pesan presensi pulang hanya untuk status hadir
TC6	Sudah presensi pulang	1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 10 → 16 → 18 → 20 → 21 → 26	Muncul pesan sudah melakukan presensi pulang hari ini
TC7	Presensi pulang valid	1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 10 → 16 → 18 → 20 → 22 → 23 → 24 → 26	Data presensi pulang berhasil diperbarui
TC8	Terjadi <i>error</i> sistem	1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 25 → 26	Muncul pesan gagal menyimpan presensi

Sumber: data diolah oleh penulis 2026

h. *Statement Coverage* Fungsi Presensi Masuk dan Pulang

Statement coverage digunakan untuk mengukur sejauh mana seluruh *statement* pada fungsi presensi masuk dan pulang telah dieksekusi melalui *test case* yang telah dirancang. Pengujian ini bertujuan untuk memastikan setiap *statement* pada proses presensi masuk dan presensi pulang telah dijalankan minimal satu kali. *Statement* yang tereksekusi pada masing-masing *test case* adalah sebagai berikut:

1. TC1:
S1, S2, S3, S4, S26

2. TC2:
S1, S2, S3, S5, S6, S7, S8, S9, S10, S11, S13, S26
3. TC3:
S1, S2, S3, S5, S6, S7, S8, S9, S10, S11, S12, S14, S15, S24, S26
4. TC4:
S1, S2, S3, S5, S6, S7, S8, S9, S10, S16, S17, S26
5. TC5:
S1, S2, S3, S5, S6, S7, S8, S9, S10, S16, S18, S19, S26
6. TC6:
S1, S2, S3, S5, S6, S7, S8, S9, S10, S16, S18, S20, S21, S26
7. TC7:
S1, S2, S3, S5, S6, S7, S8, S9, S10, S16, S18, S20, S22, S23, S24, S26
8. TC8:
S1, S2, S3, S5, S6, S7, S8, S9, S25, S26

Berdasarkan hasil pengujian, seluruh *statement* pada fungsi presensi masuk dan pulang yaitu S1 sampai S26 berhasil dieksekusi melalui kombinasi *test case* TC1 sampai TC8. Dengan demikian diperoleh nilai *statement coverage* sebesar:

$$(26 / 26) \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *statement* pada fungsi presensi hadir telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses *white box testing*.

5.4.5 Pengujian Fungsi Pengajuan Presensi

a. Potongan *Source Code* Fungsi Pengajuan Presensi

Fungsi pengajuan presensi digunakan untuk memproses pengajuan tidak masuk kerja, yaitu sakit, izin, dan cuti. Pada fungsi ini, sistem melakukan validasi keterangan, validasi tanggal cuti, pengecekan batas maksimal tidak masuk sebanyak 12 kali dalam satu tahun, serta pengecekan apakah data presensi atau pengajuan pada tanggal yang sama sudah ada. Jika seluruh validasi terpenuhi, maka data pengajuan disimpan ke *collection* pengajuan dengan status menunggu. Berikut adalah potongan *source code* fungsi pengajuan presensi yang digunakan dalam pengujian:

```

Future<void> _simpanTidakMasuk() async {
  if (keteranganC.text.trim().isEmpty) {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('Keterangan wajib diisi')),
    );
    return;
  }
  if (isCuti && (tanggalCutiAwal == null || tanggalCutiAkhir == null)) {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(
        content: Text('Tanggal awal dan akhir cuti wajib diisi'),
      ),
    );
    return;
  }
  if (isCuti && tanggalCutiAkhir!.isBefore(tanggalCutiAwal!)) {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(
        content: Text('Tanggal akhir cuti tidak boleh lebih awal'),
      ),
    );
    return;
  }
  setState(() => isLoading = true);
  try {
    final uid = FirebaseAuth.instance.currentUser!.uid;
    final userDoc =
      await FirebaseFirestore.instance.collection('users').doc(uid).get();
    final user = userDoc.data() as Map<String, dynamic>;
    final jumlahTidakMasuk = await _hitungTidakMasukTahunIni(uid);
    int jumlahPengajuanBaru = 1;
  }
}

```

```

if (isCuti) {
    jumlahPengajuanBaru =
        tanggalCutiAkhir!.difference(tanggalCutiAwal!).inDays + 1;
}
if (jumlahTidakMasuk + jumlahPengajuanBaru > 12) {
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
            content: Text(
                'Batas tidak masuk 12 kali dalam 1 tahun sudah tercapai. Saat ini:
$jumlahTidakMasuk kali.',
            ),
        ),
    );
    return;
}
final tanggalString = _tanggalString(tanggalTidakMasuk);
final docId = '${uid}_${tanggalString}_${jenisPresensi}';
final presensiDoc = await FirebaseFirestore.instance
    .collection('presensi')
    .doc('${uid}_${tanggalString}')
    .get();
final pengajuanDoc = await FirebaseFirestore.instance
    .collection('pengajuan')
    .doc(docId)
    .get();
if (presensiDoc.exists || pengajuanDoc.exists) {
    ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(
            content: Text(
                'Data presensi atau pengajuan pada tanggal ini sudah ada',
            ),
        ),
    );
}

```

```

    );
    return;
}
await FirebaseFirestore.instance.collection('pengajuan').doc(docId).set({
    'uid': uid,
    'nama': user['nama'],
    'nik': user['nik'],
    'tanggal': Timestamp.fromDate(tanggalTidakMasuk),
    'tanggal_string': tanggalString,
    'jenis_pengajuan': jenisPresensi,
    'approval_status': 'menunggu',
    'keterangan': keteranganC.text.trim(),
    'dibuat_oleh': uid,
    'tanggal_pengajuan': Timestamp.now(),
});
if (!mounted) return;

ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
        content: Text(
            'Pengajuan ${jenisPresensi.toUpperCase()} berhasil dikirim',
        ),
    ),
);

Navigator.pop(context);
} catch (e) {
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Gagal menyimpan pengajuan: $e')),
    );
} finally {

```

```

    if (mounted) setState(() => isLoading = false);
  }
}

```

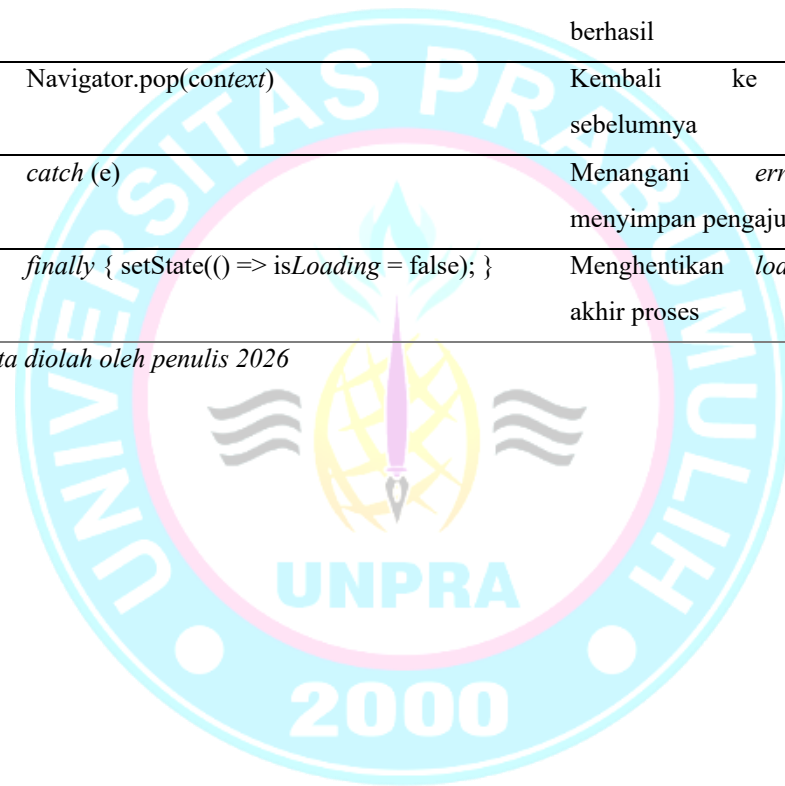
b. Identifikasi *Statement* Fungsi Pengajuan Presensi

Tabel 5. 9 Identifikasi *Statement* Fungsi Pengajuan Presensi

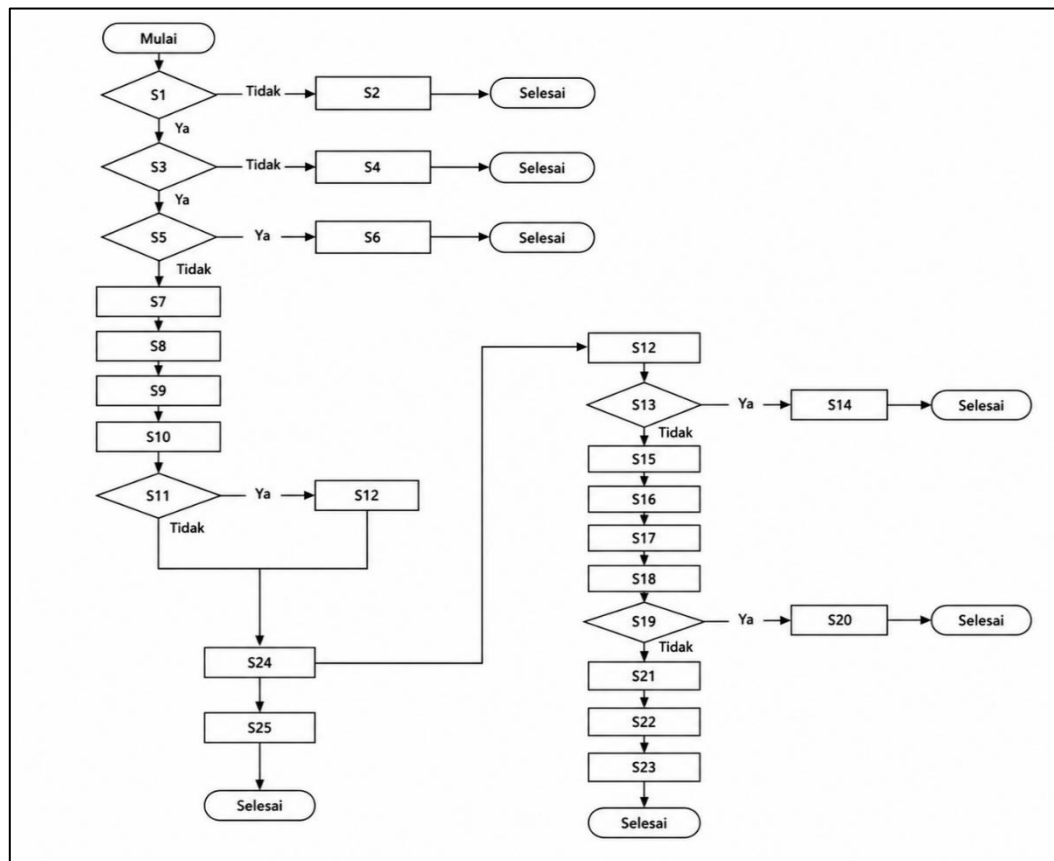
<i>Statement</i>	<i>Source Code</i>	<i>Keterangan</i>
S1	<code>if (keteranganC.text.trim().isEmpty())</code>	Memeriksa apakah keterangan pengajuan kosong
S2	<code>SnackBar('Keterangan wajib diisi')</code>	Menampilkan pesan keterangan wajib diisi
S3	<code>if (isCuti && tanggalCutiAwal == null...)</code>	Memeriksa tanggal awal dan akhir cuti
S4	<code>SnackBar('Tanggal awal dan akhir cuti wajib diisi')</code>	Menampilkan pesan tanggal cuti wajib diisi
S5	<code>if (isCuti && tanggalCutiAkhir!.isBefore...)</code>	Memeriksa apakah tanggal akhir cuti lebih awal
S6	<code>SnackBar('Tanggal akhir cuti tidak boleh lebih awal')</code>	Menampilkan pesan tanggal akhir cuti tidak valid
S7	<code>setState(() => isLoading = true)</code>	Mengaktifkan <i>loading</i>
S8	<code>FirebaseAuth.instance.currentUser!.uid</code>	Mengambil UID karyawan yang <i>login</i>
S9	<code>collection('users').doc(uid).get()</code>	Mengambil data karyawan dari <i>Firestore</i>
S10	<code>_hitungTidakMasukTahunIni(uid)</code>	Menghitung jumlah tidak masuk dalam satu tahun
S11	<code>if (isCuti)</code>	Memeriksa apakah pengajuan adalah cuti
S12	<code>jumlahPengajuanBaru = ...inDays + 1</code>	Menghitung jumlah hari cuti
S13	<code>if (jumlahTidakMasuk + jumlahPengajuanBaru > 12)</code>	Memeriksa batas tidak masuk maksimal 12 kali
S14	<code>SnackBar('Batas tidak masuk 12 kali...')</code>	Menampilkan pesan batas tidak masuk tercapai
S15	<code>final tanggalString = _tanggalString(...)</code>	Membentuk <i>format</i> tanggal pengajuan
S16	<code>final docId = '\$ {uid}_ \$ {tanggalString}_ \$ jenisPresensi'</code>	Membentuk ID dokumen pengajuan

S17	<code>collection('presensi').doc(...).get()</code>	Mengecek data presensi pada tanggal tersebut
S18	<code>collection('pengajuan').doc(docId).get()</code>	Mengecek data pengajuan pada tanggal tersebut
S19	<code>`if (presensiDoc.exists</code>	Mengecek jika data presensi tidak ada
S20	<code>SnackBar('Data presensi atau pengajuan...')</code>	Menampilkan pesan data sudah ada
S21	<code>collection('pengajuan').doc(docId).set({...})</code>	Menyimpan data pengajuan ke <i>collection</i> pengajuan
S22	<code>SnackBar('Pengajuan berhasil dikirim')</code>	Menampilkan pesan pengajuan berhasil
S23	<code>Navigator.pop(context)</code>	Kembali ke halaman sebelumnya
S24	<code>catch (e)</code>	Menangani <i>error</i> saat menyimpan pengajuan
S25	<code>finally { setState(() => isLoading = false); }</code>	Menghentikan <i>loading</i> pada akhir proses

Sumber: data diolah oleh penulis 2026



c. Flowchart Fungsi Pengajuan Presensi



Sumber : data diolah oleh penulis 2026

Gambar 5. 43 Flowchart Fungsi Pengajuan Presensi

Berdasarkan diagram alir pada gambar *flowchart* fungsi pengajuan presensi, urutan logika kerja fungsi pengajuan presensi dapat dijabarkan sebagai berikut:

1. S1 dan S2 (Validasi Keterangan Pengajuan)

Sistem memulai proses pengajuan presensi dengan memeriksa apakah keterangan pengajuan sudah diisi. Jika keterangan masih kosong, maka sistem menjalankan S2 untuk menampilkan pesan “keterangan wajib diisi”, kemudian proses dihentikan.

2. S3 dan S4 (Validasi Tanggal Cuti)

Jika keterangan sudah diisi, sistem memeriksa apakah jenis pengajuan adalah cuti dan apakah tanggal awal serta tanggal akhir cuti sudah dipilih. Jika tanggal cuti belum lengkap, maka sistem menjalankan S4 untuk menampilkan pesan bahwa tanggal awal dan akhir cuti wajib diisi.

3. S5 dan S6 (Validasi Urutan Tanggal Cuti)

Sistem memeriksa apakah tanggal akhir cuti lebih awal dari tanggal awal cuti. Jika kondisi tersebut terjadi, maka sistem menjalankan S6 untuk menampilkan pesan bahwa tanggal akhir cuti tidak boleh lebih awal.

4. S7 sampai S10 (Pengambilan Data dan Perhitungan Tidak Masuk)

Jika validasi awal terpenuhi, sistem mengaktifkan *loading*, mengambil uid karyawan yang sedang *login*, mengambil data karyawan dari *firestore*, kemudian menghitung jumlah tidak masuk karyawan dalam satu tahun.

5. S11 dan S12 (Perhitungan Jumlah Hari Cuti)

Sistem memeriksa apakah pengajuan yang dilakukan adalah cuti. Jika iya, sistem menghitung jumlah hari cuti berdasarkan rentang tanggal awal dan tanggal akhir cuti.

6. S13 dan S14 (Validasi Batas Tidak Masuk)

Sistem memeriksa apakah jumlah tidak masuk ditambah pengajuan baru melebihi batas maksimal 12 kali dalam satu tahun. Jika melebihi batas, maka sistem menjalankan S14 untuk menampilkan pesan bahwa batas tidak masuk sudah tercapai.

7. S15 sampai S18 (Pengecekan Data Presensi dan Pengajuan)

Jika batas tidak masuk masih memenuhi aturan, sistem membentuk tanggal pengajuan dan id dokumen pengajuan. Selanjutnya, sistem mengecek apakah pada tanggal tersebut sudah ada data presensi atau data pengajuan sebelumnya.

8. S19 dan S20 (Data Sudah Ada)

Jika data presensi atau pengajuan pada tanggal tersebut sudah ada, maka sistem menjalankan S20 untuk menampilkan pesan bahwa data presensi atau pengajuan pada tanggal tersebut sudah tersedia.

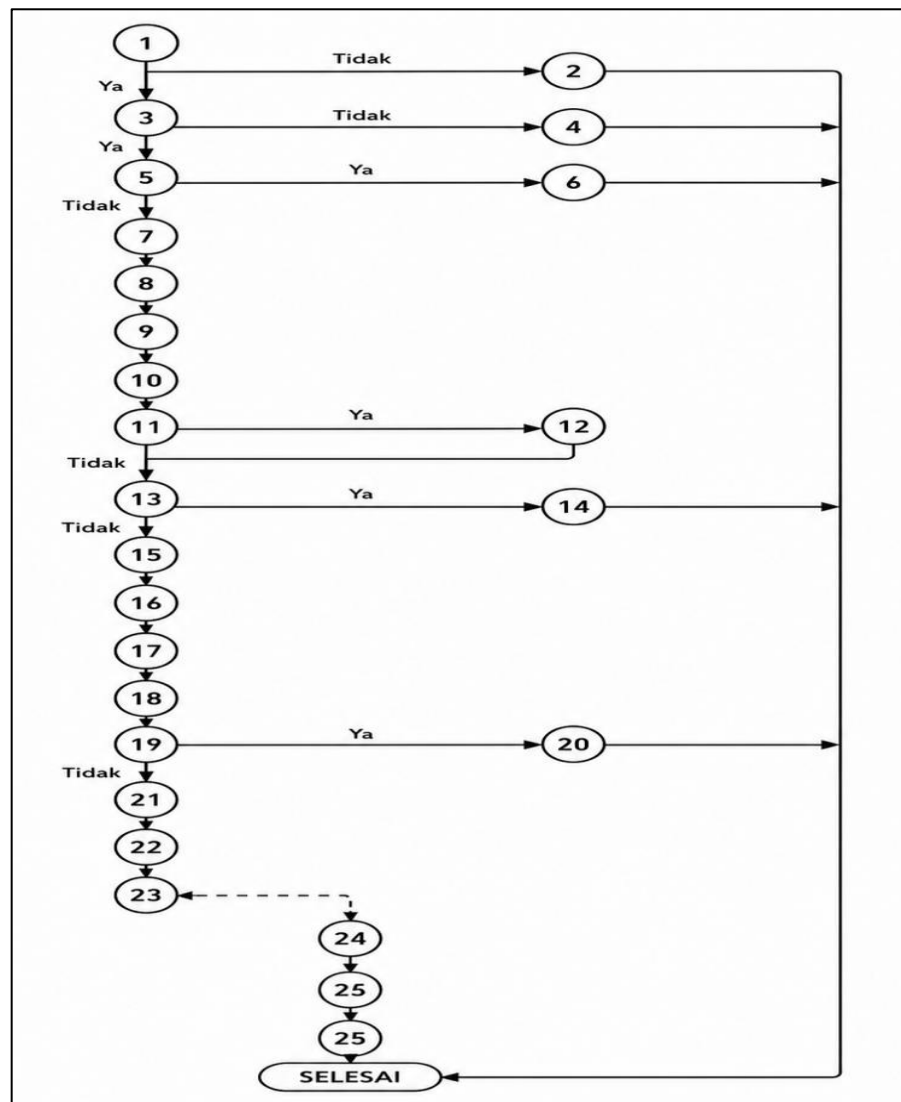
9. S21 sampai S23 (Penyimpanan Pengajuan Berhasil)

Jika seluruh validasi berhasil, sistem menjalankan S21 untuk menyimpan data pengajuan ke *collection* pengajuan dengan status menunggu. Setelah itu, sistem menjalankan S22 untuk menampilkan pesan bahwa pengajuan berhasil dikirim, kemudian S23 untuk kembali ke halaman sebelumnya.

