

RANCANG BANGUN SISTEM INFORMASI KEPEGAWAIAN (PEREKRUTAN DAN MUTASI) PERUSAHAAN PENYEDIA JASA TEKNOLOGI INFORMASI (IT)

PROYEK AKHIR

Disusun Oleh:

Grey Ari Daniel Simatupang

3312311133

Disusun Untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Ahli Madya
Teknik Informatika



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

POLITEKNIK NEGERI BATAM

BATAM

2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul “RANCANG BANGUN SISTEM INFORMASI KEPEGAWAIAN (PEREKRUTAN DAN MUTASI) PERUSAHAAN PENYEDIA JASA TEKNOLOGI INFORMASI (IT)” dengan baik dan tepat waktu.

Laporan ini disusun sebagai salah satu syarat untuk memperoleh gelar Ahli Madya Teknik Informatika di Program Studi Teknik Informatika, Politeknik Negeri Batam. Laporan ini membahas tentang merancang dan membangun sistem informasi manajemen karyawan berbasis mobile dan web yang mencakup fitur absensi, penugasan, cuti, lembur, penggajian, reset massal, log aktivitas, buka akun, manajemen departemen, peran, jabatan, *device*, potongan gaji, dan karyawan.

Penulis menyadari bahwa penyelesaian laporan ini tidak lepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Ir. Bambang Hendrawan, S.T., MSM., CIPMP., CISCIP. selaku Rektor Politeknik Negeri Batam yang telah memberikan dukungan, kebijakan, serta fasilitas sehingga penulis dapat menyelesaikan pendidikan dan penyusunan laporan Tugas Akhir ini dengan baik.
2. Ahmad Hamim Thohari, S.S.T., M.T. selaku Kepala Jurusan yang telah memberikan arahan, dukungan, serta fasilitas akademik sehingga penulis dapat menyelesaikan pendidikan dan penyusunan laporan Tugas Akhir ini dengan baik.
3. Mir'atul Khusna Mufida, Ph.D. selaku Dosen Pembimbing yang telah memberikan bimbingan, arahan, dan masukan yang sangat berharga selama proses penyusunan laporan ini.
4. Yeni Rokhayati, S.Si., M.Sc. selaku Ketua Program Studi Teknik Informatika yang telah memberikan fasilitas dan dukungan akademik.

5. Dosen Penguji yang telah memberikan penilaian, arahan, kritik, serta saran yang konstruktif selama proses pengujian, sehingga laporan Tugas Akhir ini dapat disempurnakan dan memiliki kualitas yang lebih baik.
6. PT. Kreatif System Indonesia yang telah memberikan kesempatan dan dukungan untuk melaksanakan pengembangan sistem.
7. Sorta Marlince Sinaga, Ferdinan Simatupang, Indri Anggi Simatupang, dan Euis Simatupang selaku keluarga penulis yang senantiasa memberikan doa, dukungan moral dan materil, perhatian, serta motivasi yang tiada henti selama proses penyelesaian studi dan penyusunan laporan Tugas Akhir ini.
8. Rekan terdekat penulis, Marsanda Nainggolan NIM 3322301092, yang telah memberikan dukungan, semangat, bantuan, dan motivasi selama proses penyusunan laporan ini.
9. Semua pihak yang tidak dapat penulis sebutkan satu per satu, yang telah membantu dalam penyelesaian laporan ini.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun dari pembaca demi perbaikan dan penyempurnaan laporan ini. Semoga laporan ini dapat memberikan manfaat bagi pembaca dan pihak-pihak yang membutuhkan, khususnya dalam bidang pengembangan sistem informasi.

Batam, 15 Desember 2025

Penulis,

Grey Ari Daniel Simatupang 3312311133

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	vi
DAFTAR TABEL	ix
DAFTAR LAMPIRAN	xi
ABSTRAK	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Batasan Masalah	4
1.5 Manfaat	5
BAB II TINJAUAN PUSTAKA	6
2.1 Penelitian Terkait.....	6
2.2 Landasan Teori	9
2.2.1 <i>Human Resource Management</i> (HRM)	10
2.2.2 Sistem Informasi Manajemen.....	11
2.2.3 <i>Human Resource Information System</i> (HRIS)	11
2.2.4 Agile.....	12
2.2.5 Visual Studio code.....	12
2.2.6 XAMPP	13
2.2.7 Bahasa Pemrograman	13
2.2.8 <i>Database</i>	14
2.2.9 <i>Framework</i>	15
2.2.10 <i>Payroll</i>	16
2.2.11 RBAC (<i>Roll Base Access Controller</i>).....	17
2.2.12 <i>Two Step Verification</i>	17
2.2.13 GPS (<i>Global Positioning System</i>)	18
2.2.14 <i>DEVICE ID</i>	18

2.2.15	BREVO	19
2.2.16	<i>Maintenance</i>	19
2.2.17	Pajak.....	20
2.2.18	<i>Application Programming Interface (API)</i>	20
BAB III ANALISIS DAN PERANCANGAN.....		21
3.1	Analisis Kebutuhan.....	21
3.1.1	<i>Sprint Planning</i>	22
3.1.2	Proses Bisnis Berjalan	28
3.1.3	Gambaran Umum Sistem.....	28
3.2	Perancangan.....	30
3.2.1	Kebutuhan <i>Fungsional</i>	31
3.2.2	Kebutuhan <i>Non-Fungsional</i>	33
3.2.3	Diagram <i>Use Case</i>	33
3.2.4	<i>Scenario Use Case</i>	36
3.2.5	<i>Activity Diagram</i>	69
3.2.6	ER Diagram.....	103
3.2.7	Perancangan Antarmuka (<i>Wireframe</i>)	107
BAB IV IMPLEMENTASI DAN PENGUJIAN		109
4.1	Hasil Implementasi	109
4.1.1	Implementasi Basis data	110
4.1.2	Implementasi API.....	110
4.1.3	Implementasi Pada Sisi UI.....	121
4.2	Pengujian <i>User Acceptance Testing (UAT)</i>	134
4.2.1	Pengujian Perangkat Lunak (<i>Functional Testing</i>).....	134
4.2.2	Pengujian Perangkat Lunak (<i>Non-Functional Testing</i>)	234
BAB V KESIMPULAN		244
5.1	Kesimpulan.....	244
5.2	Saran	244
DAFTAR PUSTAKA		246
DAFTAR LAMPIRAN		250
Lampiran A. Dokumen Proses Pengumpulan <i>Requirement</i>		250

Lampiran B. Link Produk.....	250
Lampiran C. Dokumen Pengujian UAT.....	250

DAFTAR GAMBAR

Gambar 3.1 Gambaran Umum Sistem	30
Gambar 3.2 Metode Pengembangan Sistem Agile	31
Gambar 3.3 Diagram <i>Use Case</i>	34
Gambar 3.4 Activity Diagram <i>Login Normal Scenario</i>	70
Gambar 3.5 Activity Diagram <i>Logout Normal Scenario</i>	70
Gambar 3.6 Activity Diagram Manajemen Jabatan <i>Normal Scenario</i>	71
Gambar 3.7 Activity Diagram Manajemen Jabatan <i>Alternative Scenario 1</i>	71
Gambar 3.8 Activity Diagram Manajemen Jabatan <i>Alternative Scenario 2</i>	72
Gambar 3.9 Activity Diagram Manajemen Jabatan <i>Alternative Scenario 3</i>	72
Gambar 3.10 Activity Diagram Manajemen Departemen <i>Normal Scenario</i>	73
Gambar 3.11 Activity Diagram Manajemen Departemen <i>Alternative Scenario 1</i> . 73	
Gambar 3.12 Activity Diagram Manajemen Departemen <i>Alternative Scenario 2</i> . 74	
Gambar 3.13 Activity Diagram Manajemen Departemen <i>Alternative Scenario 3</i> . 74	
Gambar 3.14 Activity Diagram Manajemen Peran Hak Akses <i>Normal Scenario</i> . 75	
Gambar 3.15 Activity Diagram Manajemen Peran Hak Akses <i>Normal Alternative Scenario 1</i>	75
Gambar 3.16 Activity Diagram Manajemen Peran Hak Akses <i>Normal Alternative Scenario 2</i>	76
Gambar 3.17 Activity Diagram Manajemen Peran Hak Akses <i>Normal Alternative Scenario 3</i>	76
Gambar 3.18 Activity Diagram Manajemen Karyawan <i>Normal Scenario</i>	77
Gambar 3.19 Activity Diagram Manajemen Karyawan <i>Alternative Scenario 1</i>	77
Gambar 3.20 Activity Diagram Manajemen Karyawan <i>Alternative Scenario 2</i>	78
Gambar 3.21 Activity Diagram Manajemen Karyawan <i>Alternative Scenario 3</i>	78
Gambar 3.22 Activity Diagram Manajemen Kantor <i>Normal Scenario</i>	79
Gambar 3.23 Activity Diagram Manajemen Kantor <i>Alternative Scenario 1</i>	79
Gambar 3.24 Activity Diagram Manajemen Kantor <i>Alternative Scenario 2</i>	80
Gambar 3.25 Activity Diagram Manajemen Potongan Gaji <i>Normal Scenario</i>	80
Gambar 3.26 Activity Diagram Manajemen Potongan Gaji <i>Alternative Scenario 1</i>	81
Gambar 3.27 Activity Diagram Manajemen Potongan Gaji <i>Alternative Scenario 2</i>	81
Gambar 3.28 Activity Diagram Manajemen Potongan Gaji <i>Alternative Scenario 3</i>	82
Gambar 3.29 Activity Diagram Log Aktivitas <i>Normal Scenario</i>	82
Gambar 3.30 Activity Diagram Mengelola <i>Reset Normal Scenario</i>	83
Gambar 3.31 Activity Diagram Mengelola <i>Reset Alternative Scenario 1</i>	83

Gambar 3.32 Activity Diagram Manajemen Gaji Normal Scenario	84
Gambar 3.33 Activity Diagram Manajemen Gaji Alternative Scenario 1.....	84
Gambar 3.34 Activity Diagram Export Excel Gaji Normal Scenario	85
Gambar 3.35 Activity Diagram Manajemen Pengingat Normal Scenario	85
Gambar 3.36 Activity Diagram Manajemen Pengingait Alternative Scenario 1...	86
Gambar 3.37 Activity Diagram Manajemen Pengingait Alternative Scenario 1...	86
Gambar 3.38 Activity Diagram Manajemen Pengingait Alternative Scenario 3...	87
Gambar 3.39 Activity Diagram Melihat Daftar Absensi Normal Scenario.....	87
Gambar 3.40 Activity Diagram Melakukan Absensi Normal Scenario	88
Gambar 3.41 Activity Diagram Melakukan Absensi Alternative Scenario 1.....	88
Gambar 3.42 Activity Diagram Manajemen Tugas Normal Scenario	89
Gambar 3.43 Activity Diagram Manajemen Tugas Alternative Scenario 1	89
Gambar 3.44 Activity Diagram Manajemen Tugas Alternative Scenario 2	90
Gambar 3.45 Activity Diagram Manajemen Tugas Alternative Scenario 3.....	90
Gambar 3.46 Activity Diagram Upload Lampiran Tugas Normal Scenario.....	91
Gambar 3.47 Activity Diagram Pengajuan Lembur Normal Scenario.....	91
Gambar 3.48 Activity Diagram Pengajuan Lembur Alternative Scenario 1	92
Gambar 3.49 Activity Diagram Menyetujui Lembur Awal Normal Scenario.....	92
Gambar 3.50 Activity Diagram Menyetujui Lembur Awal Alternative Scenario 1	93
Gambar 3.51 Activity Diagram Menyetujui Lembur Akhir Normal Scenario	93
Gambar 3.52 Activity Diagram Menyetujui Lembur Akhir Alternative Scenario 1	94
Gambar 3.53 Activity Diagram Pengajuan Cuti Normal Scenario	94
Gambar 3.54 Activity Diagram Pengajuan Cuti Alternative Scenario 1.....	95
Gambar 3.55 Activity Diagram Menyetujui Cuti Awal Normal Scenario	95
Gambar 3.56 Activity Diagram Menyetujui Cuti Awal Alternative Scenario 1....	96
Gambar 3.57 Activity Diagram Menyetujui Cuti Akhir Normal Scenario	96
Gambar 3.58 Activity Diagram Menyetujui Cuti Akhir Alternative Scenario 1 ...	97
Gambar 3.59 Activity Diagram Mengelola Profile Normal Scenario.....	97
Gambar 3.60 Activity Diagram Mengelola Profile Alternative Scenario 1	98
Gambar 3.61 Activity Diagram Mengelola Profile Alternative Scenario 2	98
Gambar 3.62 Activity Diagram Mengelola Pengaturan Normal Scenario	99
Gambar 3.63 Activity Diagram Mengelola Pengaturan Alternative Scenario 1....	99
Gambar 3.64 Activity Diagram Mengelola Pengaturan Alternative Scenario 2..	100
Gambar 3.65 Activity Diagram Lupa Kata Sandi Normal Scenario	100
Gambar 3.66 Activity Diagram Membuka Akun Karyawan Normal Scenario...	101
Gambar 3.67 Activity Diagram Membuka Akun Karyawan Alternative Scenario 1	101

Gambar 3.68 <i>Activity Diagram Menghapus Device Karyawan Normal Scenario</i>	102
Gambar 3.69 <i>Activity Diagram Menghapus Device Karyawan Alternative Scenario 1</i>	102
Gambar 3.70 ER Diagram	103
Gambar 3.71 <i>Wireframe Mobile</i>	107
Gambar 3.72 <i>Wireframe Web</i>	108
Gambar 4.1 Implementasi Basis Data	110
Gambar 4.2 Implementasi <i>Lending Page</i>	122
Gambar 4.3 Implementasi <i>Login</i>	122
Gambar 4.4 Implementasi <i>Dashboard</i>	123
Gambar 4.5 Implementasi Kehadiran	124
Gambar 4.6 Implementasi Cuti	125
Gambar 4.7 Implementasi Lembur	126
Gambar 4.8 Implementasi Tugas	126
Gambar 4.9 Implementasi Karyawan	127
Gambar 4.10 Implementasi Karyawan	127
Gambar 4.11 Implementasi Departemen	128
Gambar 4.12 Implementasi Jabatan	128
Gambar 4.13 Implementasi Peran	129
Gambar 4.14 Implementasi Log <i>Aktivitas</i>	129
Gambar 4.15 Implementasi Potongan	130
Gambar 4.16 Implementasi <i>Reminder</i>	131
Gambar 4.17 Implementasi Info Kantor	131
Gambar 4.18 Implementasi <i>Reset Data</i>	132
Gambar 4.19 Implementasi <i>Reset Perangkat</i>	132
Gambar 4.20 Implementasi <i>Reset Unlock</i>	133
Gambar 4.21 Implementasi <i>Reset Profile</i>	133
Gambar 4.22 Implementasi Pengaturan	134

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian Sebelumnya	6
Tabel 3.1 <i>User Story Sprint</i> Persiapan Dan Analisis	22
Tabel 3.2 <i>User Story Sprint</i> Modul Autentikasi Dan Master Data.....	23
Tabel 3.3 <i>User Story Sprint</i> Manajemen Karyawan, Log <i>Aktivitas</i> , Dan Kantor .	23
Tabel 3.4 <i>User Story Sprint</i> Sistem Absensi, Mengelola <i>Profile</i> , Dan Pengaturan	24
Tabel 3.5 <i>User Story Sprint</i> Sistem Penugasan	25
Tabel 3.6 <i>User Story Sprint</i> Sistem Cuti.....	25
Tabel 3.7 <i>User Story Sprint</i> Sistem Lembur	26
Tabel 3.8 <i>User Story Sprint</i> Sistem Penggajian, Potongan Gaji, <i>Reset Massal</i> , Dan Peningat.....	26
Tabel 3.9 Sistem Hapus device, Buka Akun, Dan <i>Deployment</i>	27
Tabel 3.10 Kebutuhan <i>Fungsional</i>	31
Tabel 3.11 Kebutuhan <i>Non-Fungsional</i>	33
Tabel 3.12 <i>Scenario Use Case Login</i>	37
Tabel 3.13 <i>Scenario Use Case Logout</i>	38
Tabel 3.14 <i>Scenario Use Case</i> Manajemen Jabatan	39
Tabel 3.15 <i>Scenario Use Case</i> Manajemen Departemen.....	40
Tabel 3.16 <i>Scenario Use Case</i> Manajemen Peran Hak Akses	42
Tabel 3.17 <i>Scenario Use Case</i> Manajemen Karyawan.....	43
Tabel 3.18 <i>Scenario Use Case</i> Manajemen Kantor.....	45
Tabel 3.19 <i>Scenario Use Case</i> Manajemen Potongan Gaji	46
Tabel 3.20 <i>Scenario Use Case</i> Manajemen Log <i>Aktivitas</i>	47
Tabel 3.21 <i>Scenario Use Case</i> Mengelola <i>Reset</i>	48
Tabel 3.22 <i>Scenario Use Case</i> Manajemen Gaji.....	49
Tabel 3.23 <i>Scenario Use Case Export</i> Excel Gaji.....	50
Tabel 3.24 <i>Scenario Use Case</i> Manajemen Peningat	51
Tabel 3.25 <i>Scenario Use Case</i> Melihat Absensi	52
Tabel 3.26 <i>Scenario Use Case</i> Melakukan Absensi	53
Tabel 3.27 <i>Scenario Use Case</i> Manajemen Tugas	54
Tabel 3.28 <i>Scenario Use Case Upload</i> Lampiran Tugas.....	56
Tabel 3.29 <i>Scenario Use Case</i> Pengajuan Lembur	56
Tabel 3.30 <i>Scenario Use Case</i> Menyetujui Lembur Awal	57
Tabel 3.31 <i>Scenario Use Case</i> Menyetujui Lembur Akhir.....	59
Tabel 3.32 <i>Scenario Use Case</i> Pengajuan Cuti	60
Tabel 3.33 <i>Scenario Use Case</i> Menyetujui Cuti Awal	61
Tabel 3.34 <i>Scenario Use Case</i> Menyetujui Cuti Akhir	62

Tabel 3.35 <i>Scenario Use Case Mengelola Profile</i>	64
Tabel 3.36 <i>Scenario Use Case Mengelola Pengaturan</i>	65
Tabel 3.37 <i>Scenario Use Case Lupa Kata Sandi</i>	66
Tabel 3.38 <i>Scenario Use Case Membuka Akun Karyawan</i>	67
Tabel 3.39 <i>Scenario Use Case Menghapus Device Karyawan</i>	68
Tabel 4.1 Dokumentasi Implementasi API	111
Tabel 4.2 <i>Black Box Testing Login</i>	135
Tabel 4.3 <i>Black Box Testing Manajemen Departemen</i>	138
Tabel 4.4 <i>Black Box Testing Manajemen Jabatan</i>	143
Tabel 4.5 <i>Black Box Testing Peran Hak Akses</i>	147
Tabel 4.6 <i>Black Box Testing Manajemen Karyawan</i>	152
Tabel 4.7 <i>Black Box Testing Log Aktivitas</i>	163
Tabel 4.8 <i>Black Box Testing Manajemen Kantor</i>	165
Tabel 4.9 <i>Black Box Testing Mengelola Reset</i>	169
Tabel 4.10 <i>Black Box Testing Manajemen Potongan Gaji</i>	172
Tabel 4.11 <i>Black Box Testing Manajemen Gaji</i>	181
Tabel 4.12 <i>Black Box Testing Export Excel Gaji</i>	183
Tabel 4.13 <i>Black Box Testing Manajemen Pengingat</i>	184
Tabel 4.14 <i>Black Box Testing Manajemen Tugas</i>	192
Tabel 4.15 <i>Black Box Testing Upload Lampiran Tugas</i>	198
Tabel 4.16 <i>Black Box Testing Melihat Daftar Absensi</i>	201
Tabel 4.17 <i>Black Box Testing Melakukan Absensi</i>	203
Tabel 4.18 <i>Black Box Testing Pengajuan Cuti</i>	206
Tabel 4.19 <i>Black Box Testing Menyetujui Cuti Awal</i>	208
Tabel 4.20 <i>Black Box Testing Menyetujui Cuti Akhir</i>	211
Tabel 4.21 <i>Black Box Testing Pengajuan Lembur</i>	213
Tabel 4.22 <i>Black Box Testing Menyetujui Lembur Awal</i>	215
Tabel 4.23 <i>Black Box Testing Menyetujui Lembur Akhir</i>	218
Tabel 4.24 <i>Black Box Testing Mengelola Profile</i>	220
Tabel 4.25 <i>Black Box Testing Mengelola Pengaturan</i>	224
Tabel 4.26 <i>Black Box Testing Menghapus Device Karyawan</i>	227
Tabel 4.27 <i>Black Box Membuka Akun Karyawan</i>	228
Tabel 4.28 <i>Black Box Testing Logout</i>	230
Tabel 4.29 <i>Black Box Testing Lupa Kata Sandi</i>	231
Tabel 4.30 <i>Performance Testing</i>	235
Tabel 4.31 <i>Security Testing</i>	240

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan <i>Requirement</i>	250
Lampiran B. Link Produk.....	250
Lampiran C. Dokumen Pengujian UAT	250

ABSTRAK

Pengelolaan SDM di Perusahaan Penyedia Jasa Teknologi Informasi (IT) masih manual dan semi-otomatis, menyebabkan kendala dalam monitoring kehadiran, risiko manipulasi data, dan *human error* pada penggajian. Untuk mengatasi permasalahan tersebut, diperlukan sistem *Human Resource Information System* (HRIS) yang dapat mengintegrasikan seluruh proses manajemen karyawan dalam satu *platform* digital yang efisien dan akurat. Tujuan dari proyek akhir ini adalah merancang dan membangun aplikasi HRIS berbasis *web* dan *mobile* yang mampu mengelola data karyawan, absensi dengan verifikasi GPS dan *video*, sistem penugasan, pengajuan cuti dan lembur dengan mekanisme *two-step approval*, serta sistem penggajian otomatis yang terintegrasi dengan perhitungan potongan gaji. Pengembangan sistem menggunakan metodo Agile dengan *framework Scrum*. Teknologi yang digunakan meliputi *framework* Laravel untuk *backend*, Flutter untuk aplikasi *mobile web*, MySQL sebagai *database*, serta integrasi dengan API Brevo untuk pengiriman notifikasi email otomatis. Hasil dari proyek ini adalah aplikasi HRIS yang menyediakan 28 fitur *fungsi*onal yang mencakup autentikasi dengan *single device login*, absensi berbasis GPS dan *video*, sistem penugasan dengan tracking lokasi, mekanisme *approval* bertingkat untuk cuti dan lembur, serta sistem penggajian yang otomatis menghitung gaji berdasarkan kehadiran, lembur, dan potongan gaji. Pengujian *fungsi*onal menunjukkan bahwa seluruh fitur berjalan sesuai dengan kebutuhan yang telah ditetapkan. Dengan adanya sistem HRIS ini, Perusahaan Penyedia Jasa Teknologi Informasi (IT) dapat meningkatkan efektivitas, efisiensi, dan akurasi dalam pengelolaan sumber daya manusia, meminimalkan potensi kesalahan manual, serta menyediakan data yang terstruktur dan dapat dipertanggungjawabkan untuk mendukung pengambilan keputusan manajerial.

Kata Kunci: *Human Resource Information System*, Agile Scrum, Absensi GPS dan Video, Sistem Penggajian Otomatis, Manajemen Karyawan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Transformasi digital telah mengubah cara organisasi dalam mengelola sumber daya manusia, khususnya pada perusahaan teknologi informasi Perusahaan Penyedia Jasa Teknologi Informasi (IT). Manajemen data karyawan yang optimal dan sistematis menjadi faktor fundamental dalam meningkatkan produktivitas serta performa organisasi. Perusahaan Penyedia Jasa Teknologi Informasi (IT) adalah perusahaan yang bergerak di bidang perangkat elektronik khususnya sistem keamanan (CCTV) dan komponen pendukungnya serta jasa instalasi yang berlokasi di Ruko Palm Spring Blok B3 No.15, Batam, Kepulauan Riau, yang menghadapi tantangan kompleks dalam pengelolaan sumber daya manusia seiring dengan perkembangan teknologi dan pertumbuhan bisnis. *Human Resource Management* (HRIS) telah berkembang menjadi kebutuhan strategis bagi korporasi kontemporer untuk mengintegrasikan proses operasional HR dengan teknologi informasi (Mirji & Kapoor, 2022). Sistem HRIS bukan hanya berfungsi sebagai alat bantu dalam Administrasi data karyawan, namun juga sebagai pendukung dalam proses decision making strategis melalui analisis data yang presisi dan berbasis real-time.