10. S24 dan S25 (Penanganan *Error* dan Akhir Proses)

Jika terjadi kesalahan saat proses penyimpanan pengajuan, sistem menjalankan S24 untuk menangani *error* dan menampilkan pesan kesalahan. Pada akhir proses, sistem menjalankan S25 untuk menghentikan *loading* melalui blok *finally*.

d. Flowgraph Fungsi Pengajuan Presensi



Sumber : data diolah oleh penulis 2026

Gambar 5. 44 Flowgraph Fungsi Pengajuan Presensi

Berdasarkan gambar *flowgraph* fungsi pengajuan presensi, alur logika program fungsi pengajuan presensi dapat dijelaskan sebagai berikut. Proses dimulai pada *node* 1, yaitu sistem memeriksa apakah keterangan pengajuan sudah diisi. Jika

keterangan masih kosong, maka alur menuju *node* 2 untuk menampilkan pesan bahwa keterangan wajib diisi, kemudian proses berakhir. Jika keterangan telah diisi, sistem melanjutkan ke *node* 3 untuk memeriksa apakah jenis pengajuan adalah cuti dan apakah tanggal awal serta tanggal akhir cuti sudah dipilih. Jika tanggal cuti belum lengkap, maka alur menuju *node* 4 untuk menampilkan pesan bahwa tanggal awal dan akhir cuti wajib diisi, kemudian proses selesai.

Jika tanggal cuti telah lengkap, sistem melanjutkan ke *node* 5 untuk memeriksa apakah tanggal akhir cuti lebih awal dari tanggal awal cuti. Jika kondisi tersebut terjadi, maka alur menuju *node* 6 untuk menampilkan pesan bahwa tanggal akhir cuti tidak boleh lebih awal dari tanggal awal cuti, kemudian proses selesai. Jika validasi tanggal berhasil, sistem menjalankan *node* 7 untuk mengaktifkan *loading*. Selanjutnya *node* 8 digunakan untuk mengambil uid karyawan yang sedang *login*, kemudian *node* 9 mengambil data karyawan dari *collection users* berdasarkan uid pengguna. Setelah itu, *node* 10 digunakan untuk menghitung jumlah tidak masuk karyawan dalam satu tahun.

Pada *node* 11, sistem memeriksa apakah jenis pengajuan adalah cuti. Jika pengajuan berupa cuti, maka sistem menjalankan *node* 12 untuk menghitung jumlah hari cuti berdasarkan selisih tanggal awal dan tanggal akhir cuti. Jika pengajuan bukan cuti, sistem langsung melanjutkan proses berikutnya. *node* 13 digunakan untuk memeriksa apakah jumlah tidak masuk ditambah jumlah pengajuan baru melebihi batas maksimal 12 kali dalam satu tahun. Jika melebihi batas, maka alur menuju *node* 14 untuk menampilkan pesan bahwa batas tidak masuk telah tercapai, kemudian proses selesai.

Jika batas tidak masuk belum tercapai, sistem melanjutkan ke *node* 15 untuk membentuk *format* tanggal pengajuan. Selanjutnya *node* 16 membentuk id dokumen pengajuan berdasarkan uid, tanggal, dan jenis pengajuan. Setelah itu, *node* 17 digunakan untuk mengecek data presensi pada tanggal tersebut di *collection presensi*, dan *node* 18 digunakan untuk mengecek data pengajuan pada tanggal yang sama di *collection pengajuan*. Pada *node* 19, sistem memeriksa apakah data presensi atau data pengajuan pada tanggal tersebut sudah tersedia. Jika data sudah ada, maka alur menuju *node* 20 untuk menampilkan pesan bahwa data presensi atau pengajuan pada tanggal tersebut sudah ada, kemudian proses selesai.

Jika data belum ada, sistem menjalankan *node* 21 untuk menyimpan data pengajuan ke *collection* pengajuan dengan status menunggu. Setelah data berhasil disimpan, sistem menjalankan *node* 22 untuk menampilkan pesan bahwa pengajuan berhasil dikirim. Selanjutnya *node* 23 digunakan untuk kembali ke halaman sebelumnya menggunakan *navigator.pop ()*. Jika terjadi kesalahan saat proses penyimpanan data, maka sistem menjalankan *node* 24 untuk menangani *error* dan menampilkan pesan kesalahan kepada pengguna. Pada akhir proses, sistem menjalankan *node* 25 untuk menghentikan *loading* melalui blok *finally*, kemudian proses berakhir.

e. Cyclomatic Complexity Fungsi Pengajuan Presensi

Berdasarkan *flowgraph* yang telah dibentuk, terdapat 6 *predicate node* (P) pada fungsi pengajuan presensi. *Predicate node* tersebut terdiri dari *node* 1 yang memeriksa apakah keterangan pengajuan kosong, *node* 3 yang memeriksa kelengkapan tanggal cuti, *node* 5 yang memeriksa validitas urutan tanggal cuti, *node* 11 yang memeriksa apakah jenis pengajuan adalah cuti, *node* 13 yang memeriksa batas maksimal tidak masuk dalam satu tahun, serta *node* 19 yang memeriksa apakah data presensi atau pengajuan pada tanggal tersebut sudah ada. Oleh karena itu, nilai *cyclomatic complexity* dapat dihitung menggunakan rumus berikut:

$$V(G) = P + 1$$

$$V(G) = 6 + 1$$

$$V(G) = 7$$

Dari hasil perhitungan tersebut diperoleh nilai *cyclomatic complexity* sebesar 7. Nilai tersebut menunjukkan bahwa terdapat tujuh jalur independen dalam proses eksekusi fungsi pengajuan presensi.

f. Independent Path Fungsi Pengajuan Presensi

Berdasarkan hasil perhitungan *cyclomatic complexity* fungsi pengajuan presensi, diperoleh nilai $V(G) = 7$. Nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses *white box testing*. Sehingga didapatkan 7 jalur independen sebagai berikut:

1. *Path 1:*

$$1 \rightarrow 2 \rightarrow 25$$

Keterangan pengajuan kosong.

2. *Path 2:*

$$1 \rightarrow 3 \rightarrow 4 \rightarrow 25$$

Pengajuan cuti tetapi tanggal awal atau tanggal akhir belum diisi.

3. *Path 3:*

$$1 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 25$$

Tanggal akhir cuti lebih awal dari tanggal awal cuti.

4. *Path 4:*

$$1 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 25$$

Jumlah tidak masuk melebihi batas maksimal 12 kali dalam satu tahun.

5. *Path 5:*

$$1 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 13 \rightarrow 15 \rightarrow 16 \rightarrow 17 \rightarrow 18 \rightarrow 19 \rightarrow 20 \rightarrow 25$$

Data presensi atau pengajuan pada tanggal tersebut sudah ada.

6. *Path 6:*

$$1 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 13 \rightarrow 15 \rightarrow 16 \rightarrow 17 \rightarrow 18 \rightarrow 19 \rightarrow 21 \rightarrow 22 \rightarrow 23 \rightarrow 25$$

Pengajuan sakit atau izin berhasil dikirim.

7. *Path 7:*

$$1 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 15 \rightarrow 16 \rightarrow 17 \rightarrow 18 \rightarrow 19 \rightarrow 21 \rightarrow 22 \rightarrow 23 \rightarrow 25$$

Pengajuan cuti berhasil dikirim.

g. *Test Case* Fungsi Pengajuan Presensi

Pengujian fungsi pengajuan presensi dilakukan berdasarkan jalur independen yang telah diperoleh dari hasil analisis *flowgraph*. Setiap *test case* digunakan untuk menguji kemungkinan alur pada proses pengajuan sakit, izin, dan cuti.

Tabel 5. 10 *Test Case* Fungsi Pengajuan Presensi

<i>Test Case</i>	Kondisi	Jalur Eksekusi	Output
TC1	Keterangan pengajuan kosong	1 → 2 → 25	Muncul pesan “Keterangan wajib diisi”
TC2	Pengajuan cuti tetapi tanggal awal atau tanggal akhir belum diisi	1 → 3 → 4 → 25	Muncul pesan “Tanggal awal dan akhir cuti wajib diisi”
TC3	Tanggal akhir cuti lebih awal dari tanggal awal cuti	1 → 3 → 5 → 6 → 25	Muncul pesan “Tanggal akhir cuti tidak boleh lebih awal”
TC4	Jumlah tidak masuk melebihi batas 12 kali dalam satu tahun	1 → 3 → 5 → 7 → 8 → 9 → 10 → 11 → 12 → 13 → 14 → 25	Muncul pesan batas tidak masuk 12 kali sudah tercapai
TC5	Data presensi atau pengajuan pada tanggal tersebut sudah ada	1 → 3 → 5 → 7 → 8 → 9 → 10 → 11 → 13 → 15 → 16 → 17 → 18 → 19 → 20 → 25	Muncul pesan data presensi atau pengajuan sudah ada
TC6	Pengajuan sakit atau izin valid	1 → 3 → 5 → 7 → 8 → 9 → 10 → 11 → 13 → 15 → 16 → 17 → 18 → 19 → 21 → 22 → 23 → 25	Data pengajuan sakit/izin berhasil dikirim
TC7	Pengajuan cuti valid	1 → 3 → 5 → 7 → 8 → 9 → 10 → 11 → 12 → 13 → 15 → 16 → 17 → 18 → 19 → 21 → 22 → 23 → 25	Data pengajuan cuti berhasil dikirim

Sumber : data diolah oleh penulis 2026

h. Statement Coverage Fungsi Pengajuan Presensi

Statement coverage digunakan untuk mengukur sejauh mana seluruh *statement* pada fungsi pengajuan presensi telah dieksekusi melalui *test case* yang telah dirancang. Pengujian ini bertujuan untuk memastikan setiap *statement* pada proses pengajuan sakit, izin, dan cuti telah dijalankan minimal satu kali. *Statement* yang tereksekusi pada masing-masing *test case* adalah sebagai berikut:

1. TC1:
S1, S2, S25
2. TC2:
S1, S3, S4, S25
3. TC3:
S1, S3, S5, S6, S25
4. TC4:
S1, S3, S5, S7, S8, S9, S10, S11, S12, S13, S14, S25
5. TC5:
S1, S3, S5, S7, S8, S9, S10, S11, S13, S15, S16, S17, S18, S19, S20, S25
6. TC6:
S1, S3, S5, S7, S8, S9, S10, S11, S13, S15, S16, S17, S18, S19, S21, S22, S23, S25
7. TC7:
S1, S3, S5, S7, S8, S9, S10, S11, S12, S13, S15, S16, S17, S18, S19, S21, S22, S23, S25

Berdasarkan hasil pengujian, seluruh *statement* pada fungsi pengajuan presensi yaitu S1 sampai S25 berhasil dieksekusi melalui kombinasi *test case* TC1 sampai TC7. Dengan demikian diperoleh nilai *statement coverage* sebesar:

$$(25 / 25) \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *statement* pada fungsi pengajuan presensi telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses *white box testing*.

5.4.6 Pengujian Fungsi Persetujuan Presensi

a. Potongan *Source Code* Fungsi Persetujuan Presensi

Fungsi persetujuan presensi digunakan oleh *hrd* untuk memproses pengajuan sakit, izin, dan cuti yang dikirim oleh karyawan. Pada fungsi ini, sistem memperbarui status pengajuan pada *collection* pengajuan. Jika pengajuan disetujui, maka data akan disimpan ke *collection* presensi sesuai jenis pengajuan. Jika pengajuan ditolak, maka data presensi akan disimpan dengan status *alpha*. Selain itu, setiap proses persetujuan maupun penolakan juga dicatat ke *collection*

log_aktivitas. Berikut adalah potongan *source code* fungsi persetujuan presensi yang digunakan dalam pengujian:

```
Future<void> _updatestatus(
    BuildContext pageContext,
    String docId,
    String approvalStatus,
    String jenisPengajuan,
    Map<String, dynamic> data,
) async {
    try {
        final pengajuanRef = FirebaseFirestore.instance
            .collection('pengajuan')
            .doc(docId);
        final tanggalString =
            data['tanggal_string'] ??
            DateFormat('yyyy-MM-dd').format(
                (data['tanggal'] as Timestamp).toDate(),
            );
        final uid = '${data['uid']}';
        final presensiDocId = '${uid}_$tanggalString';
        await pengajuanRef.update({
            'approval_status': approvalStatus,
            'tanggal_persetujuan': Timestamp.now(),
            'diproses_oleh': FirebaseAuth.instance.currentUser?.uid,
        });
        if (approvalStatus == 'disetujui') {
            await FirebaseFirestore.instance
                .collection('presensi')
                .doc(presensiDocId)
                .set({
                    'uid': data['uid'],
                    'nama': data['nama'],
```

```

    'nik': data['nik'],
    'tanggal': data['tanggal'],
    'tanggal_string': tanggalString,
    'masuk': '-',
    'pulang': '-',
    'status': jenisPengajuan,
    'approval_status': 'disetujui',
    'keterangan': data['keterangan'] ?? '-',
    'metode': 'Pengajuan ${jenisPengajuan.toUpperCase()}',
    'dibuat_oleh': data['dibuat_oleh'],
    'tanggal_pengajuan': data['tanggal_pengajuan'],
    'tanggal_persetujuan': Timestamp.now(),
  });
  await _simpanLogAktivitas(
    aktivitas: 'Setujui Pengajuan',
    keterangan:
      'HRD menyetujui pengajuan $jenisPengajuan milik ${data['nama'] ?? '-'}',
  );
}
if (approvalStatus == 'ditolak') {
  await FirebaseFirestore.instance
    .collection('presensi')
    .doc(presensiDocId)
    .set({
      'uid': data['uid'],
      'nama': data['nama'],
      'nik': data['nik'],
      'tanggal': data['tanggal'],
      'tanggal_string': tanggalString,
      'masuk': '-',
      'pulang': '-',
      'status': 'alpha',

```

```

        'approval_status': 'ditolak',
        'jenis_pengajuan': jenisPengajuan,
        'hasil_pengajuan': 'ditolak',
        'keterangan': data['keterangan'] ?? '-',
        'metode': 'Pengajuan ${jenisPengajuan.toUpperCase()} Ditolak',
        'dibuat_oleh': data['dibuat_oleh'],
        'tanggal_pengajuan': data['tanggal_pengajuan'],
        'tanggal_persetujuan': Timestamp.now(),
    });
    await _simpanLogAktivitas(
      aktivitas: 'Tolak Pengajuan',
      keterangan:
        'HRD menolak pengajuan $jenisPengajuan milik ${data['nama'] ?? '-'} dan
status presensi menjadi alpha',
    );
  }
  if (!pageContext.mounted) return;
  ScaffoldMessenger.of(pageContext).showSnackBar(
    SnackBar(
      content: Text(
        approvalStatus == 'disetujui'
          ? 'Pengajuan berhasil disetujui'
          : 'Pengajuan ditolak dan menjadi alpha',
      ),
    ),
  );
} catch (e) {
  await _simpanLogAktivitas(
    aktivitas: 'Persetujuan Pengajuan',
    keterangan: 'Gagal memproses pengajuan ${data['nama'] ?? '-'}: $e',
  );
  if (!pageContext.mounted) return;

```

```

ScaffoldMessenger.of(pageContext).showSnackBar(
  SnackBar(content: Text('Gagal memperbarui pengajuan: $e')),
);
}
}

```

b. Identifikasi *Statement* Fungsi Persetujuan Presensi

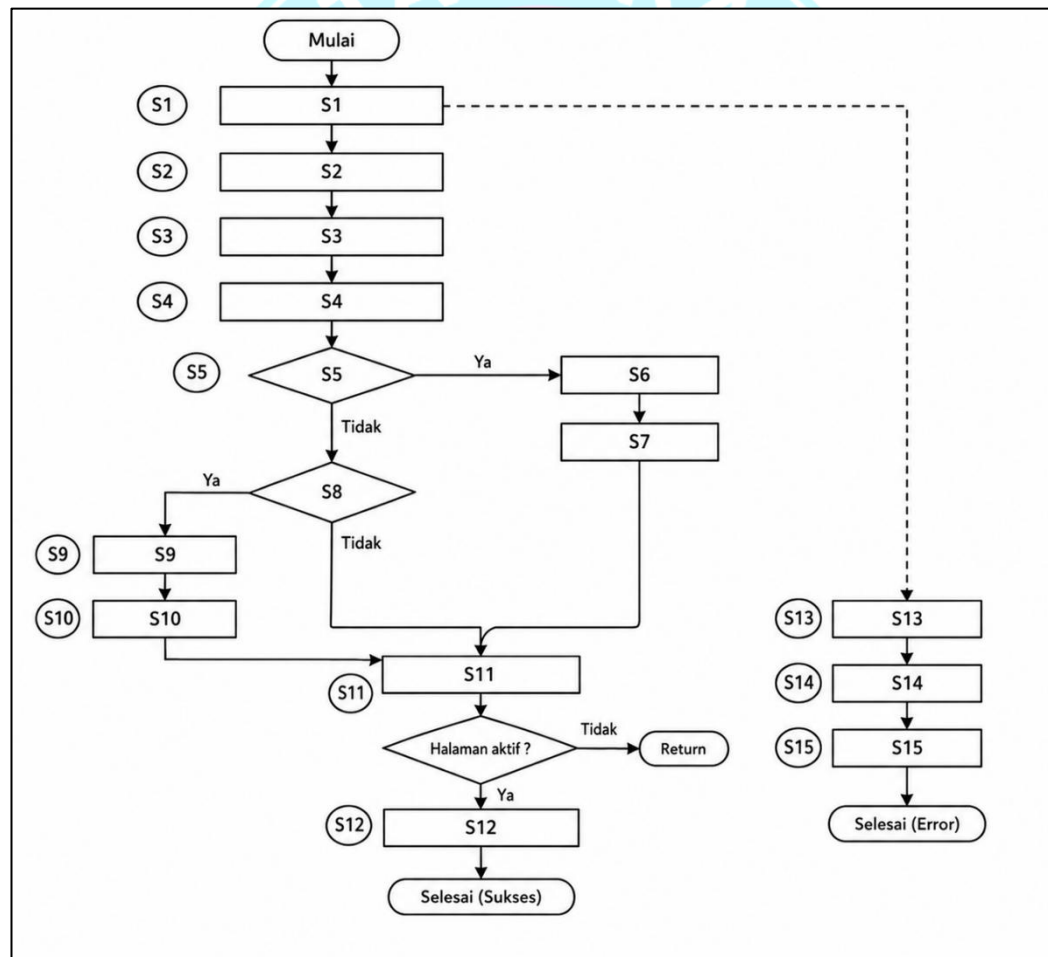
Tabel 5. 11 Identifikasi *Statement* Fungsi Persetujuan Presensi