Perusahaan Penyedia Jasa Teknologi Informasi (IT) dalam posisinya sebagai entitas bisnis yang fokus pada sektor teknologi di wilayah Batam mengalami hambatan dalam menangani data karyawan yang bertambah rumit seiring dengan ekspansi perusahaan. Perusahaan di bidang teknologi memiliki kecenderungan tingkat turnover yang cukup signifikan dan memerlukan sistem manajemen SDM yang responsive serta adaptif terhadap perusahaan (Nisa & Putra, 2022). Penanganan data karyawan dengan metode konvensional atau melalui sistem yang belum terintegrasi dapat memicu berbagai problematika seperti redundansi data, inkonsistensi informasi, penundaan dalam proses Administrative, dan kesulitan dalam mengakses informasi karyawan secara instan. Kondisi tersebut tentunya akan berimplikasi pada efektivitas operasional perusahaan dan tingkat kepuasan karyawan terhadap pelayanan departemen HR (Human Resource). Saat ini,

penyimpanan data masih menggunakan file Excel dan dokumen pendukung seperti KTP yang disimpan di server internal, sehingga tidak dapat diakses ketika server mengalami gangguan (down). Proses absensi pun belum optimal, siswa PKL yang ditugaskan ke cabang perusahaan harus mengirim foto kehadiran melalui WhatsApp, sedangkan mahasiswa dan siswa magang/PKL (Praktek Kerja Lapangan) masih melakukan absensi ganda menggunakan *fingerprint* dan konfirmasi WhatsApp. Pengajuan cuti juga dilakukan secara manual melalui WhatsApp, sementara penghitungan gaji masih menggunakan Excel dan belum mencakup potongan pajak secara otomatis. Selain itu, perusahaan memerlukan fitur pengingat untuk jadwal servis kendaraan operasional, yang hingga kini masih dilakukan secara manual sehingga rawan terlewat dan berpotensi menghambat kelancaran aktivitas perusahaan.

Untuk mengatasi permasalahan tersebut, implementasi sistem HRIS menjadi solusi yang sangat strategis. Dalam *study “Implementing Human Resource Information System”* (HRIS), menunjukkan bahwa keberadaan HRIS memiliki potensi untuk meningkatkan efektivitas dan efisiensi HRM (Panjaitan, 2023). Sistem HRIS yang terintegrasi memberikan kemampuan otomatisasi untuk proses-proses operasional seperti manajemen kehadiran, kalkulasi payroll, Administrasi cuti, dan Administrasi kinerja, sehingga dapat mengatasi permasalahan duplikasi data dan keterlambatan proses Administratif yang dialami PT. KSI. Di samping itu sistem HRIS juga menyediakan *dashboard* interaktif dan reporting yang memfasilitasi manajemen dalam monitoring serta evaluasi performa SDM secara holistic. Melalui implementasi sistem berbasis komputer, organisasi dapat meminimalkan risiko kesalahan manusia dan meningkatkan tingkat akurasi data karyawan.

Dalam era modernisasi saat ini, setiap perusahaan dituntut untuk mampu mengelola data karyawan dengan cepat, tepat, dan terintegrasi agar dapat bersaing secara efektif. Perusahaan Penyedia Jasa Teknologi Informasi (IT) sebagai perusahaan yang bergerak di bidang teknologi informasi membutuhkan sistem pengelola sumber daya manusia yang mendukung aktivitas operasional sehari-hari. Proses pencatatan manual yang masih digunakan menimbulkan berbagai kendala,

jika kondisi tersebut dibiarkan, maka dapat menghambat produktivitas dan menurunkan kualitas pelayanan internal perusahaan. Oleh karena itu, diperlukan suatu sistem informasi yang mampu membantu pengelolaan data pegawai secara lebih efisien, akurat, dan mudah diakses. *Human Resource Information System* (HRIS) menjadi salah satu solusi yang dapat memberikan manfaat signifikan, tidak hanya dalam mempermudah Administrasi pegawai, tetapi juga dalam menyediakan informasi yang mendukung pengambilan keputusan strategis. Berdasarkan latar belakang tersebut, maka disusunlah Projek akhir dengan judul “Rancang Bangun Aplikasi HRIS Pengelolaan Data Pegawai Perusahaan Penyedia Jasa Teknologi Informasi (IT).

1.2 Rumusan Masalah

Rumusan masalah disusun untuk memperjelas fokus dalam pembuatan aplikasi HRIS yang dapat meningkatkan skala stabilitas pekerjaan di Perusahaan Penyedia Jasa Teknologi Informasi (IT). Permasalahan yang akan dibahas dalam proposal ini antara lain:

1. Bagaimana solusi yang dapat diterapkan terkait belum adanya mekanisme absensi karyawan, mahasiswa, maupun siswa yang bertugas di luar kantor?
2. Bagaimana merancang dan mengembangkan sistem absensi yang terintegrasi sehingga proses pencatatan kehadiran, termasuk perhitungan potongan pajak gaji, dapat direkap secara otomatis serta dijadikan dasar yang valid dalam pembayaran gaji?
3. Bagaimana solusi dalam mengatasi kendala pengingat untuk melakukan servis kendaraan operasional secara berkala?
4. Bagaimana merancang sistem pengajuan serta pencatatan cuti dan lembur agar dapat dikelola secara efektif dan terstruktur?

1.3 Tujuan

Adapun tujuan dari proyek ini, yaitu:

1. Merancang dan membangun sistem tugas dan absensi karyawan yang bertugas di luar kantor maupun di kantor dengan memanfaatkan teknologi GPS dan *video* sebagai bukti kehadiran yang akurat.

2. Merancang dan mengembangkan sistem absensi dan potongan pajak yang terintegrasi sehingga proses pencatatan kehadiran dapat dilakukan secara lebih efektif, mudah rekapitulasi, serta dapat dijadikan dasar yang valid dalam perhitungan pembayaran gaji.
3. Merancang dan membangun sistem pengingat service kendaraan berkala yang dapat mengirim notifikasi *email* secara otomatis.
4. Merancang dan membangun sistem yang dapat mencatat dan melakukan verifikasi cuti/lembur agar dapat diverifikasi oleh manajer.

1.4 Batasan Masalah

Bagian ini memuat ruang lingkup pengembangan sistem agar lebih terarah dan sesuai dengan kondisi aktual. Batasan sistem yang dirancang meliputi:

1. Aplikasi HRIS ini hanya digunakan untuk pengolahan data pegawai, absensi, cuti, lembur, tugas gaji, departemen, peran, jabatan, potongan gaji, kantor, pengingat, dan *log* aktivitas karyawan di Perusahaan Penyedia Jasa Teknologi Informasi (IT).
2. Aplikasi hanya akan dikembangkan untuk akses internal Perusahaan Penyedia Jasa Teknologi Informasi (IT) melalui *mobile* dan web bukan untuk *public*.
3. Proposal ini tidak membahas UI/UX hanya berfokus pada sistem (*Backend*).
4. Studi kasus proposalnya ini berada pada Perusahaan Penyedia Jasa Teknologi Informasi (IT).
5. Pengambilan lokasi disarankan hanya melalui smartphone karena akurasi lokasi pada perangkat *mobile* dengan GPS berkisar 5–20 meter, sedangkan jika dilakukan melalui web (laptop) hanya mengandalkan Wi-Fi positioning (20–200 meter) atau IP address (bisa meleset hingga puluhan kilometer). Hal ini berpotensi menimbulkan kesenjangan akurasi yang signifikan dan dapat mempengaruhi kevalidan data absensi.
6. Aplikasi mobile yang dikembangkan dibatasi hanya untuk platform Android dalam bentuk file APK (*Android Package Kit*).
7. Sistem penggajian hanya terhubung dengan data absensi dan potongan gaji, serta tidak terintegrasi dengan data cuti dan lembur.

1.5 Manfaat

Adapun manfaat dari proyek ini, yaitu:

1. Membantu perusahaan dalam mengelola data karyawan secara lebih terorganisir, terstruktur, dan efisien.
2. Meningkatkan akurasi serta mengurangi risiko *human error* pada proses absensi dan penggajian.
3. Mempermudah monitoring karyawan baik di kantor maupun di lapangan dengan dukungan GPS dan *log* aktivitas.
4. Mengotomatisasi pengingat kewajiban Administratif sehingga mengurangi potensi keterlambatan.
5. Menyediakan *dashboard* interaktif yang mendukung manajemen dalam pengambilan keputusan strategis.
6. Menambah wawasan tentang penerapan *Human Resource Information System* (HRIS) dalam pengelolaan data karyawan di perusahaan.
7. Memberikan pengalaman praktis dalam merancang dan mengembangkan aplikasi HRIS berbasis web dan *mobile*, sehingga memperkaya pengetahuan dan keterampilan di bidang teknologi informasi.
8. Menjadi sarana untuk mengimplementasikan teori yang diperoleh selama perkuliahan ke dalam praktik nyata melalui proyek akhir.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Berikut adalah beberapa kajian atau referensi tinjauan pustaka yang dilihat pada tabel 2.1.

Tabel 2.1 Perbandingan Penelitian Sebelumnya

Penelitian /Tahun	Judul Penelitian	Teknologi	Metode	Kelebihan	Kekurangan
Suwarno & Jesselyn (2024)	Pengembangan Aplikasi HRIS pada PT. XYZ	<i>Firebase/MySQL</i>	<i>Agile Scrum Blackbox Testing</i>	Fleksibilitas tinggi, integrasi <i>real-time</i> dengan <i>database</i> <i>Firebase/MySQL</i> , mendukung <i>agile development</i> dan pengujian otomatis	Dokumentasi fitur HRIS masih terbatas, skala sistem belum diuji untuk jumlah karyawan sangat besar.
Nusantoko & Prapanca (2025)	Pngembangan Aplikasi HRIS <i>Mobile</i> dengan Absensi Wajah menggunakan CNN pada BMT Bahtera Pekalongan	<i>Firebase / SQLite</i>	<i>Waterfall, CNN, Blackbox Testing, SUS</i>	Memanfaatkan CNN untuk validasi data/foto, mudah diakses di Android/IOS, mendukung <i>offline</i> dan sinkronisasi data	Performa CNN cukup berat di <i>cevice</i> low-end, potensi kesalahan akurasi jika data training terbatas, membutuhkan infrastruktur tambahan untuk training model.
Aris Sunandar & Henri Septanto (2023)	Perancangan HRIS Berbasis Web di PT The Master Steel	MySQL	SDLC <i>Waterfall, Blackbox Testing</i>	Teknologi familiar dan mudah dipelihara, jelas tahapan pengembangan,	Tidak adaptif terhadap perubahan, integrasi <i>mobile</i>

				sistem absensi dan cuti terstruktur dibanding manual.	belum tersedia, keterbatasan pada fitur <i>real-time</i> monitoring.
Riko Jovanka & Fredian Simanjuntak (2025)	Perancangan HRIS Berbasis Website untuk Perusahaan Swasta di Batam	MySQL	<i>Prototyping, Blackbox Testing</i>	Memudahkan komunikasi dengan <i>user</i> , implementasi cepat dan biaya mudah, sistem mampu mengelola data karyawan dasar	Kurang <i>scalable</i> untuk jumlah <i>user</i> besar, keamanan data masih terbatas, tidak mendukung integrasi lintas <i>platform</i>
Raudha Allisya & Sari (2024)	Rancang Bangun Aplikasi HRIS pada PT Stream	MySQL	<i>Waterfall, UML Blackbox Testing</i>	Mudah diterapkan di UKM, alur pengembangan jelas, mendukung pengelolaan data kepegawaian standar.	Kurang fleksibel jika ada perubahan kebutuhan <i>user</i> , tidak mendukung aplikasi <i>mobile</i> , skalabilitas terbatas untuk enterprise level.
Grey Ari Daniel Simatupang (2025)	RANCANG BANGUN SISTEM INFORMASI KEPEGAWAIAN (PEREKRUTAN DAN	MySQL, Laravel, Flutter	<i>Agile</i>	Menggunakan <i>Role-Based Access Control</i> (RBAC), sehingga sistem lebih fleksibel dalam	Dikarenakan keterbatasan perangkat dan kebutuhan, aplikasi belum