<i>Statement</i>	<i>Source Code</i>	<i>Keterangan</i>
S1	<i>collection</i> ('pengajuan').doc(docId)	Mengambil referensi dokumen pengajuan
S2	<i>tanggalString</i> = data['tanggal_string'] ?? ...	Mengambil atau membentuk tanggal pengajuan
S3	<i>presensiDocId</i> = '\$ {uid} _\$tanggalString'	Membentuk ID dokumen presensi
S4	<i>pengajuanRef.update</i> ({...})	Memperbarui status pengajuan
S5	if (<i>approvalStatus</i> == 'disetujui')	Memeriksa apakah pengajuan disetujui
S6	<i>collection</i> ('presensi').doc(...).set({...})	Menyimpan data presensi sesuai jenis pengajuan
S7	_simpanLogAktivitas('Setujui Pengajuan')	Menyimpan log persetujuan pengajuan
S8	if (<i>approvalStatus</i> == 'ditolak')	Memeriksa apakah pengajuan ditolak
S9	<i>collection</i> ('presensi').doc(...).set({...status: alpha...})	Menyimpan data presensi dengan status alpha
S10	_simpanLogAktivitas('Tolak Pengajuan')	Menyimpan log penolakan pengajuan
S11	if (! <i>pageContext.mounted</i>) return	Memeriksa apakah halaman masih aktif

S12	SnackBar(...)	Menampilkan pesan hasil persetujuan atau penolakan
S13	<i>catch</i> (e)	Menangani <i>error</i> saat memproses pengajuan
S14	_simpanLogAktivitas('Persetujuan Pengajuan')	Menyimpan log gagal memproses pengajuan
S15	SnackBar('Gagal memperbarui pengajuan')	Menampilkan pesan <i>error</i> kepada pengguna

Sumber : data diolah oleh penulis, 2026

c. Flowchart Fungsi Persetujuan Presensi



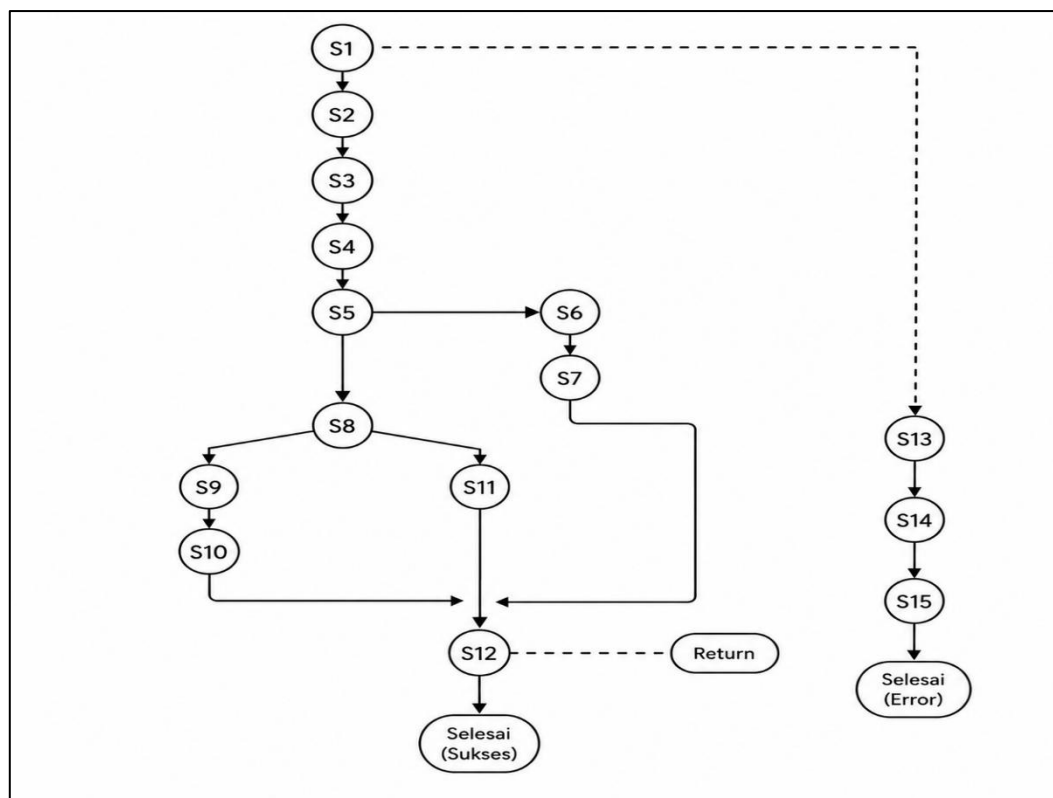
Sumber : data diolah oleh penulis, 2026

Gambar 5. 45 Flowchart Fungsi Persetujuan Presensi

Berdasarkan diagram alir pada gambar *flowchart* fungsi persetujuan presensi, urutan logika kerja fungsi persetujuan presensi dapat dijabarkan sebagai berikut:

1. S1 sampai S4 (Pengambilan dan Pembaruan Data Pengajuan)
Sistem memulai proses dengan mengambil referensi dokumen pengajuan berdasarkan docId. Selanjutnya, sistem mengambil atau membentuk tanggal pengajuan, membentuk id dokumen presensi, dan memperbarui status pengajuan pada *collection* pengajuan.
2. S5 sampai S7 (Pengajuan Disetujui)
Sistem memeriksa apakah status persetujuan bernilai disetujui. Jika benar, sistem menyimpan data presensi ke *collection* presensi dengan status sesuai jenis pengajuan, yaitu sakit, izin, atau cuti. Setelah itu, sistem menyimpan *log* aktivitas bahwa *hrd* telah menyetujui pengajuan.
3. S8 sampai S10 (Pengajuan Ditolak)
Jika status persetujuan bernilai ditolak, sistem menyimpan data presensi ke *collection* presensi dengan status alpha. Setelah itu, sistem menyimpan *log* aktivitas bahwa *hrd* telah menolak pengajuan dan status presensi menjadi *alpha*.
4. S11 dan S12 (Menampilkan Hasil Proses)
Setelah proses persetujuan atau penolakan selesai, sistem memeriksa apakah halaman masih aktif. Jika halaman masih aktif, sistem menampilkan pesan bahwa pengajuan berhasil disetujui atau ditolak.
5. S13 sampai S15 (Penanganan *Error*)
Jika terjadi kesalahan saat proses persetujuan, sistem menjalankan bagian *catch*, menyimpan *log* aktivitas gagal memproses pengajuan, kemudian menampilkan pesan *error* kepada pengguna.

d. *Flowgraph* Fungsi Persetujuan Presensi



Sumber : data diolah oleh penulis 2026

Gambar 5. 46 *Flowgraph* Fungsi Persetujuan Presensi

Berdasarkan gambar *flowgraph* fungsi persetujuan presensi, alur logika program fungsi persetujuan presensi dapat dijelaskan sebagai berikut. Proses dimulai pada *node* 1 dengan mengambil referensi dokumen pengajuan dari *collection* pengajuan berdasarkan *docId*. Selanjutnya pada *node* 2 sistem mengambil atau membentuk nilai tanggal_string yang akan digunakan sebagai identitas tanggal presensi. Pada *node* 3 sistem membentuk id dokumen presensi berdasarkan uid karyawan dan tanggal pengajuan. Setelah itu, *node* 4 digunakan untuk memperbarui status pengajuan, tanggal persetujuan, dan identitas pengguna yang memproses pengajuan.

Pada *node* 5 sistem memeriksa apakah status persetujuan bernilai disetujui. Jika kondisi bernilai benar, maka alur menuju *node* 6 untuk menyimpan data presensi ke *collection* presensi dengan status sesuai jenis pengajuan, yaitu sakit, izin, atau cuti. Setelah data berhasil disimpan, sistem menjalankan *node* 7 untuk mencatat aktivitas persetujuan ke *collection* log_aktivitas.

Jika kondisi pada *node* 5 bernilai salah, sistem melanjutkan ke *node* 8 untuk memeriksa apakah status persetujuan bernilai ditolak. Jika kondisi bernilai benar, maka sistem menjalankan *node* 9 untuk menyimpan data presensi ke *collection* presensi dengan status alpha dan status persetujuan ditolak. Selanjutnya pada *node* 10 sistem menyimpan *log* aktivitas penolakan pengajuan.

Baik jalur persetujuan maupun penolakan akan menuju *node* 11 untuk memeriksa apakah halaman masih aktif (*pagecontext.mounted*). Jika halaman masih aktif, sistem melanjutkan ke *node* 12 untuk menampilkan pesan hasil proses kepada pengguna, baik berupa pesan pengajuan berhasil disetujui maupun pengajuan berhasil ditolak. Apabila selama proses terjadi kesalahan, maka alur menuju *Node* 13 untuk menangani *exception* yang muncul.

Selanjutnya pada *node* 14 sistem menyimpan *log* aktivitas bahwa proses persetujuan pengajuan gagal dilakukan. Setelah itu, *node* 15 digunakan untuk menampilkan pesan *error* kepada pengguna yang menjelaskan bahwa proses pembaruan pengajuan gagal dilakukan. Dengan demikian, *flowgraph* fungsi persetujuan presensi menggambarkan tiga kemungkinan alur utama, yaitu alur persetujuan pengajuan, alur penolakan pengajuan, dan alur penanganan kesalahan (*error handling*), yang seluruhnya berakhir setelah sistem memberikan umpan balik kepada pengguna mengenai hasil proses yang dilakukan.

e. Cyclomatic Complexity Fungsi Persetujuan Presensi

Berdasarkan *flowgraph* yang telah dibentuk, terdapat 4 *predicate node* (P) pada fungsi persetujuan presensi. *Predicate node* tersebut yaitu *node* 5 yang memeriksa apakah pengajuan berstatus disetujui, *node* 8 yang memeriksa apakah pengajuan berstatus ditolak, *node* 11 yang memeriksa apakah halaman masih aktif, dan *node* 13 yang menangani *error* pada proses persetujuan. Oleh karena itu, nilai *cyclomatic complexity* dapat dihitung menggunakan rumus berikut:

$$V(G) = P + 1$$

$$V(G) = 4 + 1$$

$$V(G) = 5$$

Dari hasil perhitungan tersebut diperoleh nilai *cyclomatic complexity* sebesar 5. Nilai tersebut menunjukkan bahwa terdapat lima jalur independen dalam proses eksekusi fungsi persetujuan presensi.

f. Independent Path Fungsi Persetujuan Presensi

Berdasarkan hasil perhitungan *cyclomatic complexity* fungsi persetujuan presensi, diperoleh nilai $V(G) = 5$. Nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses *white box testing*. Sehingga didapatkan 5 jalur independen sebagai berikut:

1. *Path 1:*

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 11 \rightarrow 12$

Pengajuan disetujui dan data presensi disimpan sesuai jenis pengajuan.

2. *Path 2:*

$\rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12$

Pengajuan ditolak dan data presensi disimpan dengan status *alpha*.

3. *Path 3:*

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 8 \rightarrow 11 \rightarrow 12$

Status pengajuan tidak sesuai dengan disetujui atau ditolak.

4. *Path 4:*

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 11$

Pengajuan berhasil diproses, tetapi halaman sudah tidak aktif sehingga pesan tidak ditampilkan.

5. *Path 5:*

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 13 \rightarrow 14 \rightarrow 15$

Terjadi *error* saat memproses pengajuan sehingga sistem menampilkan pesan gagal memperbarui pengajuan.

g. Test Case Fungsi Persetujuan Presensi

Pengujian fungsi persetujuan presensi dilakukan berdasarkan jalur independen yang telah diperoleh dari hasil analisis *flowgraph*. Setiap *test case*

digunakan untuk menguji proses persetujuan, penolakan, status tidak valid, kondisi halaman tidak aktif, dan penanganan *error*.

Tabel 5. 12 *Test Case* Fungsi Persetujuan Presensi

<i>Test Case</i>	Kondisi	Jalur Eksekusi	<i>Output</i>
TC1	Pengajuan disetujui	1 → 2 → 3 → 4 → 5 → 6 → 7 → 11 → 12	Data masuk ke <i>collection</i> presensi sesuai jenis pengajuan
TC2	Pengajuan ditolak	1 → 2 → 3 → 4 → 5 → 8 → 9 → 10 → 11 → 12	Data presensi tersimpan dengan status alpha
TC3	Status pengajuan tidak sesuai	1 → 2 → 3 → 4 → 5 → 8 → 11 → 12	Data tidak diproses sebagai disetujui atau ditolak
TC4	Halaman sudah tidak aktif	1 → 2 → 3 → 4 → 5 → 6 → 7 → 11	Proses berhenti tanpa menampilkan pesan
TC5	Terjadi <i>error</i> saat proses persetujuan	1 → 2 → 3 → 4 → 13 → 14 → 15	Muncul pesan gagal memperbarui pengajuan

Sumber : data diolah oleh penulis 2026

h. *Statement Coverage* Fungsi Persetujuan Presensi

Statement coverage digunakan untuk mengukur sejauh mana seluruh *statement* pada fungsi persetujuan presensi telah dieksekusi melalui *test case* yang telah dirancang. Pengujian ini bertujuan untuk memastikan setiap *statement* pada proses persetujuan, penolakan, penyimpanan data presensi, pencatatan *log* aktivitas, dan penanganan *error* telah dijalankan minimal satu kali. *Statement* yang tereksekusi pada masing-masing *test case* adalah sebagai berikut:

1. TC1:
S1, S2, S3, S4, S5, S6, S7, S11, S12
2. TC2:
S1, S2, S3, S4, S5, S8, S9, S10, S11, S12
3. TC3:
S1, S2, S3, S4, S5, S8, S11, S12
4. TC4:
S1, S2, S3, S4, S5, S6, S7, S11

5. TC5:

S1, S2, S3, S4, S13, S14, S15

Berdasarkan hasil pengujian, seluruh *statement* pada fungsi persetujuan presensi yaitu S1 sampai S15 berhasil dieksekusi melalui kombinasi *test case* TC1 sampai TC5. Dengan demikian diperoleh nilai *statement coverage* sebesar:

$$(15 / 15) \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *statement* pada fungsi persetujuan presensi telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses *white box testing*.

5.4.7 Pengujian Fungsi Tambah Karyawan

a. Potongan *Source Code* Fungsi Tambah Karyawan

Fungsi tambah karyawan digunakan oleh *hrd* untuk menambahkan data karyawan baru ke dalam aplikasi. Pada proses ini, sistem terlebih dahulu melakukan validasi *form* dan memeriksa kelengkapan tanggal tmt serta tanggal lahir. Jika data sudah lengkap, sistem membuat data karyawan, membuat akun *login* karyawan menggunakan *firebase authentication* melalui *secondary firebase app*, menyimpan data karyawan ke *collection users*, menyimpan *log* aktivitas, lalu mengembalikan pengguna ke halaman sebelumnya setelah proses berhasil. Berikut adalah potongan *source code* fungsi tambah karyawan yang digunakan dalam pengujian:

```
Future<void> _simpan() async {
  if (!_formKey.currentState!.validate()) return;
  if (tmt == null || tanggalLahir == null) {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('TMT dan Tanggal Lahir wajib diisi')),
    );
    return;
  }

  setState(() => isLoading = true);

  try {
```

```

final data = {
    'nik': nikC.text.trim(),
    'nik_ktp': nikKtpC.text.trim(),
    'nama': namaC.text.trim(),
    'gol': golC.text.trim(),
    'jabatan': jabatanC.text.trim(),
    'divisi': divisiC.text.trim(),
    'tmt': Timestamp.fromDate(tmt!),
    'tanggal_lahir': Timestamp.fromDate(tanggalLahir!),
    'status': statusC.text.trim().toLowerCase(),
    'email': emailC.text.trim(),
    'telepon': teleponC.text.trim(),
    'alamat': alamatC.text.trim(),
    'role': 'karyawan',
    'updated_at': FieldValue.serverTimestamp(),
};

final secondaryApp = await Firebase.initializeApp(
    name: 'SecondaryApp',
    options: DefaultFirebaseOptions.currentPlatform,
);

try {
    final secondaryAuth = FirebaseAuth.instanceFor(app: secondaryApp);

    final credential = await secondaryAuth.createUserWithEmailAndPassword(
        email: emailC.text.trim(),
        password: passwordC.text.trim(),
    );

    final uid = credential.user!.uid;

```

```

await FirebaseFirestore.instance.collection('users').doc(uid).set({
  ...data,
  'created_at': FieldValue.serverTimestamp(),
});

await _simpanLogAktivitas(
  aktivitas: 'Tambah Karyawan',
  keterangan: 'HRD menambahkan karyawan baru ${namaC.text.trim()}',
);

await secondaryAuth.signOut();
} finally {
  await secondaryApp.delete();
}

if (!mounted) return;

ScaffoldMessenger.of(context).showSnackBar(
  const SnackBar(
    content: Text('Data karyawan dan akun login berhasil dibuat'),
  ),
);

Navigator.pop(context);
} on FirebaseAuthException catch (e) {
  String message = 'Gagal membuat akun';

  if (e.code == 'email-already-in-use') {
    message = 'Email sudah digunakan oleh akun lain';
  } else if (e.code == 'weak-password') {
    message = 'Password terlalu lemah, minimal 6 karakter';
  } else if (e.code == 'invalid-email') {
    message = 'Format email tidak valid';
  }
}

```

```

    }

    await _simpanLogAktivitas(
      aktivitas: 'Tambah Karyawan',
      keterangan: '$message untuk ${namaC.text.trim()}',
    );

    if (!mounted) return;

    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text(message)),
    );
  } catch (e) {
    await _simpanLogAktivitas(
      aktivitas: 'Tambah Karyawan',
      keterangan: 'Gagal menyimpan data ${namaC.text.trim()}: $e',
    );

    if (!mounted) return;

    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text('Gagal menyimpan data: $e')),
    );
  } finally {
    if (mounted) {
      setState(() => isLoading = false);
    }
  }
}

```

b. Identifikasi *Statement* Fungsi Tambah Karyawan

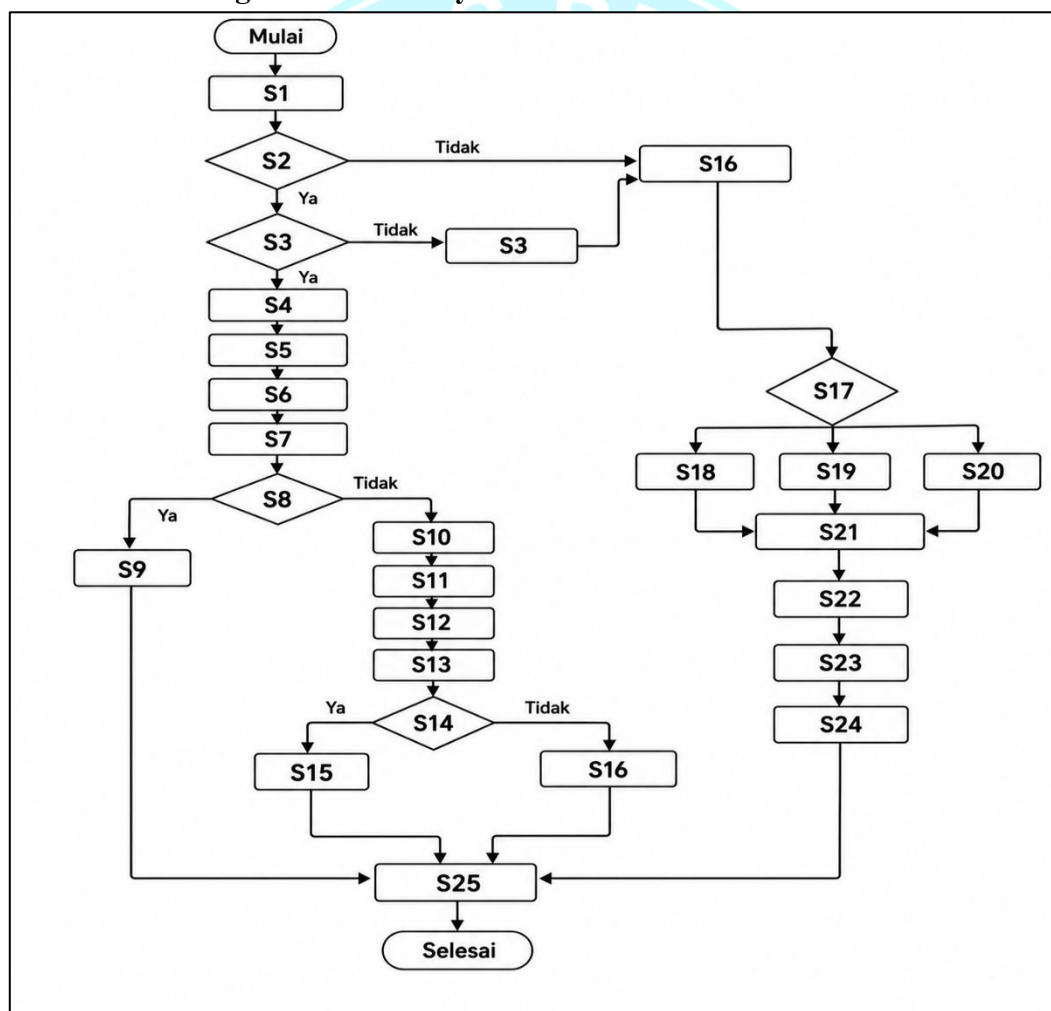
Tabel 5. 13 Identifikasi *Statement* Fungsi Tambah Karyawan

<i>Statement</i>	<i>Source Code</i>	<i>Keterangan</i>
S1	<code>_formKey.currentState!.validate()</code>	Memvalidasi seluruh <i>input form</i> tambah karyawan
S2	<code>`if (tmt == null</code>	
S3	<code>SnackBar("TMT dan Tanggal Lahir wajib diisi")</code>	Menampilkan pesan TMT dan tanggal lahir wajib diisi
S4	<code>setState(() => isLoading = true)</code>	Mengaktifkan <i>loading</i>
S5	<code>final data = {...}</code>	Membentuk data karyawan yang akan disimpan
S6	<code>Firebase.initializeApp(name: 'SecondaryApp')</code>	Membuat <i>secondary Firebase App</i>
S7	<code>FirebaseAuth.instanceFor(app: secondaryApp)</code>	Mengambil <i>instance Firebase Auth</i> dari <i>secondary app</i>
S8	<code>createUserWithEmailAndPassword(...)</code>	Membuat akun <i>login</i> karyawan baru
S9	<code>final uid = credential.user!.uid</code>	Mengambil UID akun karyawan baru
S10	<code>collection('users').doc(uid).set({...})</code>	Menyimpan data karyawan ke <i>collection users</i>
S11	<code>_simpanLogAktivitas("Tambah Karyawan")</code>	Menyimpan log aktivitas tambah karyawan
S12	<code>secondaryAuth.signOut()</code>	<i>Logout</i> dari <i>secondary auth</i>
S13	<code>secondaryApp.delete()</code>	Menghapus <i>secondary Firebase App</i>
S14	<code>SnackBar("Data karyawan dan akun <i>login</i> berhasil dibuat")</code>	Menampilkan pesan tambah karyawan berhasil
S15	<code>Navigator.pop(context)</code>	Kembali ke halaman sebelumnya
S16	<code>on FirebaseAuthException catch (e)</code>	Menangani <i>error Firebase Authentication</i>
S17	<code>if (e.code == 'email-already-in-use')</code>	Memeriksa <i>email</i> sudah digunakan
S18	<code>else if (e.code == 'weak-password')</code>	Memeriksa <i>password</i> terlalu lemah
S19	<code>else if (e.code == 'invalid-email')</code>	Memeriksa <i>format email</i> tidak valid
S20	<code>_simpanLogAktivitas(...) pada error Firebase</code>	Menyimpan log gagal tambah karyawan

S21	SnackBar(message)	Menampilkan pesan <i>error</i> <i>Firebase</i>
S22	<i>catch</i> (e)	Menangani <i>error</i> umum saat menyimpan data
S23	<i>_simpanLogAktivitas</i> (...) pada <i>error</i> umum	Menyimpan log <i>error</i> umum
S24	SnackBar('Gagal menyimpan data')	Menampilkan pesan gagal menyimpan data
S25	<i>finally</i> { <i>setState</i> ((<i>=></i> <i>isLoading</i> = <i>false</i>); }	Menghentikan <i>loading</i> pada akhir proses

Sumber : data diolah oleh penulis 2026

c. Flowchart Fungsi Tambah Karyawan



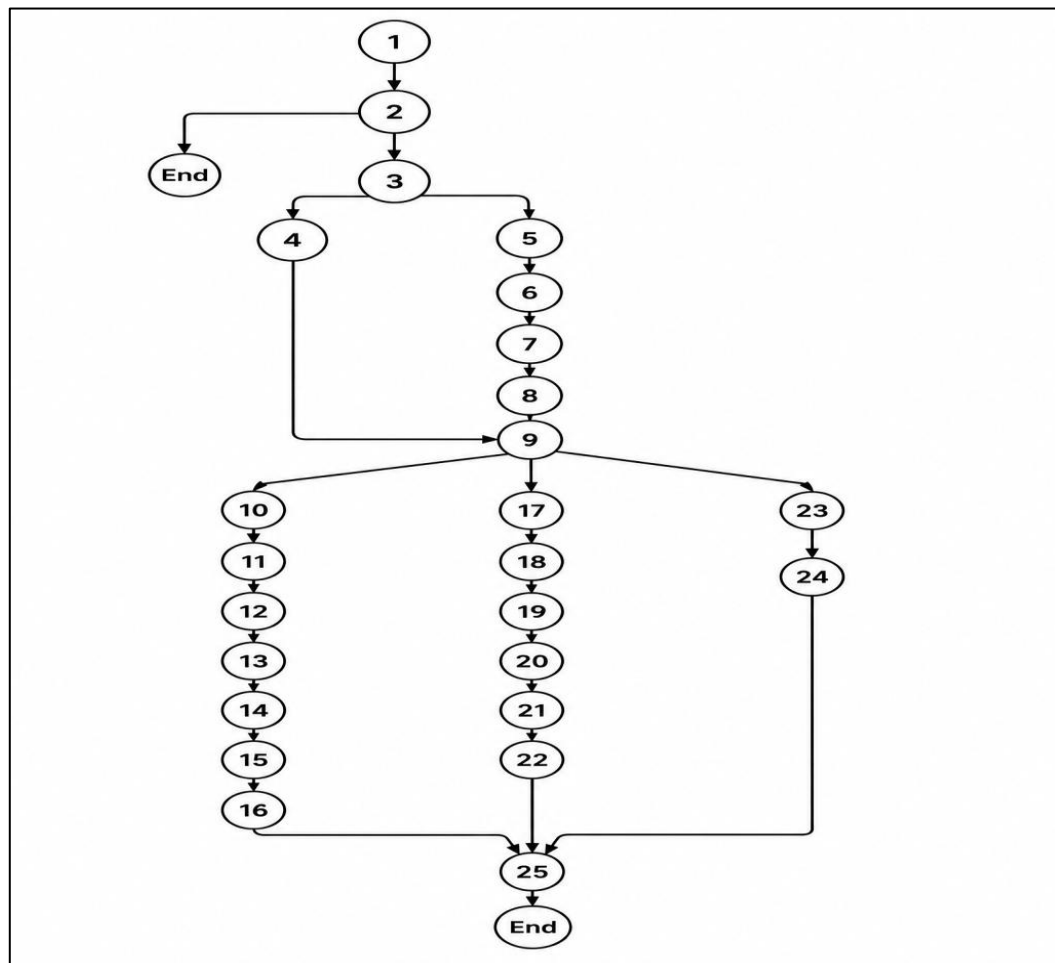
Sumber : data diolah oleh penulis 2026

Gambar 5. 47 Flowchart fungsi tambah, *edit*, dan hapus data karyawan

Berdasarkan diagram alir pada gambar *flowchart* fungsi tambah karyawan, urutan logika kerja fungsi tambah karyawan dapat dijabarkan sebagai berikut:

1. S1 sampai S3 (Validasi *Form* dan Tanggal)
Sistem memulai proses tambah karyawan dengan memvalidasi seluruh *input form*. Jika *form* belum *valid*, proses dihentikan. Selanjutnya sistem memeriksa apakah tmt dan tanggal lahir sudah diisi. Jika belum, sistem menampilkan pesan “tmt dan tanggal lahir wajib diisi”.
2. S4 sampai S8 (Persiapan Data dan Pembuatan Akun)
Jika data sudah *valid*, sistem mengaktifkan *loading*, membentuk data karyawan, membuat *secondary firebase app*, mengambil *instance firebase Auth*, lalu membuat akun *login* karyawan baru menggunakan *email* dan *password*.
3. S9 sampai S13 (Penyimpanan Data Karyawan)
Setelah akun berhasil dibuat, sistem mengambil uid akun karyawan, menyimpan data karyawan ke *collection users*, menyimpan *log* aktivitas tambah karyawan, *logout* dari *secondary auth*, lalu menghapus *secondary firebase app*.
4. S14 dan S15 (Tambah Karyawan Berhasil)
Jika proses tambah karyawan berhasil, sistem menampilkan pesan bahwa data karyawan dan akun *login* berhasil dibuat, kemudian kembali ke halaman sebelumnya.
5. S16 sampai S21 (*Error Firebase Authentication*)
Jika terjadi *error firebase authentication*, sistem menangkap *error* tersebut, menentukan jenis *error* seperti *email* sudah digunakan, *password* terlalu lemah, atau *format email* tidak valid. Setelah itu sistem menyimpan *log* gagal dan menampilkan pesan *error* kepada pengguna.
6. S22 sampai S25 (*Error Umum dan Akhir Proses*)
Jika terjadi *error* umum saat proses penyimpanan data, sistem menyimpan *log error* dan menampilkan pesan gagal menyimpan data. Pada akhir proses, sistem menghentikan *loading* melalui blok *finally*.

d. *Flowgraph* Fungsi Tambah Karyawan



Sumber : data diolah oleh penulis 2026

Gambar 5. 48 *Flowgraph* Fungsi Tambah Karyawan

Berdasarkan gambar *flowgraph* fungsi tambah karyawan, alur logika program dapat dijelaskan sebagai berikut. Proses dimulai pada *node* 1 dengan melakukan validasi seluruh *input form* yang telah diisi oleh *hrd*. Selanjutnya pada *node* 2 sistem memeriksa apakah seluruh data pada *form* telah valid. Jika *form* tidak *valid*, maka proses dihentikan. Jika *form valid*, sistem melanjutkan ke *node* 3 untuk memeriksa apakah *tmt* dan tanggal lahir telah diisi. Jika salah satu data belum diisi, maka sistem menjalankan *node* 4 untuk menampilkan pesan bahwa *tmt* dan tanggal lahir wajib diisi, kemudian proses berakhir.

Jika seluruh data telah lengkap, sistem menjalankan *node* 5 untuk mengaktifkan *loading* dan *node* 6 untuk membentuk data karyawan yang akan disimpan ke database. Selanjutnya *node* 7 digunakan untuk membuat *secondary firebase app* dan *node* 8 digunakan untuk mengambil *instance firebase*

authentication dari *secondary app*. Pada *node 9* sistem membuat akun *login* karyawan baru menggunakan *email* dan *password* yang telah diinputkan. Jika proses pembuatan akun berhasil, maka sistem menjalankan *node 10* untuk mengambil uid akun yang baru dibuat. Selanjutnya *node 11* digunakan untuk menyimpan data karyawan ke *collection users*, kemudian *node 12* digunakan untuk menyimpan *log* aktivitas tambah karyawan.

Setelah data berhasil disimpan, sistem menjalankan *node 13* untuk *logout* dari *secondary authentication* dan *node 14* untuk menghapus *secondary firebase app*. Selanjutnya sistem menjalankan *node 15* untuk menampilkan pesan bahwa data karyawan dan akun *login* berhasil dibuat, kemudian *node 16* digunakan untuk kembali ke halaman sebelumnya. Apabila terjadi kesalahan pada proses pembuatan akun menggunakan *firebase authentication*, maka sistem menjalankan *node 17* untuk menangkap *exception firebase*.

Selanjutnya *node 18* digunakan untuk memeriksa apakah *email* sudah digunakan oleh akun lain, *node 19* memeriksa apakah *password* terlalu lemah, dan *node 20* memeriksa apakah *format email* tidak valid. Setelah jenis *error* ditentukan, sistem menjalankan *node 21* untuk menyimpan *log* aktivitas gagal tambah karyawan dan *node 22* untuk menampilkan pesan *error* kepada pengguna.

Jika terjadi kesalahan umum selain *firebase authentication*, maka sistem menjalankan *node 23* untuk menangani *exception* umum. Selanjutnya *node 24* digunakan untuk menyimpan *log* aktivitas kesalahan dan menampilkan pesan gagal menyimpan data. Pada akhir seluruh proses, sistem menjalankan *node 25* untuk menghentikan *loading* melalui blok *finally*, kemudian proses berakhir.

e. Cyclomatic Complexity Fungsi Tambah Karyawan

Berdasarkan *flowgraph* yang telah dibentuk, terdapat 5 *predicate node* (P) pada fungsi tambah karyawan. *Predicate node* tersebut yaitu *node 2* yang memeriksa apakah *form valid*, *node 3* yang memeriksa apakah tmt dan tanggal lahir sudah diisi, *node 18* yang memeriksa *error email-already-in-use*, *node 19* yang memeriksa *error weak-password*, dan *node 20* yang memeriksa *error invalid-email*. Maka nilai *cyclomatic complexity* dapat dihitung sebagai berikut:

$$V(G) = P + 1$$

$$V(G) = 5 + 1$$

$$V(G) = 6$$

Dari hasil perhitungan tersebut diperoleh nilai *cyclomatic complexity* sebesar 6. Nilai tersebut menunjukkan bahwa terdapat enam jalur independen pada fungsi tambah karyawan.

f. Independent Path Fungsi Tambah Karyawan

Berdasarkan hasil perhitungan *cyclomatic complexity* fungsi tambah karyawan, diperoleh nilai $V(G) = 6$. Nilai tersebut menunjukkan bahwa terdapat enam jalur independen yang harus diuji untuk memastikan seluruh kemungkinan alur program telah dieksekusi. Adapun jalur independen pada fungsi tambah karyawan adalah sebagai berikut:

1. Path 1

$1 \rightarrow 2$

Jalur ini terjadi ketika *form* tidak valid sehingga proses langsung dihentikan.

2. Path 2

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 25$

Jalur ini terjadi ketika *tmt* atau tanggal lahir belum diisi sehingga sistem menampilkan pesan bahwa *tmt* dan tanggal lahir wajib diisi.

3. Path 3

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 16 \rightarrow 25$

Jalur ini terjadi ketika seluruh data valid dan proses tambah karyawan berhasil dilakukan sampai akun *login* dan data karyawan berhasil dibuat.

4. Path 4

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 17 \rightarrow 18 \rightarrow 21 \rightarrow 22 \rightarrow 25$

Jalur ini terjadi ketika proses tambah karyawan gagal karena *email* yang digunakan sudah terdaftar pada *firebase authentication*.

5. Path 5

1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 17 → 18 → 19 → 21 → 22 → 25

Jalur ini terjadi ketika proses tambah karyawan gagal karena *password* yang digunakan terlalu lemah.

6. Path 6

1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 17 → 18 → 19 → 20 → 21 → 22 → 25

Jalur ini terjadi ketika proses tambah karyawan gagal karena *format email* yang dimasukkan tidak *valid*. Dengan demikian, keenam jalur independen tersebut telah mewakili seluruh kemungkinan alur utama pada fungsi tambah karyawan dan akan digunakan sebagai dasar penyusunan *test case* pada tahap pengujian berikutnya.

g. Test Case Fungsi Tambah Karyawan

Pengujian fungsi tambah karyawan dilakukan berdasarkan jalur independen yang telah diperoleh dari hasil analisis *flowgraph*. Setiap *test case* digunakan untuk menguji proses validasi *form*, validasi tanggal, pembuatan akun karyawan, penyimpanan data ke *collection users*, serta penanganan *error firebase authentication*.

Tabel 5. 14 Test Case Fungsi Tambah Karyawan

Test Case	Kondisi	Jalur Eksekusi	Output
TC1	Form tidak valid	1 → 2	Proses tambah karyawan dihentikan
TC2	TMT atau tanggal lahir belum diisi	1 → 2 → 3 → 4 → 25	Muncul pesan “TMT dan Tanggal Lahir wajib diisi”
TC3	Semua data valid	1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12 → 13 → 14 → 15 → 16 → 25	Data karyawan dan akun <i>login</i> berhasil dibuat
TC4	Email sudah digunakan	1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 17 → 18 → 21 → 22 → 25	Muncul pesan “Email sudah digunakan oleh akun lain”

TC5	Password terlalu lemah	1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 17 → 18 → 19 → 21 → 22 → 25	Muncul pesan “Password terlalu lemah, minimal 6 karakter”
TC6	Format email tidak valid	1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 17 → 18 → 19 → 20 → 21 → 22 → 25	Muncul pesan “Format email tidak valid”

Sumber : data diolah oleh penulis 2026

h. Statement Coverage Fungsi Tambah Karyawan

Statement coverage digunakan untuk mengukur sejauh mana seluruh *statement* pada fungsi tambah karyawan telah dieksekusi melalui *test case* yang telah dirancang. Pengujian ini bertujuan untuk memastikan setiap *statement* pada proses validasi *form*, validasi tanggal, pembuatan akun karyawan, penyimpanan data ke *collection users*, pencatatan *log* aktivitas, dan penanganan *error* telah dijalankan minimal satu kali. *Statement* yang tereksekusi pada masing-masing *test case* adalah sebagai berikut:

1. TC1:
S1, S2
2. TC2:
S1, S2, S3, S4, S25
3. TC3:
S1, S2, S3, S5, S6, S7, S8, S9, S10, S11, S12, S13, S14, S15, S16, S25
4. TC4:
S1, S2, S3, S5, S6, S7, S8, S9, S17, S18, S21, S22, S25
5. TC5:
S1, S2, S3, S5, S6, S7, S8, S9, S17, S18, S19, S21, S22, S25
6. TC6:
S1, S2, S3, S5, S6, S7, S8, S9, S17, S18, S19, S20, S21, S22, S25

Berdasarkan hasil pengujian, seluruh *statement* pada fungsi tambah karyawan yaitu S1 sampai S25 berhasil dieksekusi melalui kombinasi *test case* TC1 sampai TC6. Dengan demikian diperoleh nilai *statement coverage* sebesar:

$$(25 / 25) \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *statement* pada fungsi tambah karyawan telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses *white box testing*.