	MUTASI) PERUSAHAAN PENYEDIA JASA TEKNOLOGI INFORMASI (IT)			pengelolaan hak akses dan memungkinkan penambahan peran baru di masa mendatang. Mendukung absensi berbasis GPS dan <i>video capture</i>, sehingga kehadiran karyawan dapat divalidasi dengan lebih akurat dan minim kecurangan. Fleksibilitas kontrol perangkat pengguna, di mana secara <i>default</i> satu akun hanya dapat digunakan pada satu perangkat, namun tetap memungkinkan pengaturan sebaliknya sesuai kebutuhan perusahaan. Menyediakan fitur reset massal, yang memungkinkan <i>Admin</i> melakukan penghapusan data secara menyeluruh untuk	mendukung <i>platform</i> iOS.
--	---	--	--	---	---------------------------------------

				mengoptimalkan penyimpanan dan mencegah beban berlebih pada server.	
--	--	--	--	---	--

Penelitian dengan judul “RANCANG BANGUN SISTEM INFORMASI KEPEGAWAIAN (PEREKRUTAN DAN MUTASI) PERUSAHAAN PENYEDIA JASA TEKNOLOGI INFORMASI (IT)” memiliki perbedaan dibandingkan penelitian-penelitian sebelumnya, terutama pada objek penelitian, ruang lingkup sistem yang dikembangkan, serta kebutuhan yang dianalisis. Fokus penelitian ini adalah pada pengelolaan data absensi berbasis GPS dan *video* sebagai bukti kehadiran yang akurat, integrasi absensi dengan potongan pajak dan perhitungan gaji, pencatatan serta verifikasi cuti dan lembur oleh manajer, serta sistem pengingat service kendaraan yang dapat mengirimkan notifikasi/*email* secara otomatis. Data yang digunakan mencakup informasi pegawai, absensi, cuti, lembur, gaji, serta jadwal pengingat yang diperoleh dari perusahaan terkait. Sistem dirancang untuk mendukung proses operasional internal agar lebih efektif, akurat, dan efisien. Hasil penelitian ini berupa perancangan dan pembangunan aplikasi HRIS berbasis *web* dan *mobile* dengan penekanan pada pengembangan *backend system*. Aplikasi yang dihasilkan diharapkan dapat meningkatkan efektivitas pencatatan kehadiran, mempercepat proses rekapitulasi, mempermudah pengelolaan cuti/lembur, serta mendukung digitalisasi manajemen kepegawaian di Perusahaan Penyedia Jasa Teknologi Informasi (IT).

2.2 Landasan Teori

1. Konsep Manajemen Karyawan

Manajemen karyawan adalah proses mengelola sumber daya manusia dalam suatu organisasi agar dapat bekerja secara efektif dan efisien untuk mencapai tujuan perusahaan. Menurut (Yunianti & SE.MM, 2025), manajemen karyawan mencakup kegiatan perencanaan, pengorganisasian, pengarahan, dan pengendalian terhadap tenaga kerja agar memberikan kontribusi optimal.

Sementara (Mulang, 2023) menjelaskan bahwa manajemen karyawan tidak hanya berfokus pada Administrasi, tetapi juga pada pengembangan kompetensi, motivasi, serta penciptaan sistem kerja yang adil dan produktif. Manajemen karyawan bertujuan memastikan seluruh proses yang berkaitan dengan karyawan, mulai dari rekrutmen, penempatan, pengembangan, hingga kompensasi, dapat berjalan sistematis dan terintegrasi, sehingga meningkatkan kinerja individu maupun organisasi.

2. Studi Kasus Perusahaan Penyedia Jasa Teknologi Informasi (IT)

Perusahaan Penyedia Jasa Teknologi Informasi (IT) adalah perusahaan yang bergerak di bidang Teknologi Informasi (IT), melayani perusahaan kecil, menengah, besar, baik swasta maupun pemerintahan, serta berbagai industri lainnya. Perusahaan ini juga menyediakan jasa konsultasi dan implementasi berbagai solusi teknologi. Beberapa produk andalan Perusahaan Penyedia Jasa Teknologi Informasi (IT) meliputi *Security System* (CCTV pengawasan terintegrasi, pengontrol akses, *biometric*, *Auto Gate*), *Audio Paging System*, *Telephone PBX System*, *Computer Network Infrastructure*, dan *Thermal Camera Screening System*.

3. Kondisi Manajemen Karyawan pada Perusahaan Penyedia Jasa Teknologi Informasi (IT)

Berdasarkan hasil observasi, penerapan sistem manajemen karyawan di Perusahaan Penyedia Jasa Teknologi Informasi (IT) masih dilakukan secara manual. Hal ini terlihat dari beberapa aspek berikut:

- Pengajuan cuti masih dilakukan melalui aplikasi pesan instan (WhatsApp), sehingga tidak terdokumentasi secara formal dan menyulitkan proses monitoring.
- Informasi lembur juga disampaikan melalui WhatsApp, yang berpotensi menimbulkan ketidakteraturan dalam pencatatan jam kerja lembur.

2.2.1 *Human Resource Management* (HRM)

Human Resource Management (HRM) ujung tombak dalam melakukan kegiatan untuk mencapai visi sebuah organisasi (Syakoer, 2023). *Human Resource Management* (HRM) memiliki peran yang sangat penting dalam suatu organisasi

terutama pada saat banyaknya persaingan. Perkembangan teknologi mengharuskan HRM mampu melakukan berbagai macam strategi agar dapat bertahan dan berkembang dalam meningkatkan bisnis. Sebuah organisasi dengan tata kelola yang baik tentu memiliki HRM yang bertanggung jawab serta mampu mengarahkan tim dalam perusahaan. HRM berperan secara strategis dalam menghadapi dinamika perubahan organisasi, yakni dengan mempersiapkan sumber daya manusia agar siap beradaptasi melalui pemberian pelatihan, pengemangan, maupun training.

2.2.2 Sistem Informasi Manajemen

Sistem Informasi Manajemen (SIM) mencakup tiga aspek utama: sistem, informasi, dan manajemen. Sistem Informasi Manajemen (SIM) merupakan seperangkat prosedur gabungan untuk mengumpulkan dan menghasilkan data yang andal, relevan, dan terorganisir dengan baik sehingga dapat mendukung proses pengambilan keputusan suatu organisasi. Dalam penelitian (Wijoyo, 2022), Sistem Informasi Manajemen (SIM) adalah suatu sistem yang terdiri dari perangkat lunak, hardware, data, prosedur, dan personil yang terintegrasi dengan tujuan menghasilkan informasi yang berguna dan relevan bagi organisasi dalam mengambil keputusan yang efektif dan efisien. SIM dapat membantu organisasi untuk mengurangi biaya, meningkatkan produktivitas, dan mengambil keputusan yang lebih akurat dan cepat.

2.2.3 Human Resource Information System (HRIS)

HRIS atau *Human Resource Information System* adalah sebuah sistem yang digunakan oleh tim HR dalam mengelola data karyawan, menyederhanakan proses Administratif, dan mendukung pengambilan keputusan strategis di bidang sumber daya manusia. Menurut (Edhy Permata, 2019) HRIS merupakan sebuah bentuk interseksi/pertemuan antara bidang ilmu manajemen sumber daya manusia dan teknologi informasi, sistem ini menggabungkan SDM sebagai suatu disiplin yang utamanya mengaplikasikan teknologi informasi ke dalam aktifitas-aktifitas SDM seperti dalam hal perencanaan, menyusun sistem pemrosesan data dan hasil yang didapat dari sistem dalam serangkaian langkah-langkah yang terstandarisasi dan terangkum dalam aplikasi perencanaan sumber daya perusahaan.

Human Resource Information System (HRIS) adalah sistem informasi yang membantu mengelola data karyawan, proses Administrasi, pengembangan karyawan, proses Administrasi, pengembangan karyawan, dan komunikasi dalam organisasi. Manajemen SDM yang baik bertujuan menciptakan lingkungan kerja yang kondusif, meningkatkan kepuasan karyawan, dan mendorong keterlibatan mereka karena karyawan yang puas dan terlibat cenderung memiliki kinerja lebih baik serta loyal terhadap perusahaan (Abdullah et al., 2024).

HRIS tidak hanya berperan dalam mencatat data karyawan, tetapi juga mencakup berbagai fungsi penting seperti manajemen perekrutan, pelatihan, penggajian, evaluasi kinerja, serta Administrasi cuti dan absensi. Dengan adanya HRIS, perusahaan dapat meningkatkan produktivitas dan mengurangi beban Administrasi terkait pengelolaan SDM.

2.2.4 Agile

Metode Agile adalah pendekatan pengembangan perangkat lunak yang menekankan fleksibilitas, kolaborasi tim, dan pengiriman produk yang cepat dan iteratif. Pendekatan ini berbeda dari model pengembangan tradisional yang lebih linear dan berbasis tahapan, seperti *Waterfall* (Abdul Rahman Ismail, 2025). *Agile development* adalah model pengembangan perangkat lunak dalam jangka pendek, untuk kemudian diadaptasi secara cepat dalam mengatasi setiap perubahan (Hikmah et al., 2021). Nilai terpenting dari *Agile development* ini adalah memungkinkan sebuah tim dalam mengambil keputusan dengan cepat, kualitas dan prediksi yang baik, serta memiliki potensi yang baik dalam menangani setiap perubahan.

2.2.5 Visual Studio code

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 (Kahlert & Giza, 2016). Editor ini secara bawaan sudah mendukung bahasa pemrograman JavaScript, *TypeScript*, dan Node.js, serta dapat diperluas untuk bahasa lain seperti C++, C#, Python, Go, dan Java melalui plugin yang tersedia di marketplace Visual Studio code. Aplikasi ini dilengkapi berbagai fitur, antara lain Intellisense, integrasi Git, debugging, dan dukungan ekstensi yang

semakin memperkaya fungsionalitasnya. Seiring dengan peningkatan versinya, fitur-fitur tersebut juga terus berkembang. Visual Studio Code melakukan pembaruan versi secara rutin setiap bulan, yang menjadi salah satu keunggulannya dibandingkan editor teks lain.

2.2.6 XAMPP

XAMPP merupakan tool yang menyediakan paket perangkat lunak dalam satu buah paket. XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan komplikasi dari beberapa program (Palit et al., 2015). Dengan menggunakan XAMPP, pengguna tidak perlu lagi melakukan instalasi maupun konfigurasi manual untuk web server *Apache*, PHP, dan MySQL, karena semuanya akan diatur secara otomatis melalui fitur auto konfigurasi.

Sebagai perangkat lunak bebas yang mendukung berbagai sistem operasi, XAMPP merupakan kumpulan program yang berfungsi sebagai server lokal (*localhost*). Paket ini terdiri dari Apache HTTP Server, MySQL Database, serta interpreter untuk bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (artinya dapat dijalankan di berbagai sistem operasi), *Apache*, MySQL, PHP, dan Perl. Dirilis di bawah lisensi GNU *General Public License*, XAMPP dikenal sebagai web server yang praktis digunakan untuk menampilkan halaman web dinamis, dan dapat diperoleh secara gratis melalui situs resminya.

2.2.7 Bahasa Pemrograman

2.2.7.1 PHP

PHP adalah singkatan dari “PHP: *Hypertext Preprocessor*”, yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. PHP merupakan bahasa server side scripting dimana dapat berguna untuk membuat suatu desain agar dapat digunakan untuk membuat web (Lincopinis et al., 2023). Yang dimana *server-side scripting* adalah sebuah *script function* yang mana didefinisikan sebagai serangkaian suatu proses perintah yang diberikan akan sepenuhnya dijalankan di server tetapi disertakan pada dokumen HTML biasa. PHP

disebut sebagai bahasa pemrograman *server-side* karena dieksekusi atau diproses di server komputer. Kontras dengan bahasa pemrograman *client-side* seperti JavaScript yang dieksekusi di dalam web browser. PHP sebuah bahasa pemrograman yang bersifat *open source*, artinya dapat dikembangkan oleh siapa saja secara bebas dan gratis untuk tujuan umum.

2.2.7.2 Dart

Dart merupakan bahasa pemrograman yang dibuat oleh Google dan didesain oleh *Lars Bak* dan *Kasper Lund* (Haniefam et al., 2022). Bahasa pemrograman *Dart* dapat digunakan untuk membangun aplikasi server atau dalam bentuk *command line interface*, web, ataupun *mobile* (Android dan iOS). *Dart* merupakan Bahasa pemrograman yang mendukung adanya pendefinisian fungsi di luar kelas atau sering disebut dengan *top-level function*. pada web *Dart* memakai *dart2js* yang artinya *Dart* diubah ke *Javascript* agar bisa dimengerti browser. Pada perangkat desktop *Dart* memakai *dart2aot* yang mengubah *Dart* menjadi bahasa mesin. Pada perangkat *mobile* *Dart* memakai *Flutter*. *Syntax* pada bahasa *Dart* sangat mudah dipelajari, karena *Dart* memiliki kemiripan dengan *syntax-syntax* pemrograman lain, seperti *Javascript* dan *Java*. *Dart* bisa melakukan pemrograman *Asynchronous*, yaitu fungsi yang *non-blocking* yaitu memungkinkan program untuk menjalankan kode lain sambil menunggu fungsi *asynchronous* selesai. Contoh penggunaan sehari-hari adalah pada saat aplikasi sedang mengambil data dari web, aplikasi juga menampilkan halaman loading. *Dart* juga bisa melakukan pemrograman *Reactive*, yaitu pemrograman *asynchronous* dengan stream data. *Stream* data artinya datanya bisa diterima lebih dari sekali, kebalikan dengan *asynchronous* yang datanya hanya sekali terima saja. Pada *Dart asynchronous* menggunakan kata *Future*, *reactive* menggunakan *Stream*.

2.2.8 Database

Database adalah sekumpulan data atau informasi terstruktur yang saling berhubungan dan disimpan secara elektronik di dalam sistem komputer, dimana data tersebut dapat diakses, dikelola, dan diperbarui dengan mudah menggunakan perangkat lunak khusus yang disebut Sistem Manajemen *Database* (DBMS). *Database* atau basis data adalah kumpulan informasi yang disimpan di dalam

komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari data tersebut (Andy Andaru, 2020).

2.2.8.1 MySQL

MySQL merupakan sistem manajemen *database management system* (RDBMS) seperti SQL Server. MySQL merupakan bahasa komputer ataupun bahasa pemrograman yang difokuskan untuk *database* atau penyimpanan data (Putra & Nita, 2019). MySQL memiliki dua bentuk lisensi, yaitu *Free Software* dan *Freeware*. MySQL *Freeware* dibawah lisensi GNY/GPL (*General Public License*). Selain sebagai program *database* gratis dan *open source*, ada juga MySQL yang bersifat komersial, yaitu MySQL AB. Dalam MySQL, terdapat sun bahasa, yaitu *Data Definition Language* (DDL), *Data Manipulation language* (DML). DDL digunakan untuk membangun objek-objek dalam *database*, seperti tabel dan index. DML digunakan untuk menambah, mencari, mengubah, dan menghapus baris dalam tabel. Sedangkan DCL digunakan untuk menangani masalah security/keamanan dalam *database* (Ery Hartati, 2022).

2.2.9 Framework

Framework adalah kerangka kerja yang berisi kumpulan fungsi untuk mengembangkan aplikasi berbasis *website* maupun desktop. *Framework* adalah kumpulan-kumpulan program-program yang dipadukan menjadi satu kerja yang digunakan untuk membantu dalam pembuatan sebuah aplikasi (Ii & Pustaka, 2015). *Framework* beroperasi menyediakan struktur atau kerangka kerja yang telah dirancang sebelumnya untuk konstruksi aplikasi atau situs web. Dalam pengembangan perangkat lunak dan aplikasi, *framework* membantu pengembangan menghemat waktu dan upaya yang biasanya diperlukan untuk membangun fitur dan fungsi dasar.

2.2.9.1 Framework Laravel

Salah satu *Framework* yang membantu dalam pembangunan aplikasi penjualan adalah *Laravel*. *Laravel* merupakan sebuah kerangka kerja pemrograman yang berbasis *opensource* yang dipakai oleh banyak *developer* dari seluruh dunia (Desma Aipina & Harry Witriyono, 2022). Kecepatan pengembangan adalah faktor krusial dalam dunia teknologi yang bergerak cepat, dan *Laravel* yang diklaim dapat

mengurangi waktu yang dibutuhkan untuk menulis dan mengelola kode melalui berbagai fitur otomatis dan *scaffolding*. Laravel juga menjadi salah satu *framework* yang dapat membantu developer untuk memaksimalkan penggunaan PHP di dalam proses pengembangan website. Selain itu, Laravel juga memiliki beberapa fitur unggulan, seperti *template engine*, *routing*, dan *modularity*.

2.2.9.2 Framework Flutter

Flutter merupakan sebuah *framework opensource* yang dibuat oleh Google dengan tujuan untuk membangun antarmuka pengguna (UI) yang konsisten dan menarik di berbagai *platform*, termasuk iOS, Android, dan web (Ainah et al., 2024). Dengan *Flutter* pengembang dapat membuat aplikasi untuk *platform-platform* tersebut menggunakan bahasa pemrograman *Dart*, *Flutter* memungkinkan pengembang untuk membuat aplikasi yang responsive dan cepat dengan menggunakan *widget* yang dapat disesuaikan, serta menyediakan berbagai alat pengembangan yang kuat seperti *hot reload* untuk literasi cepat dalam pengembangan aplikasi. Selain itu *Flutter* menyediakan kerangka kerja *reactive functional*, mesin render 2D, *widget* siap dipakai, dan berbagai alat pengembangan yang membantu dalam membuat aplikasi.

2.2.10 Payroll

Sistem *Payroll* adalah serangkaian langkah dan prosedur untuk menghitung, memproses, dan membayar gaji karyawan di perusahaan (Teja & Patra, 2023). Sistem ini meliputi perhitungan gaji pokok, tunjangan, potongan, pajak, serta pencatatan dan pelaporan terkait penggajian. *Aktivitas* pemberian gaji diharapkan dapat dijalankan dengan metode yang cepat supaya karyawan dapat menerima gajinya tepat waktu. Pengelolaan *payroll* secara otomatis (*automated payroll*) semakin banyak dijadikan fokus penelitian dan aplikasi praktis karena mampu mengurangi kesalahan manusia dan mempercepat proses penggajian. Penerapan *payroll* otomatis juga berdampak pada transparansi dan kontrol internal. Dengan *payroll* digital, data absensi, lembur, potongan gaji, dan tunjangan bisa langsung diambil/dihitung sistem mengurangi inkonsistensi yang sering terjadi dalam sistem manual. Keuntungan bahwa audit internal menjadi lebih mudah dilaksanakan karena jejak transaksi dan perubahan terekam dengan baik dalam

sistem, sehingga bila ada sengketa atau kebutuhan verifikasi, data tersedia (Odun-Ayo & Aina, 2020).

2.2.11 RBAC (*Roll Base Access Controller*)

RBAC adalah model kontrol akses yang membatasi akses pengguna ke sumber daya berdasarkan peran (*role*) mereka dalam organisasi, bukan berdasarkan individu secara langsung. Dalam penelitian (M Sahyudi & Erliyan Redy Susanto, 2025) menemukan bahwa penerapan RBAC dalam sistem ERP mampu meningkatkan keamanan data, mengurangi waktu pengelolaan hak akses, dan memperkecil potensi pelanggaran akses karena pengguna yang tidak berwenang. Implementasi RBAC juga telah diuji di berbagai jenis sistem dan domain dari manajemen pelanggan hingga *supply chain management*. Pada penelitian (Nasich et al., 2025), menunjukkan bahwa sistem berbasis web dengan RBAC menggantikan sistem *spreadsheet* lama, menghasilkan penurunan pelanggaran akses sebesar signifikan (dari lebih dari 70% kesalahan akses menjadi hampir 0) dan pengurangan hak akses yang tidak perlu. Domain *supply chain*, menunjukkan bahwa RBAC yang diimplementasikan dengan *framework laravel + Spatie library* mampu menjaga integritas otoritas dalam sistem dengan banyak modul/pengguna berbeda, yang penting untuk menjaga keamanan sistem yang kompleks (Yuricha, 2023).

2.2.12 Two Step Verification

Two step verification atau verifikasi dua langkah pada WhatsApp merupakan fitur keamanan tambahan yang dirancang untuk melindungi data pribadi pengguna, melengkapi sistem keamanan sebelumnya seperti verifikasi nomor telepon, kunci aplikasi, dan enkripsi *end-to-end* (Rajput et al., 2024). Tujuan utama dari metode ini adalah mencegah penyalahgunaan nomor telepon yang sudah kedaluwarsa atau didaur ulang oleh *provider*, yang bisa berpotensi digunakan untuk mengakses akun pada perangkat lain. Selain itu, penerapan verifikasi dua langkah juga meningkatkan perlindungan terhadap upaya peretasan akun oleh pihak tidak bertanggung jawab. Fitur ini memastikan bahwa hanya pemilik asli yang tetap memiliki kendali penuh atas akunnya. Dengan adanya mekanisme ini, pengguna

juga lebih mudah dan aman saat berpindah ke perangkat baru karena akses hanya dapat dilakukan oleh pemilik sah.

2.2.13 GPS (*Global Positioning System*)

GPS (*Global Positioning System*) merupakan sistem navigasi berbasis satelit yang banyak dimanfaatkan dalam berbagai bidang, termasuk absensi karyawan. GPS bekerja dengan prinsip triangulasi, yaitu perangkat penerima (*receiver*) menerima sinyal dari minimal tiga satelit untuk menghitung koordinat lokasi berupa lintang (*latitude*) dan bujur (*longitude*). Saat ini, GPS sudah terintegrasi dalam hampir semua perangkat *smartphone* sehingga dapat dimanfaatkan untuk berbagai kebutuhan, mulai dari navigasi transportasi, pelacakan kendaraan, hingga sistem absensi berbasis lokasi. Dengan adanya GPS, organisasi dapat memastikan kehadiran karyawan atau aktivitas tertentu benar-benar dilakukan di lokasi yang telah ditentukan, sehingga meningkatkan transparansi dan akuntabilitas. Selain efisiensi, penggunaan GPS dalam absensi online juga meningkatkan validitas data kehadiran karena aplikasi tidak hanya mencatat koordinat lokasi (*latitude* dan *longitude*), tetapi juga mengintegrasikan verifikasi dengan kamera *selfie* untuk mencegah kecurangan.

Dalam (Seran et al., 2023), menemukan bahwa absensi manual dengan tanda tangan rawan hilang, tidak efisien, serta berpotensi dimanipulasi. Dengan absensi berbasis GPS yang terintegrasi di *smartphone*, pegawai dapat langsung mencatat kehadiran dari lokasi kerja masing-masing. Sistem ini juga dilengkapi verifikasi tambahan berupa foto *selfie* untuk memastikan fasilitas absensi, sehingga menghasilkan data yang lebih akurat dan dapat diandalkan untuk Administrasi pemerintahan.

2.2.14 DEVICE ID

Device ID merupakan identitas unik yang terdapat pada setiap perangkat, baik *smartphone*, komputer, maupun tablet, yang digunakan untuk membedakan satu perangkat lainnya. Identitas ini bisa berupa IMEI (*International Mobile Equipment Identity*), *mac Address*, Android ID, maupun kode unik lain yang melekat pada sistem perangkat. Dalam konteks keamanan sistem informasi, *device ID* berfungsi sebagai pengenalan tambahan yang dapat digunakan untuk mengikat

akun pengguna hanya pada perangkat tertentu. Dengan cara lain, organisasi dapat membatasi potensi penyalahgunaan akun seperti *login* ganda dari perangkat yang berbeda, manipulasi identitas perangkat, maupun aktivitas ilegal yang tidak diotorisasi.