5.4.8 Pengujian Fungsi *Edit Data Karyawan*

a. Potongan *Source Code* Fungsi *Edit Data Karyawan*

Fungsi *edit* data karyawan digunakan oleh *hrd* untuk memperbarui data karyawan yang sudah tersimpan pada *collection users*. Pada proses *edit*, halaman menerima *docId* dan *initial* data, kemudian data lama ditampilkan ke dalam *form*. Setelah *hrd* melakukan perubahan, sistem memvalidasi *form*, memeriksa kelengkapan tmt dan tanggal lahir, membentuk data karyawan yang baru, lalu memperbarui data pada *firestore* menggunakan *update* (data). Setelah proses berhasil, sistem menyimpan *log* aktivitas *edit* karyawan, menampilkan pesan berhasil, dan kembali ke halaman sebelumnya. Berikut adalah potongan *source code* fungsi *edit* data karyawan yang digunakan dalam pengujian:

```
Future<void> _simpan() async {
  if (!_formKey.currentState!.validate()) return;

  if (tmt == null || tanggalLahir == null) {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('TMT dan Tanggal Lahir wajib diisi')),
    );
    return;
  }

  setState(() => isLoading = true);

  try {
    final data = {
      'nik': nikC.text.trim(),
      'nik_ktp': nikKtpC.text.trim(),
```

```

'nama': namaC.text.trim(),
'gol': golC.text.trim(),
'jabatan': jabatanC.text.trim(),
'divisi': divisiC.text.trim(),
'tmt': Timestamp.fromDate(tmt!),
'tanggal_lahir': Timestamp.fromDate(tanggalLahir!),
'status': statusC.text.trim().toLowerCase(),
'email': emailC.text.trim(),
'telepon': teleponC.text.trim(),
'alamat': alamatC.text.trim(),
'role': 'karyawan',
'updated_at': FieldValue.serverTimestamp(),
};

if (isEdit) {
  await FirebaseFirestore.instance
    .collection('users')
    .doc(widget.docId)
    .update(data);

  await _simpanLogAktivitas(
    aktivitas: 'Edit Karyawan',
    keterangan: 'HRD memperbarui data karyawan ${namaC.text.trim()}',
  );
}

if (!mounted) return;

ScaffoldMessenger.of(context).showSnackBar(
  const SnackBar(
    content: Text('Data karyawan berhasil diperbarui'),
  ),
);

```

```

);

Navigator.pop(context);
} catch (e) {
  await _simpanLogAktivitas(
    aktivitas: 'Edit Karyawan',
    keterangan: 'Gagal menyimpan data ${namaC.text.trim()}: $e',
  );

  if (!mounted) return;

  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text('Gagal menyimpan data: $e')),
  );
} finally {
  if (mounted) {
    setState(() => isLoading = false);
  }
}
}

```

b. Identifikasi *Statement Fungsi Edit Karyawan*

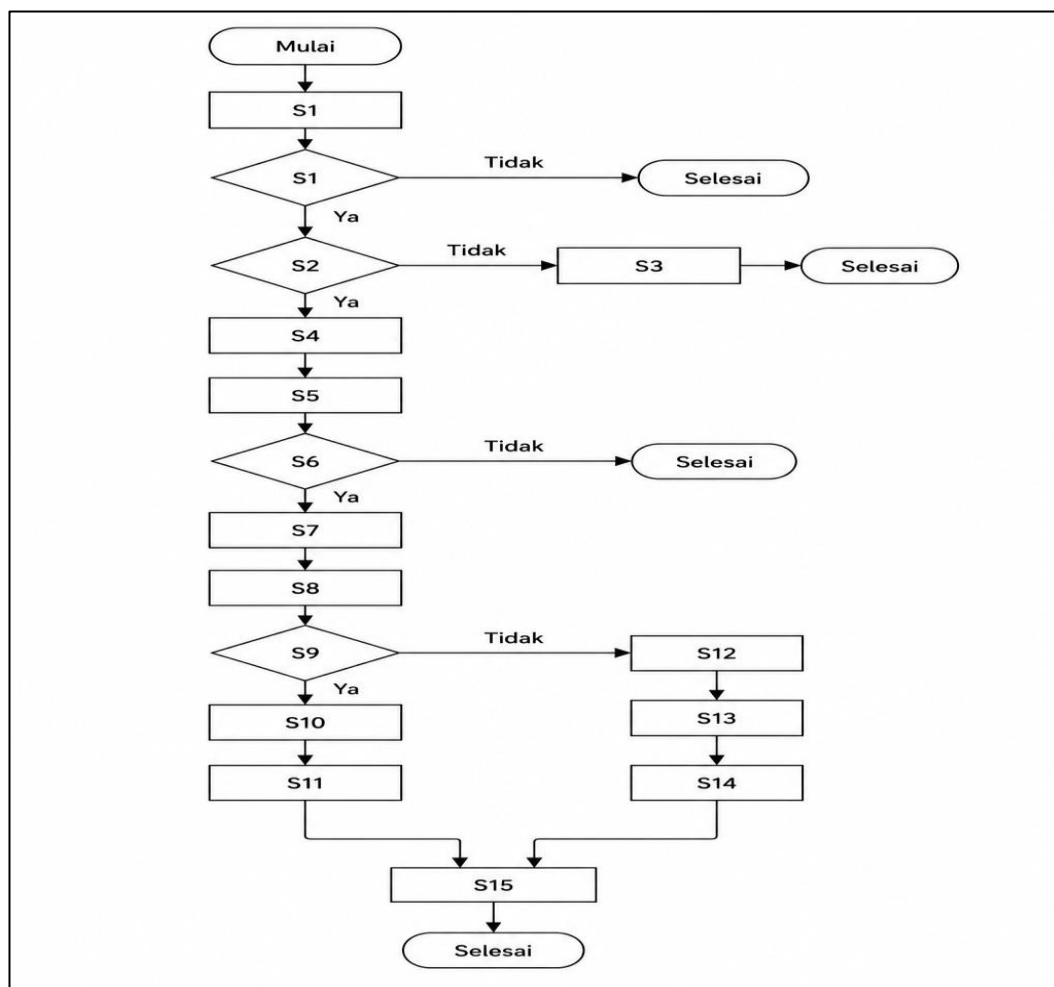
Tabel 5. 15 Identifikasi *Statement Fungsi Edit Karyawan*

<i>Statement</i>	<i>Source Code</i>	<i>Keterangan</i>
S1	<code>_formKey.currentState!.validate()</code>	Memvalidasi seluruh <i>input form edit</i> karyawan
S2	<code>`if (tmt == null</code>	
S3	<code>SnackBar('TMT dan Tanggal Lahir wajib diisi')</code>	Menampilkan pesan TMT dan tanggal lahir wajib diisi
S4	<code>setState(() => isLoading = true)</code>	Mengaktifkan <i>loading</i>
S5	<code>final data = {...}</code>	Membentuk data karyawan yang akan diperbarui

S6	<code>if (isEdit)</code>	Memeriksa apakah halaman berada pada <i>mode edit</i>
S7	<code>collection('users').doc(widget.docId).update(data)</code>	Memperbarui data karyawan pada <i>collection users</i>
S8	<code>_simpanLogAktivitas('Edit Karyawan')</code>	Menyimpan log aktivitas <i>edit</i> karyawan
S9	<code>if (!mounted) return</code>	Memeriksa apakah halaman masih aktif
S10	<code>SnackBar('Data karyawan berhasil diperbarui')</code>	Menampilkan pesan data karyawan berhasil diperbarui
S11	<code>Navigator.pop(context)</code>	Kembali ke halaman sebelumnya
S12	<code>catch (e)</code>	Menangani <i>error</i> saat proses <i>edit data</i>
S13	<code>_simpanLogAktivitas(...)</code> pada <i>error</i>	Menyimpan log gagal <i>edit data</i> karyawan
S14	<code>SnackBar('Gagal menyimpan data')</code>	Menampilkan pesan gagal menyimpan data
S15	<code>finally { setState(() => isLoading = false); }</code>	Menghentikan <i>loading</i> pada akhir proses

Sumber : data diolah oleh penulis 2026

c. Flowchart Fungsi Edit Karyawan



Sumber : data diolah oleh penulis 2026

Gambar 5. 49 Flowchart Fungsi Edit Karyawan

Berdasarkan diagram alir pada gambar *flowchart* fungsi *edit* karyawan, urutan logika kerja fungsi *edit* karyawan dapat dijabarkan sebagai berikut:

1. S1 sampai S3 (Validasi *Form* dan Tanggal)

Sistem memulai proses *edit* karyawan dengan memvalidasi seluruh *input form*. Jika *form* belum valid, proses dihentikan. Selanjutnya, sistem memeriksa apakah tmt dan tanggal lahir sudah diisi. Jika belum, sistem menampilkan pesan “tmt dan tanggal lahir wajib diisi”.

2. S4 sampai S6 (Persiapan Data *Edit*)

Jika data sudah *valid*, sistem mengaktifkan *loading*, membentuk data karyawan yang akan diperbarui, lalu memeriksa apakah halaman berada pada *mode edit*.

3. S7 dan S8 (*Update Data Karyawan*)

Jika *mode edit* benar, sistem memperbarui data karyawan pada *collection users*, kemudian menyimpan log aktivitas *edit* karyawan.

4. S9 sampai S11 (*Edit Berhasil*)

Setelah proses *update* berhasil, sistem memeriksa apakah halaman masih aktif. Jika masih aktif, sistem menampilkan pesan bahwa data karyawan berhasil diperbarui, kemudian kembali ke halaman sebelumnya.

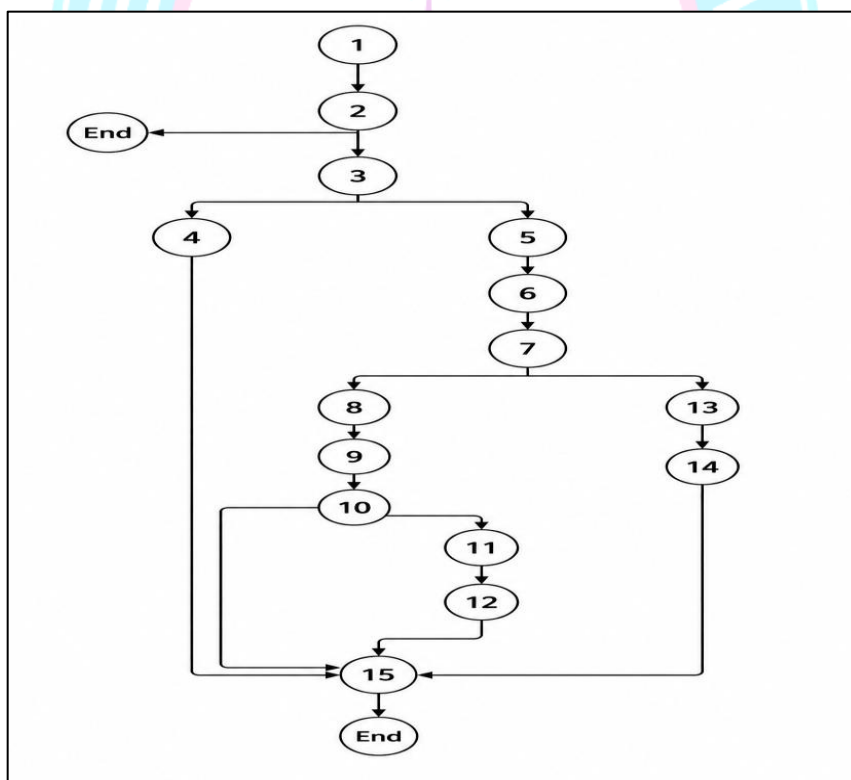
5. S12 sampai S14 (*Penanganan Error*)

Jika terjadi *error* saat proses *edit* data, sistem menangani *error*, menyimpan *log* gagal *edit* data karyawan, lalu menampilkan pesan gagal menyimpan data kepada pengguna.

6. S15 (*Akhir Proses*)

Pada akhir proses, sistem menghentikan *loading* melalui blok *finally*.

d. Flowgraph Fungsi Edit Karyawan



Sumber : data diolah oleh penulis 2026

Gambar 5. 50 Flowgraph Fungsi Edit Karyawan

Berdasarkan gambar *flowgraph* fungsi *edit* karyawan, alur logika program dapat dijelaskan sebagai berikut. Proses dimulai pada *node* 1 dengan melakukan validasi seluruh *input form* yang telah diisi oleh *hrd*. Selanjutnya pada *node* 2 sistem memeriksa apakah *form* valid. Jika *form* tidak valid, maka proses dihentikan. Jika *form* valid, sistem melanjutkan ke *node* 3 untuk memeriksa apakah *tmt* dan tanggal lahir telah diisi. Jika salah satu data belum diisi, maka sistem menjalankan *node* 4 untuk menampilkan pesan bahwa *tmt* dan tanggal lahir wajib diisi, kemudian proses berakhir.

Jika seluruh data telah lengkap, sistem menjalankan *node* 5 untuk mengaktifkan *loading* dan *node* 6 untuk membentuk data karyawan yang akan diperbarui. Selanjutnya *node* 7 digunakan untuk memeriksa apakah halaman berada pada *mode edit*. Jika kondisi *mode edit* bernilai benar, sistem menjalankan *node* 8 untuk memperbarui data karyawan pada *collection users*. Setelah proses *update* berhasil, sistem menjalankan *node* 9 untuk menyimpan *log* aktivitas *edit* karyawan.

Selanjutnya sistem menjalankan *node* 10 untuk memeriksa apakah halaman masih aktif (*mounted*). Jika halaman masih aktif, maka sistem menjalankan *node* 11 untuk menampilkan pesan bahwa data karyawan berhasil diperbarui dan *node* 12 untuk kembali ke halaman sebelumnya menggunakan *navigator.pop()*. Apabila terjadi kesalahan selama proses *update* data, sistem akan menjalankan *node* 13 untuk menangani *exception*. Selanjutnya *node* 14 digunakan untuk menyimpan *log* aktivitas gagal *edit* karyawan dan menampilkan pesan kesalahan kepada pengguna. Pada akhir seluruh proses, sistem menjalankan *node* 15 untuk menghentikan *loading* melalui blok *finally*, kemudian proses berakhir.

e. Cyclomatic Complexity Fungsi Edit Karyawan

Berdasarkan *flowgraph* yang telah dibentuk, terdapat 4 *predicate node* (P) pada fungsi *edit* karyawan. *Predicate node* tersebut yaitu *node* 2 yang memeriksa apakah *form* valid, *node* 3 yang memeriksa apakah *tmt* dan tanggal lahir sudah diisi, *node* 7 yang memeriksa apakah halaman berada pada *mode edit*, dan *node* 10 yang memeriksa apakah halaman masih aktif. Maka nilai *cyclomatic complexity* dapat dihitung sebagai berikut:

$$V(G) = P + 1$$

$$V(G) = 4 + 1$$

$$V(G) = 5$$

Dari hasil perhitungan tersebut diperoleh nilai *cyclomatic complexity* sebesar 5. Nilai tersebut menunjukkan bahwa terdapat lima jalur independen pada fungsi *edit* karyawan.

f. *Independent Path* Fungsi *Edit Karyawan*

Berdasarkan hasil perhitungan *cyclomatic complexity* fungsi *edit* karyawan, diperoleh nilai $V(G) = 5$. Nilai tersebut menunjukkan bahwa terdapat lima jalur independen yang harus diuji. Adapun jalur independen pada fungsi *edit* karyawan adalah sebagai berikut:

1. *Path 1*

$1 \rightarrow 2$

Form tidak valid sehingga proses *edit* karyawan dihentikan.

2. *Path 2*

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 15$

Tmt atau tanggal lahir belum diisi sehingga sistem menampilkan pesan wajib diisi.

3. *Path 3*

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 15$

Data *valid*, *mode edit* aktif, data karyawan berhasil diperbarui.

4. *Path 4*

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 15$

Data berhasil diperbarui, tetapi halaman sudah tidak aktif sehingga pesan berhasil tidak ditampilkan.

5. *Path 5*

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 13 \rightarrow 14 \rightarrow 15$

Terjadi *error* saat proses *edit* data karyawan.

g. Test Case Fungsi Edit Karyawan

Pengujian fungsi *edit* karyawan dilakukan berdasarkan jalur independen yang telah diperoleh dari hasil analisis *flowgraph*. Setiap *test case* digunakan untuk menguji proses validasi *form*, validasi tanggal, pembaruan data karyawan, kondisi halaman aktif, serta penanganan *error*.

Tabel 5. 16 Test Case Fungsi Edit Karyawan

Test Case	Kondisi	Jalur Eksekusi	Output
TC1	Form tidak valid	1 → 2	Proses <i>edit</i> karyawan dihentikan
TC2	TMT atau tanggal lahir belum diisi	1 → 2 → 3 → 4 → 15	Muncul pesan “TMT dan Tanggal Lahir wajib diisi”
TC3	Data valid dan halaman aktif	1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12 → 15	Data karyawan berhasil diperbarui
TC4	Data valid tetapi halaman tidak aktif	1 → 2 → 3 → 5 → 6 → 7 → 8 → 9 → 10 → 15	Data diperbarui, tetapi pesan berhasil tidak ditampilkan
TC5	Terjadi <i>error</i> saat <i>update data</i>	1 → 2 → 3 → 5 → 6 → 7 → 13 → 14 → 15	Muncul pesan gagal menyimpan data

Sumber : data diolah oleh penulis 2026

h. Statement Coverage Fungsi Edit Karyawan

Statement coverage digunakan untuk mengukur sejauh mana seluruh *statement* pada fungsi *edit* karyawan telah dieksekusi melalui *test case* yang telah dirancang. Pengujian ini bertujuan untuk memastikan setiap *statement* pada proses validasi *form*, validasi tanggal, pembaruan data karyawan, pencatatan *log* aktivitas, dan penanganan *error* telah dijalankan minimal satu kali. *Statement* yang tereksekusi pada masing-masing *test case* adalah sebagai berikut:

1. TC1:
S1, S2
2. TC2:
S1, S2, S3, S4, S15
3. TC3:
S1, S2, S3, S5, S6, S7, S8, S9, S10, S11, S12, S15

4. TC4:

S1, S2, S3, S5, S6, S7, S8, S9, S10, S15

5. TC5:

S1, S2, S3, S5, S6, S7, S13, S14, S15

Berdasarkan hasil pengujian, seluruh *statement* pada fungsi *edit* karyawan yaitu S1 sampai S15 berhasil dieksekusi melalui kombinasi *test case* TC1 sampai TC5. Dengan demikian diperoleh nilai *statement coverage* sebesar:

$$(15 / 15) \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *statement* pada fungsi *edit* karyawan telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses *white box testing*.

5.4.9 Pengujian Fungsi Hapus Data Karyawan

a. Potongan *Source Code* Fungsi Hapus Data Karyawan

Fungsi hapus data karyawan digunakan oleh *hrd* untuk menghapus data karyawan dari *collection users*. Proses ini diawali dengan menampilkan dialog konfirmasi agar pengguna tidak menghapus data secara tidak sengaja. Jika pengguna memilih tidak, maka proses hapus dibatalkan. Jika pengguna memilih ya, maka sistem menghapus dokumen karyawan berdasarkan *docid*, kemudian menampilkan pesan bahwa data karyawan berhasil dihapus. Berikut adalah potongan *source code* fungsi hapus data karyawan yang digunakan dalam pengujian:

```
void _hapus(String docId) {
  showDialog(
    context: context,
    builder: (_) => AlertDialog(
      title: const Text('Hapus Data'),
      content: const Text(
        'Apakah Anda yakin ingin menghapus data karyawan ini?',
      ),
      actions: [
```

```

    TextButton(
      onPressed: () => Navigator.pop(context),
      child: const Text('Tidak'),
    ),
    ElevatedButton(
      onPressed: () async {
        await FirebaseFirestore.instance
          .collection('users')
          .doc(docId)
          .delete();

        if (!mounted) return;

        Navigator.pop(context);
        ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(
            content: Text('Data karyawan berhasil dihapus'),
          ),
        );
      },
      child: const Text('Ya'),
    ),
  ],
),
);
}

```

b. Identifikasi *Statement* Fungsi Hapus Data Karyawan

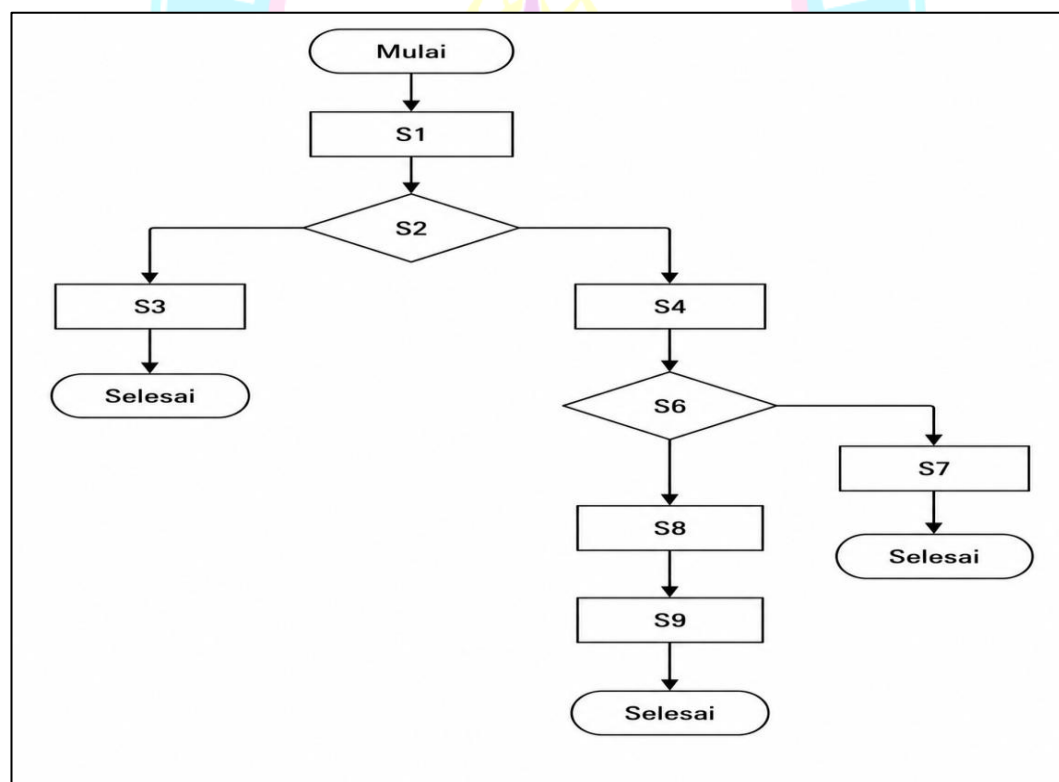
Tabel 5. 17 Identifikasi *Statement* Fungsi Hapus Data Karyawan

<i>Statement</i>	<i>Source Code</i>	Keterangan
S1	showDialog(...)	Menampilkan dialog konfirmasi hapus data

S2	<code>TextButton(onPressed: () => Navigator.pop(context))</code>	Tombol Tidak dipilih
S3	<code>Navigator.pop(context)</code>	Menutup dialog dan membatalkan proses hapus
S4	<code>ElevatedButton(...)</code>	Tombol Ya dipilih
S5	<code>collection('users').doc(docId).delete()</code>	Menghapus data karyawan dari <i>collection users</i>
S6	<code>if (!mounted) return</code>	Memeriksa apakah halaman masih aktif
S7	<code>return</code>	Menghentikan proses jika halaman tidak aktif
S8	<code>Navigator.pop(context)</code>	Menutup dialog konfirmasi setelah data berhasil dihapus
S9	<code>SnackBar('Data karyawan berhasil dihapus')</code>	Menampilkan pesan bahwa data karyawan berhasil dihapus

Sumber : data diolah oleh penulis 2026

c. Flowchart Fungsi Hapus Data Karyawan



Sumber : data diolah oleh penulis 2026

Gambar 5. 51 Flowchart Fungsi Hapus Data Karyawan

Berdasarkan diagram alir pada gambar *flowchart* fungsi hapus data Karyawan, urutan logika kerja fungsi hapus data karyawan dapat dijabarkan sebagai berikut:

1. S1 (Menampilkan Dialog Konfirmasi)

Sistem memulai proses hapus data dengan menampilkan dialog konfirmasi kepada *hrd* untuk memastikan apakah data karyawan benar-benar ingin dihapus.