Dalam penelitian (Djotchuang et al., 2025), *device* ID didefinisikan sebagai identitas tunggal milik perangkat yang dapat dijadikan lapisan autentikasi tambahan pada sebuah aplikasi. Pada penelitian tersebut, *device* ID dipadukan dengan faktor keamanan lain seperti kata sandi, sidik jari, dan OTP untuk memastikan bahwa akun hanya bisa diakses melalui perangkat yang terdaftar, hasil penelitian tersebut menunjukkan bahwa penerapan metode ini mampu meningkatkan keamanan sistem, mencegah peretasan melalui perangkat yang tidak sah, serta mengurangi risiko duplikasi akun dalam layanan berbasis *mobile*.

2.2.15 BREVO

Bravo adalah *platform* komunikasi dan pemasaran terpadu (*all-in-one*) yang menggabungkan kemampuan *email* marketing, SMS, chat, automasi pemasaran, dan fungsi CRM untuk mengelola hubungan pelanggan. *Platform* ini memudahkan pembuatan kampanye *email* dengan template siap pakai, automasi berbasis event (*drip/workflow*), serta integrasi kanal lain seperti WhatsApp dan SMS, sehingga tim pemasaran atau HR dapat mengirim pemberitahuan transaksional/otomatis (misalnya. pengingat, notifikasi penggajian) dari satu sistem terpusat (Sari et al., 2023). Bravo menawarkan paket gratis dengan batas pengiriman tertentu (contoh: 300 *email*/hari pada paket gratis) dan dirancang agar mudah dipakai oleh usaha kecil-menengah.

2.2.16 Maintenance

Maintenance didefinisikan sebagai aktivitas pemeliharaan sistem yang bertujuan untuk meningkatkan reliabilitas, memperbaiki kesalahan, serta menyesuaikan sistem terhadap kebutuhan baru yang muncul. *Maintenance* atau pemeliharaan adalah serangkaian aktivitas yang dilakukan untuk menjaga agar suatu sistem, perangkat, atau infrastruktur tetap berfungsi dengan baik sesuai tujuan awalnya (Troisi et al., 2021). Dalam konteks teknologi informasi, *maintenance* mencakup perawatan perangkat keras (*hardware*), perangkat lunak (*software*), serta

basis data agar sistem tetap berjalan optimal, aman, dan minim gangguan (Junaidi et al., 2018). Kegiatan *maintenance* meliputi pembaruan sistem, pengecekan bug, pencadangan data, serta pencegahan kerusakan. Dengan adanya *maintenance* yang terjadwal dan berkelanjutan, sistem dapat bekerja lebih efisien, umur pakai perangkat meningkat, dan risiko kerusakan mendadak bisa diminimalisir.

2.2.17 Pajak

Pajak adalah iuran wajib yang ditetapkan oleh peraturan perundang-undangan yang harus dibayar oleh orang pribadi atau badan kepada negara. pajak digambarkan sebagai kontribusi wajib yang harus dibayar oleh warga negara atau badan hukum berdasarkan undang-undang, yang bersifat memaksa, dan tanpa imbalan langsung (Ariffin & Sitabuana, 2022). Pajak tersebut digunakan oleh pemerintah untuk membiayai pengeluaran publik, seperti pembangunan infrastruktur, layanan kesehatan, pendidikan, dan fungsi pemerintahan lainnya. Dalam penelitian (Rioni & Syauqi, 2020), menjelaskan bahwa pemotongan PPh 21 menyebabkan gaji yang diterima karyawan berkurang dari jumlah kotor (*gross salary*) menjadi jumlah bersih (*take home pay*). Hal ini berdampak pada daya beli dan motivasi karyawan, sehingga perusahaan perlu memastikan perhitungan pajak dilakukan dengan benar agar tidak menimbulkan masalah kepuasan kerja. Selain itu, kepatuhan pajak perusahaan dalam memotong dan menyetorkan PPh 21 juga berimplikasi pada citra dan legalitas perusahaan di mata pemerintah.

2.2.18 Application Programming Interface (API)

Application Programming Interface (API) merupakan suatu dokumentasi yang terdiri dari *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak (Meng et al., 2019). *Application Programming Interface (API)* adalah antarmuka yang dibangun oleh pengembang sistem sehingga semua fungsi sistem dapat diakses secara terprogram. Adanya API memberikan kemudahan bagi *programmer* untuk menganalisis suatu perangkat lunak. API berfungsi sebagai penghubung antar aplikasi sehingga programmer dapat memanfaatkan fungsi-fungsi sistem yang tersedia, dengan pengelolaan melalui sistem operasi.

BAB III

ANALISIS DAN PERANCANGAN

Analisis dan perancangan pada bagian ini difokuskan untuk menghasilkan rancangan aplikasi HRIS yang sesuai dengan ruang lingkup penelitian. Perancangan dilakukan agar sistem yang dibangun benar-benar dapat digunakan sebagai solusi dalam pengelolaan data pegawai, absensi, cuti, lembur, tugas, dan penggajian di Perusahaan Penyedia Jasa Teknologi Informasi (IT). Setiap tahapan analisis diarahkan untuk mengidentifikasi kebutuhan fungsional maupun non-fungsional, sehingga sistem yang dihasilkan tidak hanya memenuhi aspek teknis, tetapi juga mampu meningkatkan efektivitas, efisiensi, dan akurasi dalam mendukung kegiatan operasional perusahaan. Selain itu, perancangan ini juga mempertimbangkan aspek keamanan, kemudahan penggunaan, serta integrasi dengan teknologi pendukung, sehingga aplikasi HRIS dapat berfungsi optimal dan berkelanjutan.

3.1 Analisis Kebutuhan

Proses manajemen karyawan di Perusahaan Penyedia Jasa Teknologi Informasi (IT) masih berjalan secara semi-otomatis dan menghadapi sejumlah kendala. Sistem absensi, misalnya, masih dilakukan secara manual dengan mengirimkan foto di lokasi tugas bagi teknisi yang ditugaskan di lapangan, sehingga rawan manipulasi. Sementara itu, untuk karyawan yang bekerja di kantor sudah tersedia absensi *fingerprint*, namun belum terintegrasi dengan sistem absensi lapangan. Monitoring tugas juga menjadi tantangan karena *Admin* harus memantau karyawan di lokasi yang berbeda-beda tanpa dukungan sistem yang memadai. Selain itu, pengingat pembayaran kewajiban seperti pajak kendaraan atau jadwal servis kendaraan masih dilakukan secara manual oleh *Admin*, sehingga rawan terlewat.

Kendala lain muncul pada aspek keamanan, penggajian, dan Administrasi karyawan. Untuk membatasi potensi kecurangan, dibutuhkan sistem dengan aturan satu akun hanya untuk satu perangkat (kecuali *Admin Super*) serta *log* aktivitas guna memantau setiap tindakan yang dilakukan pengguna. Proses penggajian masih

dilakukan secara manual, sehingga berisiko menimbulkan *human error*; karena itu diperlukan sistem *payroll* otomatis yang menghitung gaji dan potongan gaji berdasarkan kinerja karyawan. Dari sisi pengelolaan hak akses, perusahaan membutuhkan penerapan *Role-Based Access Control (RBAC)* agar lebih fleksibel ketika harus menambahkan peran dengan akses fitur berbeda di masa depan. Selain itu, sistem cuti dan lembur juga perlu diintegrasikan, dengan mekanisme *two-step approval* untuk memastikan validitas pengajuan dan transparansi data, sehingga manajemen karyawan menjadi lebih efisien, akurat, dan dapat dipertanggungjawabkan.

3.1.1 *Sprint Planning*

1. Persiapan Dan Analisis (Minggu 1-2)

Tabel 3.1 *User Story Sprint* Persiapan Dan Analisis

ID	User Story	Task	Estimasi	Status
US-1	Sebagai <i>developer</i> , saya perlu menganalisis kebutuhan sistem	- Wawancara dengan <i>stakeholder</i> - Dokumentasi kebutuhan fungsional - Dokumentasi kebutuhan non-fungsional	3 hari	Done
US-2	Sebagai <i>developer</i> , saya perlu menyiapkan environment	- Setup database - Setup Laravel backend - Setup Flutter project - Setup repository	1 hari	Done
US-3	Sebagai <i>developer</i> , saya perlu merancang arsitektur sistem	- Diagram ERD - Diagram use case - Diagram Activity - Desain UI/UX mockup	6 hari	Done
US-4	Sebagai <i>developer</i> , saya perlu merancang database	- Buat struktur tabel - Definisi relasi antar tabel	2 hari	Done

2. Modul Autentikasi Dan Master Data (Minggu 3-4)

Tabel 3.2 *User Story Sprint* Modul Autentikasi Dan Master Data

ID	User Story	Task	Estimasi	Status
US-1	Sebagai Admin Super/Admin Office/Teknisi, saya ingin login ke sistem	- Buat API login - <i>Single device</i> login dengan <i>device</i> ID - Validasi role akses - 3 Kali gagal login akun terkunci	4 hari	Done
US-2	Sebagai Admin Super, saya ingin mengelola data departemen	- CRUD departemen - UI departemen management - Validasi data	1 hari	Done
US-3	Sebagai Admin Super, saya ingin mengelola data jabatan	- CRUD jabatan - UI jabatan management - Validasi data	1 hari	Done
US-4	Sebagai Admin Super, saya ingin mengelola data peran	- CRUD peran dan izin fitur - UI peran - Validasi data	3 hari	Done
US-5	Sebagai user, saya ingin logout dari sistem	- API logout - Clear token - UI logout button	1 hari	Done
US-6	Sebagai user, saya ingin reset password jika lupa	- <i>Forgot password flow</i> - Email verification - <i>Reset password form</i> - UI <i>reset password</i>	2 hari	Done

3. Manajemen Karyawan, Log Aktivitas, Dan Kantor (Minggu 5-6)

Tabel 3.3 *User Story Sprint* Manajemen Karyawan, Log Aktivitas, Dan Kantor

ID	User Story	Task	Estimasi	Status
US-1	Sebagai Admin Super, saya ingin mengelola data karyawan	- CRUD karyawan - UI karyawan - Validasi data	3 hari	Done
US-2	Sebagai Admin Super, saya ingin melihat log aktivitas user	- <i>Logging system</i> - UI log viewer	5 hari	Done

US-3	Sebagai Admin Super, saya ingin mengelola data kantor	- Tambah dan edit kantor - Input koordinat GPS kantor - Radius absensi kantor	3 hari	Done
------	---	---	--------	------

4. Sistem Absensi, Mengelola *Profile*, Dan Pengaturan (Minggu 7-8)

Tabel 3.4 *User Story Sprint* Sistem Absensi, Mengelola *Profile*, Dan Pengaturan

ID	User Story	Task	Estimasi	Status
US-1	Sebagai Admin Office/Teknisi, saya ingin absen dengan GPS dan video	- API absensi - Capture GPS <i>location</i> - <i>Record</i> video <i>verification</i> - Upload video ke server - Validasi radius kantor	4 hari	Done
US-2	Sebagai user, saya ingin melihat daftar absensi	- API get attendance list - UI daftar absensi - <i>Filter by role</i>	1 hari	Done
US-3	Sebagai Admin Super/ <i>Office</i> /Teknisi, saya ingin mengelola profil	- View profil user - Edit profil - Update email dan <i>password</i> user	2 hari	Done
US-4	Sebagai user, saya ingin mengatur tema dan bahasa	- Setting tema - Setting bahasa	2 hari	Done
US-5	Testing dan bug fixing absensi	- Test GPS accuracy - Test video recording - Test validasi radius - Fix bugs	2 hari	Done

5. Sistem Penugasan (Minggu 9-10)

Tabel 3.5 *User Story Sprint* Sistem Penugasan

ID	User Story	Task	Estimasi	Status
US-1	Sebagai Admin Super/ <i>Office</i> , saya ingin mengelola tugas	- CRUD tugas - Assign tugas ke teknisi - Set lokasi tugas - Set deadline tugas	4 hari	Done
US-2	Sebagai Teknisi, saya ingin upload lampiran pengerjaan tugas	- Upload foto progress - Upload dokumen - Validasi data	3 hari	Done
US-3	Sebagai Admin, saya ingin melihat progress tugas	- UI tugas - Status tracking - Timeline tugas	2 hari	Done
US-4	Testing tugas	- Integration testing - Bug fixing	3 hari	Done

6. Sistem Cuti (Minggu 11-12)

Tabel 3.6 *User Story Sprint* Sistem Cuti

ID	User Story	Task	Estimasi	Status
US-1	Sebagai Admin Office/Teknisi, saya ingin mengajukan cuti	- Form pengajuan cuti - Pilih tanggal mulai-selesai - Input alasan cuti - Submit pengajuan	2 hari	Done
US-2	Sebagai Admin Office, saya ingin verifikasi cuti tahap 1	- List pengajuan cuti - Detail pengajuan - <i>Approve/reject</i> tahap 1 - Input catatan penolakan	2 hari	Done
US-3	Sebagai Admin Super, saya ingin verifikasi cuti tahap 2	- List pengajuan cuti <i>approved</i> tahap 1 - Final approval/reject - Input catatan penolakan	2 hari	Done
US-4	Sebagai user, saya ingin notifikasi status cuti	- Push notification FCM	3 hari	Done
US-5	Testing sistem cuti	- Test approval flow	3 hari	Done

		- Test notification - Bug fixing		
--	--	-------------------------------------	--	--

7. Sistem Lembur (Minggu 13-14)

Tabel 3.7 *User Story Sprint* Sistem Lembur

ID	User Story	Task	Estimasi	Status
US-1	Sebagai Admin Office/Teknisi, saya ingin mengajukan lembur	- Form pengajuan lembur - Input tanggal dan jam lembur - Input alasan lembur - Input estimasi durasi - Submit pengajuan	2 hari	Done
US-2	Sebagai Admin Office, saya ingin verifikasi lembur tahap 1	- List pengajuan lembur - Detail pengajuan - <i>Approve/reject</i> tahap 1 - Input catatan penolakan	2 hari	Done
US-3	Sebagai Admin Super, saya ingin verifikasi lembur tahap 2	- List lembur <i>approved</i> tahap 1 - Final <i>approval/reject</i> - Input catatan penolakan	2 hari	Done
US-4	Sebagai user, saya ingin notifikasi status lembur	- Push notification FCM	3 hari	Done
US-5	Testing sistem lembur	- Test approval <i>flow</i> - Test calculation - Bug fixing	3 hari	Done

8. Sistem Penggajian, Potongan Gaji, Reset Massal, Dan Pengingat (Minggu 15-16)

Tabel 3.8 *User Story Sprint* Sistem Penggajian, Potongan Gaji, Reset Massal, Dan Pengingat

ID	User Story	Task	Estimasi	Status
US-1	Sebagai Admin Super/Office, saya ingin mengelola potongan gaji	- CRUD potongan gaji	2 hari	Done

		- Perhitungan <i>Persentase</i> otomatis di gaji		
US-2	Sebagai Admin Super/Office, saya ingin mengelola gaji	- Generate slip gaji - Perhitungan otomatis - Integrasi absensi	3 hari	Done
US-3	Sebagai Admin Super/Office, saya ingin export gaji ke Excel	- <i>Export function</i> - <i>Filter periode export</i>	1 hari	Done
US-4	Sebagai Admin Super/Office, saya ingin kelola pengingat pembayaran	- CRUD pengingat - <i>Auto send email</i> - <i>Cron job scheduler</i> - Kirim email pengingat	2 hari	Done
US-5	Sebagai Admin Super, saya ingin reset data masal	- <i>Reset data</i> (log, absensi, cuti, lembur, gaji, tugas) - Hapus data di storage	1 hari	Done
US-6	Testing gaji, reset massal, dan pengingat	- <i>Test calculation accuracy</i> - Test reset massal - Test email reminder - <i>Bug fixing</i>	3 hari	Done

9. Sistem Hapus device, Buka Akun, Dan Deployment (Minggu 17-18)

Tabel 3.9 Sistem Hapus device, Buka Akun, Dan Deployment

ID	User Story	Task	Estimasi	Status
US-1	Sebagai Admin Super, saya ingin hapus device user	- List device terdaftar per user - <i>Delete device</i>	1 hari	Done
US-2	Sebagai Admin Super, saya ingin buka akun terkunci	- List akun terkunci - Buka akun terkunci	1 hari	Done
US-3	Integrasi dan testing menyeluruh	- Integration testing semua modul - <i>Cross-module testing</i> - <i>Performance testing</i> - <i>Security testing</i>	3 hari	Done
US-4	Bug fixing dan optimization	- <i>Fix critical bugs</i> - <i>Code refactoring</i>	2 hari	Done

US-5	Migrasi ke server kantor	- Deploy ke server kantor - <i>fix bug production</i>	7 hari	Done
------	--------------------------	--	--------	------

3.1.2 Proses Bisnis Berjalan

Sebelum adanya sistem HRIS, proses manajemen karyawan di Perusahaan Penyedia Jasa Teknologi Informasi (IT) masih dilakukan secara manual dan semi-otomatis. Untuk absensi, sebagian besar karyawan melakukan pencatatan kehadiran melalui mesin *fingerprint* yang tersedia di kantor. Namun, bagi teknisi yang ditugaskan melakukan instalasi atau pekerjaan khusus di perusahaan klien, absensi dilakukan dengan cara mengirimkan foto diri di lokasi melalui aplikasi WhatsApp. Mekanisme ini dinilai kurang efektif karena data absensi tidak langsung terintegrasi ke dalam sistem, sehingga membutuhkan verifikasi tambahan dari *Admin*. Selain itu, proses pengajuan cuti maupun lembur juga masih dilakukan secara manual melalui pesan WhatsApp, tanpa adanya sistem digital yang dapat mencatat, memverifikasi, dan merekap data secara otomatis.

Permasalahan lain juga muncul dalam proses pengajian. Saat ini, perhitungan gaji, lembur, potongan, serta rekap absensi masih dikerjakan secara manual oleh *Admin*. Kondisi ini berisiko menimbulkan *human error*, memakan waktu lebih lama, dan tidak efisien karena data dari berbagai sumber (absensi *fingerprint*, laporan teknisi, dan pengajuan cuti/lembur via WhatsApp) harus dikompilasi secara terpisah. Keterbatasan tersebut menunjukkan bahwa sistem yang berjalan saat ini belum mampu mendukung kebutuhan perusahaan untuk mengelola karyawan secara efektif dan transparan. Oleh karena itu, diperlukan perancangan sistem HRIS yang dapat mengintegrasikan seluruh proses bisnis mulai dari absensi, pengajuan cuti dan lembur, hingga penggajian dalam satu *platform* digital.

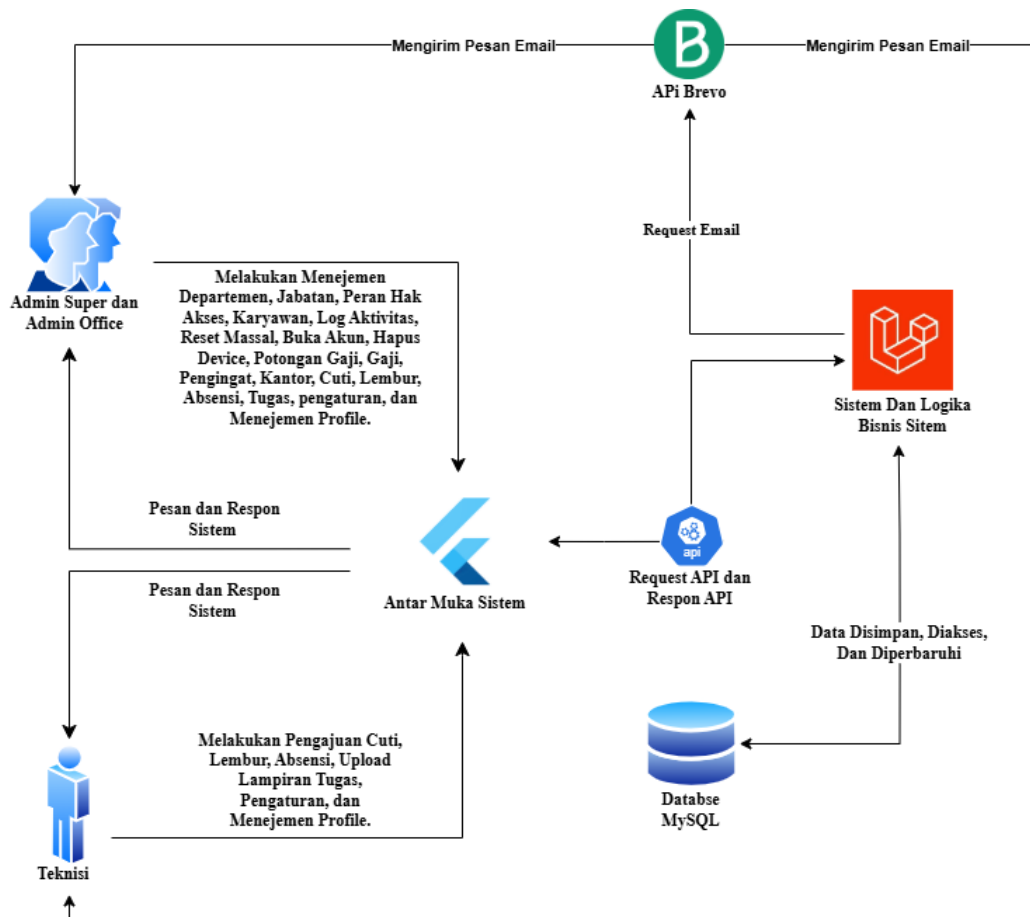
3.1.3 Gambaran Umum Sistem

Sistem HRIS yang dikembangkan pada Perusahaan Penyedia Jasa Teknologi Informasi (IT) melibatkan tiga aktor utama, yaitu *Admin Super*, *Admin Office*, dan Teknisi. *Admin Super* berperan dalam pengawasan penuh terhadap seluruh modul sistem, sedangkan *Admin Office* mengelola data terkait cuti, lembur,

absensi, tugas, potongan gaji, gaji, pengingat, kantor, pengaturan, profil, dan lupa kata sandi. Sementara itu, Teknisi memiliki akses untuk absensi, cuti, lembur, mengunggah laporan tugas, pengaturan, profil, dan lupa kata sandi.

Interaksi antar pengguna dilakukan melalui antarmuka sistem yang dirancang menggunakan *framework Flutter*. Pada sisi logika bisnis, sistem menggunakan *framework Laravel* yang berfungsi untuk memproses seluruh aturan dan alur kerja yang berlaku. Setiap permintaan dari pengguna yang dikirim melalui antarmuka sistem akan diteruskan ke server melalui API, kemudian diolah oleh logika bisnis sesuai kebutuhan. Data yang dihasilkan, baik berupa absensi, pengajuan cuti/lembur, maupun informasi penggajian, disimpan dan dikelola pada *database MySQL*. Dengan mekanisme ini, seluruh data karyawan dapat diakses, diperbarui, dan diintegrasikan secara *real-time*, sehingga meminimalkan potensi inkonsistensi data.

Untuk mendukung notifikasi otomatis, sistem diintegrasikan dengan API Brevo sebagai media pengiriman *email*. Setiap aktivitas penting, seperti pengingat jadwal dan link untuk ganti kata sandi dapat langsung dikirimkan melalui *email* kepada pengguna terkait. Proses ini dilakukan dengan menghubungkan logika bisnis sistem ke API Brevo yang kemudian meneruskan pesan ke alamat *email* karyawan. Dengan adanya integrasi ini, sistem HRIS tidak hanya berfungsi sebagai sarana pencatatan dan manajemen data, tetapi juga sebagai media komunikasi otomatis yang mempercepat arus informasi antara perusahaan dan karyawan.