2. S2 dan S3 (Hapus Dibatalkan)

Jika *hrd* memilih tombol tidak, maka sistem menjalankan S3 untuk menutup dialog konfirmasi dan proses hapus data dibatalkan.

3. S4 dan S5 (Hapus Dikonfirmasi)

Jika *hrd* memilih tombol ya, maka sistem menjalankan S5 untuk menghapus data karyawan dari *collection users* berdasarkan *docid*.

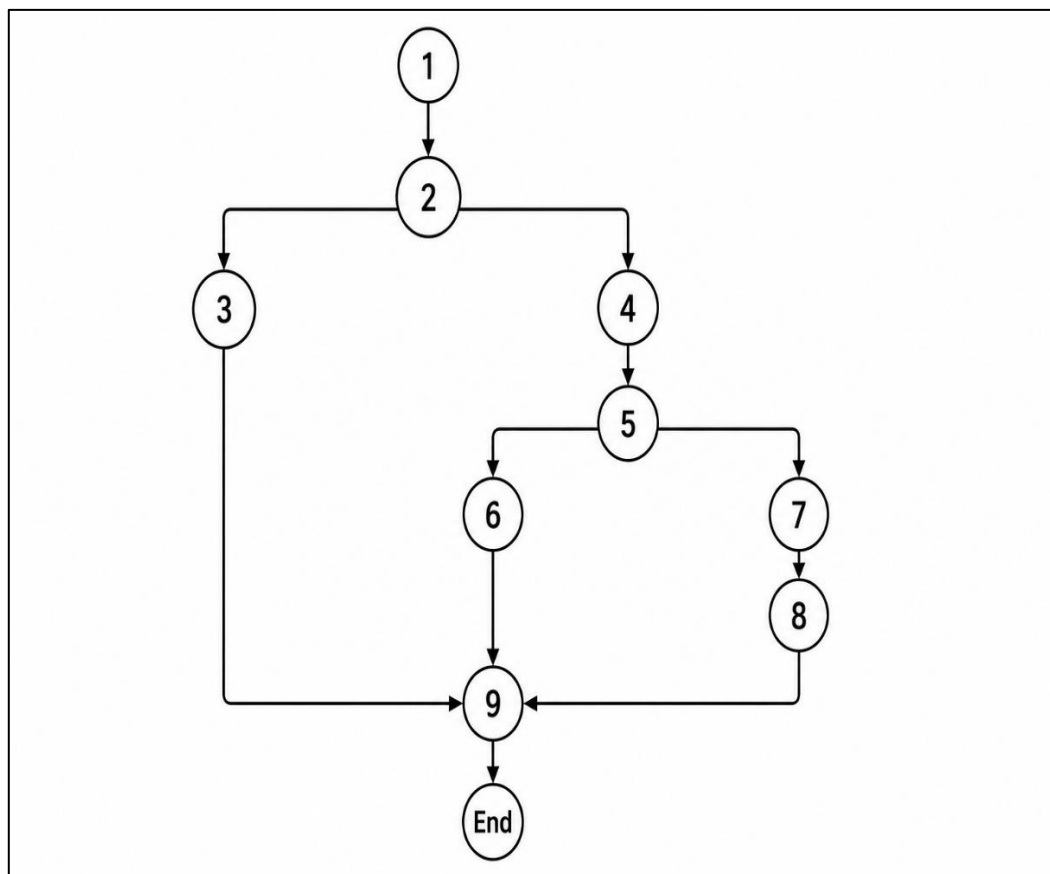
4. S6 dan S7 (Pengecekan Halaman Aktif)

Setelah proses hapus dilakukan, sistem memeriksa apakah halaman masih aktif. Jika halaman sudah tidak aktif, maka proses dihentikan.

5. S8 dan S9 (Hapus Berhasil)

Jika halaman masih aktif, sistem menutup dialog konfirmasi dan menampilkan pesan bahwa data karyawan berhasil dihapus.

d. *Flowgraph* Fungsi Hapus Data Karyawan



Sumber : data diolah oleh penulis 2026

Gambar 5. 52 *Flowgraph* Fungsi Hapus Data Karyawan

Berdasarkan gambar *flowgraph* fungsi hapus data karyawan, alur logika program dapat dijelaskan sebagai berikut. Proses dimulai pada *node* 1 ketika sistem menampilkan dialog konfirmasi penghapusan data karyawan. Selanjutnya pada *node* 2 sistem memeriksa pilihan pengguna pada dialog konfirmasi. Jika pengguna memilih Tidak, maka sistem menjalankan *node* 3 untuk menutup dialog konfirmasi dan membatalkan proses penghapusan data.

Jika pengguna memilih ya, maka sistem menjalankan *node* 4 untuk menghapus data karyawan dari *collection users* berdasarkan *docid*. Setelah proses penghapusan selesai, sistem menjalankan *node* 5 untuk memeriksa apakah halaman masih aktif (*mounted*). Apabila halaman sudah tidak aktif, sistem menjalankan *node* 6 untuk menghentikan proses. Namun apabila halaman masih aktif, sistem menjalankan *node* 7 untuk menutup dialog konfirmasi dan *node* 8 untuk

menampilkan pesan bahwa data karyawan berhasil dihapus. Setelah seluruh proses selesai dijalankan, alur berakhir pada *node* 9 sebagai *node* akhir (*end*).

f. Independent Path Fungsi Hapus Data Karyawan

Berdasarkan hasil perhitungan *cyclomatic complexity* fungsi hapus data karyawan, diperoleh nilai $V(G) = 3$. Nilai tersebut menunjukkan bahwa terdapat tiga jalur independen yang harus diuji untuk memastikan seluruh kemungkinan alur program telah dieksekusi. Adapun jalur independen pada fungsi hapus data karyawan adalah sebagai berikut:

1. Path 1

$1 \rightarrow 2 \rightarrow 3 \rightarrow 9$

Jalur ini terjadi ketika pengguna memilih tidak pada dialog konfirmasi sehingga proses penghapusan dibatalkan dan dialog ditutup.

2. Path 2

$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 9$

Jalur ini terjadi ketika pengguna memilih ya, data karyawan berhasil dihapus, tetapi halaman sudah tidak aktif (*mounted = false*) sehingga proses dihentikan.

3. Path 3

$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 9$

Jalur ini terjadi ketika pengguna memilih ya, data karyawan berhasil dihapus, halaman masih aktif (*mounted = true*), dialog konfirmasi ditutup, dan sistem menampilkan pesan "data karyawan berhasil dihapus". Dengan demikian, ketiga jalur independen tersebut telah mewakili seluruh kemungkinan alur utama pada fungsi hapus data karyawan dan akan digunakan sebagai dasar penyusunan *test case* pada tahap pengujian berikutnya.

g. Test Case Fungsi Hapus Data Karyawan

Pengujian fungsi hapus data karyawan dilakukan berdasarkan jalur independen yang telah diperoleh dari hasil analisis *flowgraph*. Setiap *test case* digunakan untuk menguji proses konfirmasi hapus, pembatalan hapus,

penghapusan data dari *firestore*, pengecekan kondisi halaman aktif, serta penampilan pesan berhasil.

Tabel 5. 18 *Test Case* Fungsi Hapus Data Karyawan

<i>Test Case</i>	Kondisi	Jalur Eksekusi	Output
TC1	Pengguna memilih Tidak pada dialog konfirmasi	1 → 2 → 3 → 9	Dialog ditutup dan proses hapus data dibatalkan
TC2	Pengguna memilih Ya, tetapi halaman sudah tidak aktif	1 → 2 → 4 → 5 → 6 → 9	Data terhapus, tetapi proses dihentikan tanpa menampilkan pesan
TC3	Pengguna memilih Ya dan halaman masih aktif	1 → 2 → 4 → 5 → 7 → 8 → 9	Data karyawan berhasil dihapus dan muncul pesan “Data karyawan berhasil dihapus”

Sumber : data diolah oleh penulis 2026

h. *Statement Coverage* Fungsi Hapus Data Karyawan

Statement coverage digunakan untuk mengukur sejauh mana seluruh *statement* pada fungsi hapus data karyawan telah dieksekusi melalui *test case* yang telah dirancang. Pengujian ini bertujuan untuk memastikan setiap *statement* pada proses konfirmasi hapus, pembatalan hapus, penghapusan data, pengecekan halaman aktif, dan penampilan pesan berhasil telah dijalankan minimal satu kali.

Statement yang tereksekusi pada masing-masing *test case* adalah sebagai berikut:

1. TC1:
S1, S2, S3, S9
2. TC2:
S1, S2, S4, S5, S6, S7, S9
3. TC3:
S1, S2, S4, S5, S6, S8, S9

Berdasarkan hasil pengujian, seluruh *statement* pada fungsi hapus data karyawan yaitu S1 sampai S9 berhasil dieksekusi melalui kombinasi *test case* TC1 sampai TC3. Dengan demikian diperoleh nilai *statement coverage* sebesar:

$$(9 / 9) \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *statement* pada fungsi hapus data karyawan telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses *white box testin*



BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil penelitian mengenai digitalisasi sistem informasi karyawan dan presensi menggunakan *qr code* berbasis *android* pada PT. Kendi Arindo, maka dapat disimpulkan sebagai berikut:

1. Perancangan Sistem Informasi Karyawan Berbasis *Android*

Penelitian ini berhasil merancang dan membangun sistem informasi karyawan berbasis aplikasi *android* yang mampu mengelola dan menampilkan data karyawan secara detail, terintegrasi, dan terpusat. Sistem yang dikembangkan menyediakan fitur pengelolaan data karyawan, detail informasi karyawan, serta pengelolaan akun pengguna sehingga dapat membantu proses administrasi kepegawaian secara digital menjadi lebih efektif dan efisien dibandingkan dengan metode sebelumnya.

2. Penerapan Sistem Presensi Menggunakan *QR Code*

Penelitian ini berhasil menerapkan sistem presensi berbasis pemindaian *qr code* pada aplikasi *android* sebagai pengganti sistem *fingerprint*. Sistem presensi yang dikembangkan dilengkapi dengan validasi *qr code*, pengambilan foto *selfie*, dan pencatatan lokasi *gps* sehingga dapat meningkatkan akurasi dan keamanan pencatatan kehadiran karyawan serta mengurangi potensi kecurangan, seperti titip absen. Selain itu, sistem juga mendukung pengajuan sakit, izin, dan cuti yang dapat diproses melalui mekanisme persetujuan oleh pihak yang berwenang.

3. Penyediaan Laporan Presensi dan Informasi Karyawan Secara *Real-Time*

Penelitian ini berhasil menyediakan laporan presensi dan informasi karyawan secara *real-time* melalui aplikasi *android* yang dapat diakses oleh pimpinan perusahaan. Fitur laporan yang tersedia memungkinkan pimpinan melakukan *monitoring* kehadiran karyawan, melihat rekap presensi, serta memperoleh informasi karyawan dengan lebih cepat dan akurat. Berdasarkan hasil pengujian menggunakan metode *white box testing*, seluruh fungsi utama aplikasi dapat berjalan sesuai dengan kebutuhan operasional PT. Kendi Arindo sehingga mampu mendukung proses evaluasi kehadiran dan pengambilan keputusan manajemen.

6.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, maka saran yang dapat diberikan adalah sebagai berikut:

1. PT. Kendi Arindo disarankan untuk menerapkan aplikasi digitalisasi sistem informasi karyawan dan presensi menggunakan *qr code* berbasis *android* secara berkelanjutan guna mendukung pengelolaan data karyawan dan presensi yang lebih efektif, efisien, serta terintegrasi. Dengan penerapan sistem ini, proses administrasi kepegawaian dan *monitoring* kehadiran karyawan dapat dilakukan secara lebih cepat dan akurat.
2. Perlu dilakukan pemeliharaan dan pengembangan sistem secara berkala agar aplikasi tetap berjalan dengan baik serta mampu menyesuaikan kebutuhan operasional perusahaan yang terus berkembang. Pemeliharaan meliputi pembaruan sistem, peningkatan keamanan data, perbaikan *bug*, serta optimalisasi performa aplikasi untuk menjaga kualitas layanan kepada pengguna.
3. Penelitian selanjutnya diharapkan dapat mengembangkan aplikasi dengan menambahkan fitur yang lebih lengkap, seperti notifikasi otomatis pengajuan sakit, izin, dan cuti, integrasi tanda tangan digital untuk proses persetujuan, *dashboard* analitik kehadiran karyawan, serta fitur penggajian yang terintegrasi dengan data presensi karyawan. Dengan adanya fitur tersebut, sistem tidak hanya berfungsi sebagai sarana pengelolaan informasi karyawan dan presensi, tetapi juga dapat mendukung proses administrasi kepegawaian secara lebih komprehensif dan mendukung pengambilan keputusan oleh manajemen perusahaan.

DAFTAR PUSTAKA

- Aidah, Arifudin, O., & Ibrahim, T. (2024). Pengembangan Sistem Informasi Manajemen Dalam Dunia Pendidikan. *Jurnal Tahsinia*, 5(6), 966–977. [Http://Jurnal.Rakeyansantang.Ac.Id/Tahsinia/Article/View/604](http://Jurnal.Rakeyansantang.Ac.Id/Tahsinia/Article/View/604)
- Akbar, M. F. (2022). Penerapan Metode Waterfall Pada Sistem Informasi Penjualan Dan Persediaan Pada Warung Makan Hejo Karawang. *Journal Computer Science*, 2(1), 29–34. [Https://Jurnal.Bsi.Ac.Id/Index.Php/Ijcs/Article/View/1902](https://Jurnal.Bsi.Ac.Id/Index.Php/Ijcs/Article/View/1902)
- Aldi, M., & Aji, M. I. (2025). Perancangan Sistem Digital Surat Masuk Dan Surat Keluar Di Uptd Kesmavet Berbasis Web. *Jurnal Multidisiplin Ilmu Akademik*, 2(2), 65–77. <https://doi.org/10.61722/Jmia.V2i2.4145>
- Al-Faruq, M. N. M., Nur'aini, S., & Aufan, M. H. (2022). Perancangan Ui/Ux Semarang Virtual Tourism Dengan Figma. *Walisongo Journal Of Information Technology*, 4(1), 43–52. <https://doi.org/10.21580/Wjit.2022.4.1.12079>
- Ansori, S., & Hendradi, P. (2023). Penerapan Metode Design Thinking Dalam Perancangan Ui / Ux Aplikasi Mobile Sipropmawa. *Journal Of Information System Research (Josh)*, 4(4). <https://doi.org/10.47065/Josh.V4i4.3648>
- Arianti, T., Fa'izi, A., Adam, S., Wulandari, M., & Aisyiyah Pontianak, P. '. (2022). Perancangan Sistem Informasi Perpustakaan Menggunakan Diagram Uml (Unified Modelling Language). *Jurnal Ilmiah Komputer Terapan Dan Informasi*, 1(1), 19–25. <https://download.garuda.kemdikbud.go.id/article.php?article=3053203&val=27786&title=Perancangan>
- Bainar, B. (2024). Peluang Dan Tantangan Digitalisasi Bagi Pendidikan Agama Islam. *Baitul Hikmah: Jurnal Ilmiah Keislaman*, 2(2), 74–80. https://doi.org/10.46781/Baitul_Hikmah.V2i2.1092
- Dalimunthe, H. F., & Simanjuntak, P. (2023). Aplikasi Pengenalan Perangkat Keras Komputer Berbasis *Android* Menggunakan Augmented Reality. *Jurnal Comasie*, 9, 259–266. [Http://ejournal.upbatam.ac.id/index.php/comasiejournal](http://ejournal.upbatam.ac.id/index.php/comasiejournal)
- Daruhadi, G., & Sopiati, P. (2024). Pengumpulan Data Penelitian. *J-Ceki : Jurnal Cendekia Ilmiah*, 5, 5423–5443. <https://ulilalbabinstitute.co.id/index.php/J-Ceki/Article/View/5181>
- Edriati, S., Husnita, L., Amri, E., Samudra, A. A., Kamil, N., Pgri, S., & Barat, S. (2021). Penggunaan Mit *App Inventor* Untuk Merancang Aplikasi Pembelajaran Berbasis *Android*. *Jurnal Pengabdian Kepada Masyarakat*, 12(4), 652–657. [Http://journal.upgris.ac.id/index.php/E-Dimas](http://journal.upgris.ac.id/index.php/E-Dimas)

- Erzed, N., Anwar, N., Widodo, A. M., Prasetyo, E., & Juman, K. K. (2022). Implementasi Flutter Pada Aplikasi Presensi Karyawan Berbasis Mobile. *Jurnal Ikraith-Informatika*, 6, 100–106. <https://journals.upi-yai.ac.id/index.php/ikraith-informatika/issue/archive>
- Firlando, J., Franko, B., Pratama Tanzil, S., Wilyanto, N., Christianto Tan, H., & Hartati Kom, E. M. (2023). Pembuatan Website Menggunakan Visual Studio Code Di Sma Xaverius 3 Palembang. *Jurnal Pengabdian Kepada Masyarakat Fordicate*, 3(1), 2023. <https://jurnal.mdp.ac.id/index.php/fordicate/article/view/5057>
- Fitriani, S. N., & Puspitasari, T. (2024). Sistem Informasi Presensi Guru Berbasis QR-Code Pada Sman 1 Waylima. *Journal Of Ethics And Character Education Jece*, 2(1), 14–31. <https://journal.yhmm.or.id/index.php/jece>
- Gunawan, A. (2024). Mobile Programming Menggunakan Flutter Dan Visual Studio Code Untuk Pemula. Pt. Literasi Nusantara Abadi Grup. www.penerbitlitnus.co.id
- Habsy, B. A., Mufidha, N., Shelomita, C., Rahayu, I., & Muckorobin, Moch. I. (2023). Filsafat Dasar Dalam Konseling Psikoanalisis: Studi Literatur. *Indonesian Journal Of Educational Counseling*, 7(2), 189–199. <https://doi.org/10.30653/001.202372.266>
- Hafizhah, N., & Hidayat, A. T. (2025). Optimalisasi Pengembangan Sistem Presensi Karyawan Menggunakan Extreme Programming Dan Teknologi QR Code. *Jurnal Janitra Informatika Dan Sistem Informasi*, 5(1), 1–13. <https://doi.org/10.59395/X78mnf30>
- Hasibuan, M. I. Z., & Triase. (2022). Implementasi Sistem Database Nosql Secara Realtime Menggunakan Firebase Realtime Database Pada Aplikasi Ourticle. *Sibatik Journal: Jurnal Ilmiah Bidang Sosial, Ekonomi, Budaya, Teknologi, Dan Pendidikan*, 2(1), 1–24. <https://doi.org/10.54443/Sibatik.V2i1.489>
- Hasibuan, M. P., Azmi, R., Arjuna, D. B., & Rahayu, S. U. (2023). Analisis Pengukuran Temperatur Udara Dengan Metode Observasi Analysis Of Air Temperature Measurements Using The Observational Method. *Gabdimas: Jurnal Garuda Pengabdian Kepada Masyarakat*, 1, 8–15. <http://creativecommons.org/licenses/by-sa/4.0/>
- Hidayatullah, H. (2023). Strategi Pembelajaran Pendidikan Agama Islam Era Digitalisasi Di Smp Sultan Agung Seyegan Sleman Yogyakarta. *Addabana: Jurnal Pendidikan Agama Islam*, 6(2), 119–133. <https://doi.org/10.47732/Adb.V6i2.249>
- Hutagaol, A., Damanik, D. B. N., Saragih, J. R. S., Wijaya, M. F., Sitompul, R. S. M., Sugara, W. H., & Pratama, L. S. (2024). Analisis Pengaruh Digitalisasi Terhadap Umkm Di Kota Medan. *Jurnal Ekuilnomi*, 6(3), 729–738. <https://doi.org/10.36985/Kgvmq881>

- Irfankha, R. (2025). Rancang Bangun Aplikasi Data Prestasi Mahasiswa Dengan Flutter. *Jurnal Gemilang Informatika*, 3, 45–50. <https://Journal.Hdgi.Org/Index.Php/Git/Article/Download/30/33>
- Irmaliantie, D., Hamid, A., Naury, C., Mandiri, S., & Selatan, K. (2024). Sistem Informasi Presensi Berbasis *Android* Menggunakan *QR Code* Pada Kantor Desa Pandawan. *Indonesian Journal Of Information Technology And Computing*, 4(1), 210–222. <https://Journal.Polhas.Ac.Id/Index.Php/Imaging>
- Kadek Merita, I Gede Margunayasa, & Kadek Yudiana. (2024). Media Pembelajaran Pop-Up-Book Berbantuan *QR Code* Berbasis Video Animasi Bagi Siswa Kelas Iv Sekolah Dasar. *Jurnal Mimbar Pendidikan Indonesia*, 5(1), 50–57. <https://Doi.Org/10.23887/Mpi.V5i1.81257>
- Khaledi, M. T. Al, Nasri, & Hanafi. (2022). Rancang Bangun Sistem Rumah Pintar Menggunakan Platform Google *Firebase* Berbasis Iot (Internet Of Things). *194 Jurnal Tektro*, 6(02), 194–202. <https://Doi.Org/Http://Dx.Doi.Org/10.30811/Tektro.V6i2.3732>
- Lantana, D. A., & Fauziah. (2025). *E-Book Buku Design Thinking Dan Inovasi*. Lembaga Penerbitan Universitas Nasional (Lpu Unas). <https://Repository.Unas.Ac.Id/Id/Eprint/14070/1/E-Book%20buku%20design%20thinking%20dan%20inovasi.Pdf>
- Lee, F. S., Nurprihatin, F., Honni, H., & Santoso, A. P. (2024). Aplikasi Absensi Cleaning Service Dengan Metode *Model Application Development Life Cycle*. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 6(3), 435–442. <https://Doi.Org/10.47233/Jteksis.V6i3.1386>
- Manoppo, S. F., Kaparang, D. R., & Liando, O. E. S. (2025). *Pengembangan Aplikasi Presensi Kehadiran Dosen Di Fakultas Teknik Universitas Negeri Manado*. *Journal Of Education Method And Technology*, 3, 76–82. <http://Calamus.Id/Index.Php/Jemtech/Article/View/149>
- Muhammad Ibnu Rosikhin, & Farhanna Mar'i. (2024). Pengembangan Sistem Presensi Guru Upt Smpn 34 Gresik Berbasis Web. *Jupiter: Publikasi Ilmu Keteknikan Industri, Teknik Elektro Dan Informatika*, 2(1), 114–133. <https://Doi.Org/10.61132/Jupiter.V2i1.59>
- Nashrulloh, R. Y., Raihan, M., Ghifari, A. H., & Ilham, F. (2025). Perancangan Sistem Penerimaan Peserta Didik Baru Berbasis Website Menggunakan Metode Waterfall Pada Ra Al Madani. *Jriin : Jurnal Riset Informatika Dan Inovasi*, 822–830. <https://Jurnalmahasiswa.Com/Index.Php/Jriin>
- Nelly Sofi, & Riza Dharmawan. (2022). Perancangan Aplikasi Bengkel Csm Berbasis *Android* Menggunakan Framework Flutter (Bahasa Dart). *Jurnal Teknik Dan Science*, 1(2), 53–64. <https://Doi.Org/10.56127/Jts.V1i2.125>

- Nuryana, M. L., Ibrahim, T., & Arifudin, O. (2024). Implementasi Dan Transformasi Sistem Informasi Manajemen Di Era Digital. *Jurnal Tahsinia*, 5(9), 1325–1337. [Http://Jurnal.Rakeyansantang.Ac.Id/Tahsinia/Article/View/614](http://Jurnal.Rakeyansantang.Ac.Id/Tahsinia/Article/View/614)
- Pratama, E. B., Hendini, A., & Fristian, A. (2023). Pendekatan Metode Prototype Pada Aplikasi Presensi Berbasis Mobile (Studi Kasus: Kantor Desa Mekar Jaya). *Jurnal Sistem Informasi Akuntansi*, 33(1), 33–39. [Http://Jurnal.Bsi.Ac.Id/Index.Php/Justian](http://Jurnal.Bsi.Ac.Id/Index.Php/Justian)
- Putra, M. Y., & Kurniawan, D. E. (2023). Implementasi Sistem Reminder Jadwal Pada Elearning Moodle Berbasis Api Menggunakan Framework Flutter. *Journal Of Applied Computer Science And Technology*, 4(1), 7–11. [Https://Doi.Org/10.52158/Jacost.V4i1.490](https://Doi.Org/10.52158/Jacost.V4i1.490)
- Putri, V. Q., & Pratiwi, M. P. (2021). Perancangan Sistem Informasi Inventaris Sekolah Berbasis Website Menggunakan Metode Prototype Pada Smp Negeri 24 Palembang. *Jtis(Journal Of Technopreneurship And Information System)*, 4(2), 7–11. [Https://D1wqtxts1xzle7.Cloudfront.Net/84083186/Jurnal_Umb_2c_Journal_Manager_2c_2._Vidya_Qoriah_Ok.Pdf_Filename_Utf-8jurnal_Umb_2c_Journal_Manager_2c_2._Vidya_Qoriah_Ok-Libre.Pdf](https://D1wqtxts1xzle7.Cloudfront.Net/84083186/Jurnal_Umb_2c_Journal_Manager_2c_2._Vidya_Qoriah_Ok.Pdf_Filename_Utf-8jurnal_Umb_2c_Journal_Manager_2c_2._Vidya_Qoriah_Ok-Libre.Pdf)
- Qois, N., & Jumaryadi, Y. (2021). Implementasi Location Based Service Pada Sistem Informasi Kehadiran Pegawai Berbasis *Android*. *Sistemasi : Jurnal Sistem Informasi*, 10(3), 550. [Https://Doi.Org/10.32520/Stmsi.V10i3.1369](https://Doi.Org/10.32520/Stmsi.V10i3.1369)
- Raisa, A., Razi, Z., & Khalid, Z. (2024). Aplikasi Penjadwalan Kuliah Berbasis *Android* Pada Fakultas Teknik Dengan Menggunakan Bahasa Pemograman Dart *Android-Based Lecture Scheduling Application At The Faculty Of Engineering Using The Dart Programming Language*. *Jurnal Teksagro*, 5(1), 24–29. [Https://Journal.Lp2stm.Or.Id/Index.Php/Teksagro/Article/View/92](https://Journal.Lp2stm.Or.Id/Index.Php/Teksagro/Article/View/92)
- Rifaldy, Sulaeman B, Suppa R. (2022). Sistem Informasi Pelelangan Di Pegadaian Cabang Malili Berbasis *Android*. *Indonesian Journal Of Education And Humanity*, 2, 59–72. <http://ijohm.rcipublisher.org/index.php/ijohm/article/view/48>
- Riswanto, B., & Fahrudin, I. (2024). Perancangan Aplikasi Kasir Aira Motor Berbasis *Android* Menggunakan *Firebase Realtime Database*. *Digital Transformation Technology (Digitech)*, 4(2), 776–784. [Https://Doi.Org/10.47709/Digitech.V4i2.4593](https://Doi.Org/10.47709/Digitech.V4i2.4593)
- Rizaldhi, B. F., & Fadli, U. M. (2023). Analisis Budaya Kerja Karyawan Pt. Indonesia Thai Summit Auto. *Jurnal Ilmiah Multidisiplin*, 1(5), 778–787. [Https://Doi.Org/10.5281/Zenodo.8050247](https://Doi.Org/10.5281/Zenodo.8050247)
- Rohmat, C. L., & Gusti, P. P. (2023). Rancang Bangun Aplikasi Pembukuan Dokumen Pemakaman Desa Kalikoa Berbasis *Android*. *Jurnal Mahasiswa*

Teknik Informatika, 7(1), 712–719.
<https://Ejournal.Itn.Ac.Id/Index.Php/Jati/Article/View/6498>

- Romdona, S., Senja Junista, S., & Gunawan, A. (2025). Teknik Pengumpulan Data: Observasi, Wawancara Dan Kuesioner. *Jisosepol: Jurnal Ilmu Sosial Ekonomi Dan Politik*, 3(1), 39–47. <https://Samudrapublisher.Com/Index.Php/Jisosepol>
- Saádi Ahmad. (2025). Pengumpulan Data Yang Efisien Pada Penelitian Tindakan Kelas: Teknik, Alat, Dan Tantangan. *Al-Amin: Jurnal Ilmu Pendidikan Dan Sosial Humaniora*, 2, 90–108.
<https://Ejournal.Alkifayahriau.Ac.Id/Index.Php/Alamin/Article/View/377>
- Sahir, S. H. (2022). Metodologi Penelitian. Penerbit Kbm Indonesia. www.Penerbitbukumurah.Com
- Sahputra, M. A., Defriani, M., & Hermanto, T. I. (2023). Rancan Bangun Aplikasi Pelayanan E-Trayek Berbasis Mobile Menggunakan Metode Extreme Programming. *Sudo Jurnal Teknik Informatika*, 2(1), 34–44.
<https://Doi.Org/10.56211/Sudo.V2i1.229>
- Saputra, S. W. (2021). Rancang Bangun Aplikasi Mobile Learning Tenses Bahasa Inggris. *Jurnal Teknologi Informasi Jurnal Keilmuan Dan Aplikasi Bidang Teknik Informatika*, 13(2), 37–46. <https://E-Journal.Upr.Ac.Id/Index.Php/Jti/Article/View/253>
- Setiawan, P. R., Syefriani, & Wandri, R. (2025). Inovasi Teknologi Melalui Pembelajaran Flutter: Menyongsong Era Aplikasi Mobile. *Jurnal Sains Dan Teknologi Dalam Pengabdian Masyarakat*, 2, 1–9. <https://Jurnal-Rjb.Com/Index.Php/Jstpm/Article/View/29>
- Shobur, A. F., & Satrio, A. V. (2025). Perancangan Sistem Informasi Absensi Pada Sogogi Shabu Dan Grill Jambi Berbasis Mobile. *Jutekom Jurnal Teknologi Dan Ilmu Komputer*, 1, 17–23. <https://Doi.Org/10.35134/Jutekom.V9i2.1>
- Solihin, I. P., Adli, R. (2023). Sistem Absensi Mahasiswa Menggunakan QR Code Dengan Metode Zxing Berbasis Android. *Jurnal Ilmiah Informatika Dan Komputer*, 7, 1–7. <https://Jurnalteknik.Unkris.Ac.Id/>
- Solissa, Y J., Putra, A N., Nursari, S R C. (2023). Pengujian White-Box Berbasis Path Pada Form Daftar Jobstreet.co.id. *Konstelasi: Konvergensi Teknologi Dan Sistem Informasi*, 3, 353–362.
https://ojs.uajy.ac.id/index.php/konstelasi/article/download/8287/3374/26230?utm_source=chatgpt.com
- Sudradjat, B. (2021). Penggunaan Teknologi Flutter Dalam Aplikasi Mobile Untuk Pengembangan Kedai Kopi. *Riset Dan E-Jurnal Manajemen Informatika Komputer (Remik)*, 6(1), 1–8. <https://Doi.Org/10.33395/Remik.V6i1.11123>
- Surya Ningsih, K., Jamilah Aruan, N., & Taufik Al Afkari Siahaan, A. (2022). Yayasan Insan Cipta Medan Aplikasi Buku Tamu Menggunakan Fitur Kamera

- Dan Ajax Berbasis Website Pada Kantor Dispora Kota Medan. *Sitek: Jurnal Sains, Informatika, Dan Teknologi*, 1, 94–99. <https://Jurnal.Insanciptamedan.Or.Id/Index.Php/Sitek/Article/View/75>
- Sutikno, Y., Hosan, H., & Irawati, I. (2021). Implementasi Metode Penugasan Untuk Meningkatkan Kemampuan Melakukan Wawancara Untuk Mahasiswa Stab Maitreyawira. *Jurnal Maitreyawira*, 2(2), 1–8. <https://doi.org/10.69607/Jm.V2i2.40>
- Syahputri Addini Zahra. (2023). Kerangka Berfikir Penelitian Kuantitatif. *Tarbiyah: Jurnal Ilmu Pendidikan Dan Pengajaran*, 6, 160–166. <https://Jurnal.Diklinko.Id/Index.Php/Tarbiyah/https://Jurnal.Diklinko.Id/Index.Php/Tarbiyah/>
- Syam, M. L., & Erdisna. (2022). Sistem Informasi Stok Barang Menggunakan QR-Code Berbasis Android. *Jurnal Informatika Ekonomi Bisnis*, 4, 17–22. <https://doi.org/10.37034/InfEb.V4i1.108>
- Usman, Dr. E., Nafisa, L., & Nainggolan, Y. T. (2022). Buku Sistem Informasi Akuntansi (Dr. M. A. Wardana, Ed.). Cv. Intelektual Manifes Media. <https://www.infesmedia.co.id/>
- Wijaya, A. M., Samodra, J. E., & Suyoto. (2023). Sistem Presensi Pegawai Dengan Face Recognition Menggunakan Deep Learning Cnn 163 Sistem Presensi Pegawai Dengan Face Recognition Menggunakan Deep Learning Cnn. *Jurnal Informatika Atma Jogja*, 4, 163–168. <https://ojs.uajy.ac.id/index.php/Jiaj/article/view/7660>
- Yunaningsih, A., Indah, D., & Septiawan, F. E. (2021). Upaya Meningkatkan Kualitas Layanan Publik Melalui Digitalisasi. *Jurnal Altasia*, 3(1), 9–16. <https://scholar.archive.org/work/Xpuqix4l75a2jauri4i4uoj2p4/access/wayback>
- Zen, H. R. R., & Nuryasin, I. (2024). Joisie Licensed Under A Creative Commons Attribution-Sharealike 4.0 International License (Cc By-Sa 4.0) Penerapan Whitebox Testing Pada Pengujian Sistem Menggunakan Teknik Basis Path. *Journal Of Information Systems And Informatics Engineering*, 8(1), 101–111. <https://doi.org/10.35145/Joisie.V8i1.4229>
- Zulkipli. (2022). Perencanaan Manajemen Sumber Daya Manusia. *Jurnal Visionary : Penelitian Dan Pengembangan Dibiidang Administrasi Pendidikan*, 10, 57–66. <https://e-journal.undikma.ac.id/index.php/visionary>

LAMPIRAN

Lampiran 1 Surat Permohonan Penelitian


**FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH**

Prabumulih, 09 Oktober 2025

Nomor : 017 /FASILKOM/UNPRA/Inf-S1/X/2025
Lampiran : -
Perihal : Permohonan Pengambilan Data

Kepada Yth,
Pimpinan PT. KENDI ARINDO
Di
Tempat

Dalam rangka pelaksanaan skripsi mahasiswa Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih, maka kami harapkan bantuan dari bapak/ ibu kiranya mengizinkan mahasiswa:

No.	Nama	NIM	Program Studi
1.	Dimas Kurnia	2022230026	Informatika
2.	Rahmat Akbar Hidayat	2022230034	
3.	Aji Muksen	2022230033	

Untuk melaksanakan pengambilan data ditempat Bapak/Ibu Pimpin guna menyelesaikan skripsi. Pengambilan data tersebut akan dilaksanakan pada tanggal 22 Oktober 2025 s/d 28 Oktober 2025. Atas bantuan dan kerjasamanya kami ucapkan terima kasih.

Ketua Program Studi,
Informatika,



Iwan Setiawan, S.Kom., M.Kom.
NIDN.0209129101

Lampiran 2 Surat Balasan Permohonan Penelitian

		PT. Kendi Arindo (Perkebunan) Jl. Lingkar Talang Gunung Kelurahan Jaya Loka Kecamatan Tebing Tinggi, Kabupaten Empat Lawang – Sumatera Selatan.
Nomor	: 111-107/PT.KA/KC/X/2025	Tebing Tinggi, 10 Oktober 2025
Lampiran	: -	Kepada Yth,
Perihal	: Persetujuan Pengambilan Data	Dekan Fakultas Ilmu Komputer Universitas Prabumulih
		Di – Tempat.

Dengan hormat,

Menindaklanjuti surat dari Fakultas Ilmu Komputer Universitas Prabumulih Nomor : 017/FASILKOM/UNPRA/Inf-SI/X/2025 tanggal 09 Oktober 2025 perihal permohonan pengambilan data untuk keperluan tugas akhir mahasiswa, kami sampaikan bahwa PT.Kendi Arindo menyambut baik dan memberikan izin kepada mahasiswa untuk melakukan pengambilan data di lingkungan perusahaan kami.

Kami berharap kegiatan ini dapat memberikan kontribusi positif bagi pengembangan ilmu pengetahuan serta mendukung penyelesaian tugas akhir mahasiswa yang bersangkutan.

Untuk kelancaran proses pengambilan data kami harapkan agar mahasiswa dapat berkoordinasi terlebih dahulu dengan Bagian Umum dan Administrasi PT. Kendi Arindo yaitu Bpk. Agus Sutikno,SE (0852-4636-3991) dan mematuhi ketentuan serta protokol yang berlaku di perusahaan.

Demikian surat ini kami sampaikan. Atas perhatian dan kerja sama yang baik kami ucapkan terima kasih.

Hormat kami,


Ir. Ismantri Sukarno
Kepala Cabang

Lampiran 3 Dokumentasi Wawancara Dan Observasi



Lampiran 4 Daftar Pertanyaan Wawancara

DRAFT WAWANCARA 1

Nama Narasumber : Agus Sutikno, S.E.

Tempat Wawancara : Kantor PT Kendi Arindo Empat Lawang

Tanggal Wawancara : 22 Oktober 2025

Pewawancara : Aji Muksen

No	Pertanyaan	Jawaban
1	Dapatkah Bapak menjelaskan secara singkat mengenai profil PT Kendi Arindo dan kegiatan operasional utamanya?	PT Kendi Arindo merupakan perusahaan yang bergerak di bidang perkebunan karet. Kegiatan utamanya meliputi pengelolaan lahan, pemeliharaan tanaman, penyadapan, hingga pengolahan dan penyaluran hasil produksi karet.
2	Bagaimana struktur organisasi secara umum, khususnya yang berkaitan dengan pengelolaan karyawan di PT Kendi Arindo?	Struktur organisasi kami terdiri dari pimpinan perusahaan, kepala bagian, dan beberapa subbagian, termasuk bagian umum dan administrasi yang mengelola data kepegawaian, presensi, serta administrasi pendukung operasional.
3	Apa saja tugas dan tanggung jawab Bapak sebagai Kepala Bagian Umum dan Administrasi terkait pengelolaan karyawan dan presensi?	Kami bertanggung jawab mengelola data karyawan, mengatur jadwal kerja, memeriksa laporan presensi, serta memastikan administrasi kepegawaian berjalan tertib dan sesuai prosedur perusahaan.
4	Sistem presensi apa yang saat ini digunakan di PT Kendi Arindo, dan bagaimana alur pencatatan kehadiran karyawan?	Saat ini kami masih menggunakan sistem presensi <i>fingerprint</i> . Karyawan melakukan <i>scan</i> sidik jari saat datang dan pulang kerja, kemudian data direkap secara berkala oleh bagian administrasi.
5	Menurut Bapak, apa saja kelemahan atau kendala yang dirasakan dari penggunaan sistem <i>fingerprint</i> tersebut?	Beberapa kendala yang kami rasakan antara lain potensi titip absen, alat yang kadang bermasalah, serta laporan presensi yang belum dapat dilihat secara detail dan <i>real-time</i> oleh pimpinan.
6	Apakah dalam proses pengelolaan presensi dan data karyawan pernah terjadi keterlambatan laporan atau kesalahan pencatatan?	Ya, kadang terjadi keterlambatan rekap bulanan dan kemungkinan kesalahan <i>input</i> ketika data dipindahkan ke laporan manual, terutama saat jumlah karyawan yang direkap cukup banyak.
7	Data apa saja yang biasanya dibutuhkan pimpinan terkait kehadiran karyawan untuk keperluan evaluasi dan pengambilan keputusan?	Pimpinan biasanya membutuhkan data jumlah hari hadir, izin, sakit, cuti, dan hari <i>off</i> setiap karyawan dalam satu periode, serta rekap kehadiran per bagian untuk bahan evaluasi kinerja.

8	Berapa jumlah data karyawan yang saat ini dikelola oleh PT Kendi Arindo dan bagaimana klasifikasinya?	Berdasarkan data terakhir, jumlah karyawan yang dikelola sebanyak 46 orang, yang terdiri dari 45 karyawan laki-laki dan 1 karyawan perempuan. Dari status kepegawaian, terdiri atas 40 karyawan tetap, 6 karyawan kontrak.
9	Sejauh ini, apakah pimpinan dapat mengakses laporan presensi secara cepat dan komprehensif, atau masih memerlukan proses manual?	Saat ini masih memerlukan proses manual. Laporan harus disiapkan dulu oleh bagian administrasi, sehingga pimpinan belum bisa melihat data presensi secara langsung dan <i>real-time</i> .
10	Menurut Bapak, seberapa besar kebutuhan PT Kendi Arindo terhadap sistem informasi karyawan dan presensi yang terkomputerisasi secara terpadu?	Menurut kami kebutuhannya cukup besar, karena jumlah karyawan dan data yang dikelola terus meningkat, sehingga diperlukan sistem yang lebih terintegrasi dan efisien.
11	Bagaimana pandangan Bapak terkait rencana pengembangan sistem informasi karyawan dan presensi menggunakan <i>QR Code</i> berbasis <i>Android</i> ?	Kami menyambut baik rencana tersebut. Jika sistemnya mudah digunakan dan sesuai dengan kebutuhan di lapangan, kami yakin dapat membantu mempercepat pengolahan data kehadiran dan mengurangi kesalahan.
12	Menurut Bapak, fitur apa saja yang paling penting untuk ada dalam aplikasi digitalisasi sistem informasi karyawan dan presensi tersebut?	Fitur penting antara lain data profil karyawan, presensi menggunakan <i>QR Code</i> , rekap kehadiran otomatis, laporan bulanan, dan akses laporan bagi pimpinan secara <i>real-time</i> .
13	Apakah menurut Bapak penggunaan aplikasi <i>Android</i> akan memudahkan bagian administrasi dalam pengelolaan data karyawan dan presensi?	Iya, karena data dapat langsung tersimpan dalam <i>database</i> , proses rekap menjadi lebih cepat, dan pencarian informasi presensi maupun data karyawan bisa dilakukan dengan lebih mudah.
14	Manfaat apa yang Bapak harapkan bagi pimpinan dan perusahaan jika sistem digitalisasi ini sudah diterapkan?	Kami berharap pimpinan bisa memperoleh laporan presensi yang lebih lengkap, akurat, dan <i>real-time</i> , sehingga proses evaluasi kinerja dan pengambilan keputusan terkait kepegawaian menjadi lebih tepat.
15	Apakah Bapak bersedia jika nantinya dilakukan uji coba aplikasi yang dikembangkan peneliti bersama bagian umum dan administrasi?	Insyallah bersedia, selama uji coba dilakukan dengan penjelasan yang jelas, tidak mengganggu aktivitas kerja utama, dan hasilnya dapat memberi manfaat bagi perusahaan.

Menyetujui,

Narasumber



Agus Sutikno, S.E.

Kepala Bagian Umum dan Administrasi

Empat Lawang, 22 Oktober 2025

Pewawancara

Aji Muksen

DRAFT WAWANCARA 2

Nama Narasumber : Khoirul Mukminin

Tempat Wawancara : Kantor PT Kendi Arindo Empat lawang

Tanggal Wawancara : 24 Januari 2026

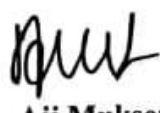
Pewawancara : Aji Muksen

No	Pertanyaan	Jawaban
1	Bagaimana sistem presensi yang saat ini anda gunakan di PT Kendi Arindo Empat Lawang?	Saat ini kami menggunakan sistem presensi <i>fingerprint</i> untuk mencatat kehadiran.
2	Apakah anda pernah mengalami kendala saat menggunakan presensi <i>fingerprint</i>	Ya, sering terjadi kendala seperti sidik jari tidak terbaca atau alat <i>error</i> .
3	Menurut anda, apakah sistem presensi <i>fingerprint</i> sudah praktis digunakan?	Cukup membantu, tetapi kurang praktis karena sering terjadi antrean dan gangguan teknis.
4	Apakah proses presensi <i>fingerprint</i> pernah menghambat aktivitas kerja Anda?	Pernah, terutama saat alat tidak berfungsi dengan baik sehingga proses presensi menjadi lama.
5	Bagaimana pendapat anda jika presensi dilakukan menggunakan aplikasi <i>android</i> berbasis <i>QR Code</i> ?	Menurut saya lebih praktis dan <i>modern</i> karena bisa dilakukan dengan cepat menggunakan ponsel.
6	Apakah anda merasa presensi <i>QR Code</i> lebih mudah dibandingkan <i>fingerprint</i> ?	Ya, karena hanya perlu memindai <i>QR Code</i> tanpa harus menggunakan alat tertentu.
7	Apakah penggunaan aplikasi <i>android</i> untuk presensi mudah dipahami oleh karyawan?	Mudah, karena hampir semua karyawan sudah terbiasa menggunakan <i>smartphone android</i> .
8	Menurut anda, apa keuntungan utama dari presensi berbasis <i>QR Code</i> ?	Proses lebih cepat, tidak perlu antre, dan lebih nyaman digunakan.
9	Apakah anda bersedia menggunakan sistem presensi berbasis <i>QR Code</i> jika diterapkan?	Bersedia, selama aplikasinya mudah digunakan dan tidak menyulitkan.
10	Apa harapan anda terhadap sistem presensi yang akan dikembangkan?	Sistem presensi yang stabil, cepat, dan memudahkan karyawan dalam mencatat kehadiran.

Menyetujui,
Narasumber


KHOIRUL MUKMININ
Karyawan


Empat Lawang, 24 Januari 2026
Pewawancara


Aji Muksen



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH**

**KESEDIAAN SEBAGAI PEMBIMBING SKRIPSI
PROGRAM STUDI INFORMATIKA**

Yang bertanda tangan di bawah ini:

Nama : Iwan Setiawan, S.Kom., M.Kom.
NUPTK : 1541769670130303
Program Studi : Informatika
Fakultas : Ilmu Komputer
Bersedia menjadi pembimbing I Skripsi, dan

Nama : Hermizahadiwidastra, S.Kom., M.Kom.
NUPTK : 2863762663130192
Program Studi : Informatika
Fakultas : Ilmu Komputer
Bersedia menjadi pembimbing II Skripsi

dari mahasiswa Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih
di bawah ini:

Nama : Aji Muksen
NIM : 2022230033
Judul Skripsi : Digitalisasi Sistem Informasi Karyawan dan Presensi
Menggunakan *QR Code* Berbasis *Android*

Demikian surat kesediaan membimbing ini kami buat, atas perhatiannya kami ucapkan terima kasih.

Dosen Pembimbing I

(Iwan Setiawan, S.Kom., M.Kom.)
NUPTK. 1541769670130303

Prabumulih, ...2026

Dosen Pembimbing II

(Hermizahadiwidastra, S.Kom., M.Kom.)
NUPTK. 2863762663130192



FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : Aji Muksen
NIM : 2022230033
PROGRAM STUDI : INFORMATIKA
NAMA PEMBIMBING I : IWAN SETIAWAN, S.Kom., M.Kom
JUDUL : DIGITALISASI SISTEM INFORMASI KARYAWAN DAN PRESENSI MENGGUNAKAN QR CODE BERBASIS ANDROID

NO	TGL. KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. I	TGL MENGHADAP KEMBALI
1	11-2-2026	BAB I ACC		
		BAB II a. Tabel		
		BAB III a. Diagram		
2.	24-2-2026	BAB II ACC		
		BAB III ACC		

Mulai Bimbingan : 11-2-2026
Batas Akhir Bimbingan :

Mengetahui Kaprodi Informatika



IWAN SETIAWAN, S.Kom., M.Kom

PERHATIAN !
TIDAK BOLEH HILANG
SETIAP BIMBINGAN HARUS
DIBAWA



FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : Aji Muksen
NIM : 2022230033
PROGRAM STUDI : INFORMATIKA
NAMA PEMBIMBING I : IWAN SETIAWAN, S.Kom., M.Kom
JUDUL : DIGITALISASI SISTEM INFORMASI KARYAWAN DAN PRESENSI MENGGUNAKAN QR CODE BERBASIS ANDROID

NO	TGL KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. I	TGL MENGHADAP KEMBALI
3.	14/4/2026	BAB IV a. perencanaan logis b. Database c. antarmuka		
4	8/5/2026	BAB IV a. Database b. UI / UX		
5	7/5/2026	BAB IV ACC		
6	26/5/2026	BAB V a. prongda kers b Bab IV data base c. detail Bab IV reka farswa karti krymen, Perjan		

Mulai Bimbingan :

Batas Akhir Bimbingan :



IWAN SETIAWAN, S.Kom., M.Kom

PERHATIAN !
TIDAK BOLEH HILANG
SETIAP BIMBINGAN HARUS
DIBAWA



FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : Aji Muksen
NIM : 2022230033
PROGRAM STUDI : INFORMATIKA
NAMA PEMBIMBING I : IWAN SETIAWAN, S.Kom., M.Kom
JUDUL : DIGITALISASI SISTEM INFORMASI KARYAWAN DAN PRESENSI MENGGUNAKAN QR CODE BERBASIS ANDROID

NO	TGL KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. I	TGL MENGHADAP KEMBALI
7	2/6/2026	BAB V ACC		
8	9/6/2026	BAB VI ACC		
		kelengkapan SIAP UJI		

Mulai Bimbingan :

Batas Akhir Bimbingan :

PERHATIAN !
TIDAK BOLEH HILANG
SETIAP BIMBINGAN HARUS
DIBAWA



Mengetahui Kaprodi Informatika

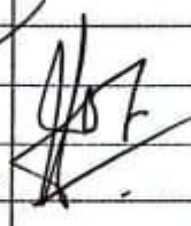
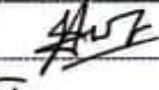
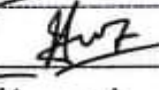
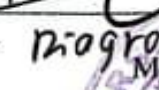
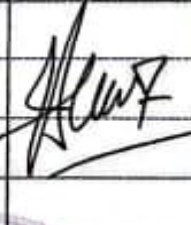
IWAN SETIAWAN, S.Kom., M.Kom



FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : AJI MUKSEN
NIM : 2022230033
PROGRAM STUDI : INFORMATIKA
NAMA PEMBIMBING II : HERMIZAHADIWIDASTRA, S.Kom., M.Kom
JUDUL : DIGITALISASI SISTEM INFORMASI KARYAWAN DAN PRESENSI MENGGUNAKAN QR CODE BERBASIS ANDROID

NO	TGL. KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. II	TGL MENGHADAP KEMBALI
1	11/02 2026	BAB I 1.5. Manfaat praktis Teori ts. ? BAB II tambahan narasi pada tabel penelitian tersebut. BAB III Sub reu & Abstrak Revisi. - tabel. Jadwal penelitian - Teori yg digunakan dan penelitian ini ? - Aktivitas diagram..		
2	11/03 2026	DCC BAB I  Lampir BAB II DCC BAB II  Lampir BAB III  BAB III 3.6.2 Program		

Mulai Bimbingan :

Batas Akhir Bimbingan :

PERHATIAN !
TIDAK BOLEH HILANG
SETIAP BIMBINGAN HARUS
DIBAWA

Mengetahui Kaprod Informatika

IWAN SETIAWAN, S.Kom., M.Kom



FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : AJI MUKSEN
NIM : 2022230033
PROGRAM STUDI : INFORMATIKA
NAMA PEMBIMBING II : HERMIZAHADIWIDASTRA, S.Kom., M.Kom
JUDUL : DIGITALISASI SISTEM INFORMASI KARYAWAN DAN PRESENSI MENGGUNAKAN QR CODE BERBASIS ANDROID

NO	TGL. KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. II	TGL MENGHADAP KEMBALI
3	8/04 2026	Dce Basis in Layout DASTU		
4	9/04 2026	- 4.2.1. - Use Case ? - Activity ? - Class ?		
5	19/05 2026	4.2.1. Use Case. Form 2.2.3. UI/UX Gambar - Format penulisan program Berhenti.		
6	26/05 2026	- DASTU - Use Case karyawan • MENU untuk & share presensi. • Login, Logout, Cuti		

Mulai Bimbingan :

Batas Akhir Bimbingan :

Mengetahui Kaprodi Informatika

IWAN SETIAWAN, S.Kom., M.Kom

PERHATIAN !
TIDAK BOLEH HILANG
SETIAP BIMBINGAN HARUS
DIBAWA



FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : AJI MUKSEN
NIM : 2022230033
PROGRAM STUDI : INFORMATIKA
NAMA PEMBIMBING II : HERMIZAHADIWIDASTRA, S.Kom., M.Kom
JUDUL : DIGITALISASI SISTEM INFORMASI KARYAWAN DAN PRESENSI MENGGUNAKAN QR CODE BERBASIS ANDROID

NO	TGL KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. II	TGL MENGHADAP KEMBALI
7	02/06/2026	BAB IV ACC ~~Hand~~ Lanjutan BAB V - format penulisan - - Ketersediaan orang - Sebaiknya lg detail gambar - Rapi tulis lg penulisan		
8	04/06/2026	BAB V - template halaman. - verifikasi presensi. (nama, password), (solusi, izin) RUM - Total letak menu karyawan, HRD, manajemen.		
9	09/06/2026	- ACC BAB V ~~Hand~~ Lanjutan BAB VI		

Mulai Bimbingan :

Batas Akhir Bimbingan :

Mengetahui Kaprodi Informatika

IWAN SETIAWAN, S.Kom., M.Kom

PERHATIAN !
TIDAK BOLEH HILANG
SETIAP BIMBINGAN HARUS
DIBAWA



FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : AJI MUKSEN
NIM : 2022230033
PROGRAM STUDI : INFORMATIKA
NAMA PEMBIMBING II : HERMIZAHADIWIDASTRA, S.Kom., M.Kom
JUDUL : DIGITALISASI SISTEM INFORMASI KARYAWAN DAN PRESENSI MENGGUNAKAN QR CODE BERBASIS ANDROID

NO	TGL. KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. II	TGL MENGHADAP KEMBALI
10	09/06 2021	- Ace Bob VI <i>[Signature]</i>	<i>[Signature]</i>	
		Stap Ygi <i>[Signature]</i>	<i>[Signature]</i>	

Mulai Bimbingan :
Batas Akhir Bimbingan :

Mengetahui Kaprodi Informatika



[Signature]
IWAN SETIAWAN, S.Kom., M.Kom

PERHATIAN I
TIDAK BOLEH HILANG
SETIAP BIMBINGAN HARUS
DIBAWA



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH**

PENILAIAN REVIEW SKRIPSI

Nama : Aji Muksen
NIM : 2022230033
Judul Skripsi : Digitalisasi Sistem Informasi Karyawan dan Presensi
Menggunakan *QR Code Berbasis Android*
Hari/Tanggal : JUMAT 19-6-2026
Saran dan masukan :
IKUTI ARAHAN PENGUJI DENGAN KONSULTASI
KE PEMBIMBING DULU

ACC SIAP JILID

16-7-2026

Rekomendasi*:

- a. Dilanjutkan tanpa perbaikan
- ☒ b. Dilanjutkan dengan perbaikan
- c. Ditolak

Prabumulih, 19-6-2026
Pembimbing I

(Iwan Setiawan, S.Kom., M.Kom.)
NUPTK. 1541769670130303

*pilih salah satu



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH**

PENILAIAN REVIEW SKRIPSI

Nama : Aji Muksen
NIM : 2022230033
Judul Skripsi : Digitalisasi Sistem Informasi Karyawan dan Presensi
Menggunakan QR Code Berbasis Android
Hari/Tanggal : Juni 19 2026
Saran dan masukan :

1. Kurang arahan dari Dosen penguji

Snap Joki 15/06 2026
#ask

Ree

Rekomendasi*:

- a. Dilanjutkan tanpa perbaikan
- ☒ b. Dilanjutkan dengan perbaikan
- c. Ditolak

Prabumulih, 19 Juni 2026
Pembimbing II

(Hernizahadiwidastra, S.Kom., M.Kom)
NUPTK. 2863762663130192

*pilih salah satu



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH**

PENILAIAN REVIEW SKRIPSI

Nama : Aji Muksen
NIM : 2022230033
Judul Skripsi : Digitalisasi Sistem Informasi Karyawan dan Presensi
Menggunakan *QR Code* Berbasis *Android*

Hari/Tanggal :

Saran dan masukan :

1. Spesifikasi kebutuhan end user (karyawan) bukti wawancara
2. filter laporan
3. kelayakan proses
4. -

Aji Muksen

Rekomendasi*:

- a. Dilanjutkan tanpa perbaikan
- b. Dilanjutkan dengan perbaikan
- c. Ditolak

Prabumulih, 21 - 6 - 2026
Dosen Penguji

(Suhartini, S.Kom., M.Kom)
NUPTK. 253475765823105

*pilih salah satu