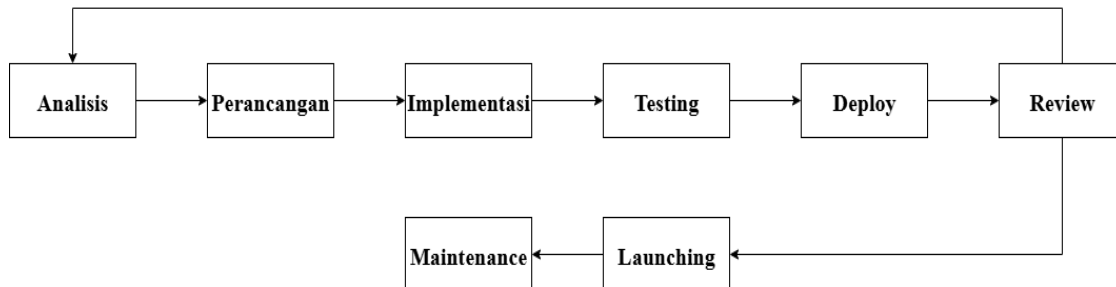


Gambar 3.1 Gambaran Umum Sistem

3.2 Perancangan

Perancangan sistem HRIS pada Perusahaan Penyedia Jasa Teknologi Informasi (IT) menggunakan metodologi *Agile* dengan *framework Scrum*. Pemilihan metode ini didasarkan pada kebutuhan akan fleksibilitas dalam pengembangan serta pentingnya keterlibatan aktif klien dalam memberikan *feedback* di setiap tahap pembangunan sistem. Mengingat Perusahaan Penyedia Jasa Teknologi Informasi (IT) tidak memiliki divisi khusus yang berfokus pada pengembangan aplikasi internal, maka diperlukan proses iteratif yang memungkinkan adanya evaluasi dan penyesuaian secara berkala berdasarkan masukan dari pihak perusahaan. Dengan pendekatan *Agile Scrum*, pengembangan dilakukan dalam siklus-siklus sprint berdurasi 2 minggu, sehingga setiap fitur yang dibangun dapat segera diuji, dievaluasi, dan disesuaikan agar benar-benar memenuhi kebutuhan operasional perusahaan.

Pada sub bab berikut ini akan dibahas kebutuhan fungsional dan non-fungsional yang harus dipenuhi oleh aplikasi HRIS pada Perusahaan Penyedia Jasa Teknologi Informasi (IT).



Gambar 3.2 Metode Pengembangan Sistem *Agile*

3.2.1 Kebutuhan *Fungsional*

Kebutuhan fungsional merupakan sekumpulan layanan atau fungsi utama yang wajib disediakan oleh sistem agar dapat berjalan sesuai dengan tujuan yang telah ditetapkan. Fitur-fitur ini dirancang untuk mendukung aktivitas manajemen karyawan secara digital, mulai dari autentikasi, absensi, penggajian, pelaporan dan lain-lain. Berikut adalah kebutuhan fungsional pada aplikasi HRIS studi kasus Perusahaan Penyedia Jasa Teknologi Informasi (IT).

Tabel 3.10 Kebutuhan *Fungsional*

Kode	Kebutuhan <i>Fungsional</i>
FR-01	<i>Admin Super, Admin Office, dan Teknisi dapat melakukan login. Sistem melakukan autentikasi pengguna dengan single device login menggunakan device ID kepada aktor yang login melalui aplikasi mobile. Admin Office login melalui web dan Teknisi hanya dapat login via aplikasi mobile. Admin Super dapat login dari web maupun aplikasi mobile tanpa batasan single device.</i>
FR-02	<i>Admin Super dapat mengelola data departemen.</i>
FR-03	<i>Admin Super dapat mengelola data jabatan.</i>
FR-04	<i>Admin Super dapat mengelola data peran serta fitur yang dapat diakses oleh masing-masing peran.</i>
FR-05	<i>Admin Super dapat mengelola data karyawan.</i>

FR-06	<i>Admin Super</i> dapat melihat data <i>log</i> aktivitas <i>user</i> .
FR-07	<i>Admin Super</i> dapat mengelola data kantor.
FR-08	<i>Admin Super</i> dapat mereset data secara masal pada tabel <i>log</i> aktivitas, absensi, cuti, lembur, gaji, dan tugas.
FR-09	<i>Admin Super</i> dan <i>Admin Office</i> dapat mengelola data potongan gaji.
FR-10	<i>Admin Super</i> dan <i>Admin Office</i> dapat mengelola data gaji.
FR-11	<i>Admin Super</i> dan <i>Admin Office</i> dapat mengekspor data gaji ke Excel.
FR-12	<i>Admin Super</i> dan <i>Admin Office</i> dapat mengelola pengingat pembayaran yang dikirim otomatis H-7 melalui <i>email</i> .
FR-13	<i>Admin Super</i> dan <i>Admin Office</i> dapat mengelola data tugas.
FR-14	Teknisi dapat mengupload lampiran pengerjaan tugas.
FR-15	<i>Admin Super</i> , <i>Admin Office</i> , dan teknisi dapat melihat daftar absensi. <i>Admin Super</i> dapat melihat seluruh daftar absen karyawan sedangkan Teknisi dan <i>Admin Office</i> hanya dapat melihat daftar absennya sendiri.
FR-16	<i>Admin Office</i> dan Teknisi dapat melakukan absensi dengan verifikasi lokasi GPS dan <i>Video</i> .
FR-17	<i>Admin Office</i> dan Teknisi dapat mengajukan cuti.
FR-18	<i>Admin Office</i> dapat memverifikasi atau menolak pengajuan cuti tahap awal.
FR-19	<i>Admin Super</i> dapat memverifikasi atau menolak cuti tahap akhir.
FR-20	<i>Admin Office</i> dan Teknisi dapat mengajukan lembur.
FR-21	<i>Admin Office</i> dapat memverifikasi atau menolak pengajuan lembur tahap awal.
FR-22	<i>Admin Super</i> dapat memverifikasi atau menolak pengajuan lembur tahap akhir.
FR-23	<i>Admin Super</i> , <i>Admin Office</i> , dan Teknisi dapat mengelola profil.
FR-24	<i>Admin Super</i> , <i>Admin Office</i> , dan Teknisi dapat mengelola pengaturan tema dan bahasa aplikasi.
FR-25	<i>Admin Super</i> dapat menghapus daftar <i>device user</i> yang terdaftar.

FR-26	<i>Admin Super</i> dapat membuka akun <i>user</i> yang terkunci.
FR-27	<i>Admin Super, Admin Office</i> , dan Teknisi dapat <i>logout</i> .
FR-28	<i>Super Admin, Admin Office</i> , dan Teknisi dapat melakukan perubahan kata sandi jika lupa <i>password</i> .

3.2.2 Kebutuhan *Non-Fungsional*

Kebutuhan *non-fungsional* merupakan spesifikasi teknis yang tidak secara langsung tampak sebagai fitur, namun sangat berpengaruh terhadap kualitas, keandalan, dan kinerja sistem. Kebutuhan ini memastikan bahwa aplikasi HRIS dapat digunakan secara optimal sesuai standar yang ditetapkan. Berikut adalah kebutuhan *non-fungsional* pada sistem yang dikembangkan di Perusahaan Penyedia Jasa Teknologi Informasi (IT):

Tabel 3.11 Kebutuhan *Non-Fungsional*

Kode	Kebutuhan <i>Non-Fungsional</i>
NFR-01	Sistem harus mampu menangani peningkatan jumlah permintaan secara bertahap tanpa mengalami penurunan kinerja yang signifikan.
NFR-02	Sistem harus memiliki mekanisme keamanan untuk melindungi data sensitif dan mencegah serangan umum seperti <i>SQL Injection</i> , <i>XSS</i> , dan <i>brute force</i> .

3.2.3 Diagram *Use Case*

Diagram *Use Case* adalah representasi visual yang menggambarkan interaksi antara aktor dengan sistem, serta fungsi-fungsi utama yang dapat dijalankan oleh masing-masing aktor. Diagram ini membantu memberikan gambaran awal mengenai bagaimana pengguna berinteraksi dengan sistem, apa saja hak akses yang dimiliki, dan layanan apa saja yang tersedia. Dengan demikian, *use case* berfungsi sebagai jembatan antara kebutuhan pengguna dan rancangan teknis sistem yang akan dibangun. Berikut ini merupakan *use case* Diagram untuk aplikasi HRIS studi kasus Perusahaan Penyedia Jasa Teknologi Informasi (IT):

Aplikasi HRIS yang digambarkan pada *use case* diagram ini dirancang dengan melibatkan tiga aktor utama, yaitu *admin super*, *admin office*, dan teknisi, yang masing-masing memiliki tingkat otorisasi dan tanggung jawab berbeda. Sistem menyediakan berbagai fitur yang saling terintegrasi, mencakup proses *login*, pengelolaan data organisasi, penggajian, administrasi kehadiran, mekanisme persetujuan cuti, lembur, dan lain-lain. Ketiga aktor tersebut berinteraksi dengan sistem sesuai batas akses yang telah ditentukan dalam kebutuhan fungsional.

Admin super memiliki hak akses paling tinggi dan bertindak sebagai pengendali utama seluruh modul aplikasi. *Admin Super* dapat melakukan *login* melalui web maupun aplikasi mobile tanpa batasan perangkat, serta berwenang mengelola data inti seperti departemen, jabatan, peran hak akses, karyawan, kantor, serta *log* aktivitas. Selain itu, *Admin Super* dapat melakukan reset data secara massal untuk modul seperti absensi, cuti, lembur, tugas, gaji, dan *log* aktivitas. Dalam proses persetujuan Cuti maupun Lembur, *admin super* berperan sebagai pihak yang melakukan verifikasi pada tahap akhir. *Admin super* juga berhak mengelola data gaji, potongan gaji, melakukan ekspor gaji ke excel, menghapus *device* pengguna, membuka akun pengguna yang terkunci, serta mengatur profil. Dengan cakupan akses ini, *admin super* berfungsi sebagai penjaga keamanan, pengelola struktur organisasi, dan penentu kebijakan akhir dalam sistem.

Admin office berfokus pada operasional administratif harian. Aktor ini melakukan *login* melalui web serta menangani berbagai proses seperti pengelolaan data tugas, melakukan, dan melihat absensi, mengelola potongan gaji serta gaji, dan mengekspor data gaji ke Excel. *Admin office* juga bertanggung jawab pada tahap awal verifikasi pengajuan cuti, dan lembur. Selain itu, *admin office* dapat mengelola pengingat pembayaran, mengatur profil, serta pengaturan aplikasi. Dengan cakupan ini, *admin office* berperan sebagai pelaksana administrasi yang menjaga kelancaran kegiatan operasional kantor sehari-hari.

Teknisi memiliki akses yang lebih terbatas dan hanya dapat *login* melalui aplikasi mobile dengan verifikasi perangkat. Teknisi dapat melakukan absensi dengan validasi GPS dan *video*, melihat daftar absensi miliknya, mengajukan Cuti maupun lembur, serta mengunggah lampiran pengerjaan tugas. Selain itu, teknisi

dapat mengelola profil pribadi, mengakses pengaturan aplikasi, dan melakukan perubahan kata sandi bila diperlukan. Peran Teknisi lebih diarahkan pada pelaporan kegiatan lapangan dan pemenuhan administrasi pribadi di dalam sistem HRIS.

Secara keseluruhan, struktur otorisasi pada HRIS ini dirancang untuk menerapkan kontrol berlapis. *Admin super* sebagai pengendali penuh sistem, *admin office* sebagai pengelola proses administratif, dan teknisi sebagai pengguna operasional di lapangan. Integrasi fitur-fitur dalam *use case* diagram mendukung otomatisasi proses HR yang sebelumnya dilakukan secara manual, meningkatkan transparansi melalui *log* aktivitas, memperjelas alur persetujuan cuti dan lembur, serta menjaga keamanan data melalui pengelolaan *device* dan autentikasi berbasis perangkat. Dengan penerapan sistem ini, proses HR perusahaan menjadi lebih efisien, akurat, dan mudah dipantau.

3.2.4 Scenario Use Case

Skenario Use Case merupakan uraian langkah-langkah interaksi antara aktor dengan sistem dalam menjalankan suatu fungsi tertentu. *Skenario* ini menjelaskan bagaimana proses dimulai, alur yang terjadi ketika aktor melakukan aksi, hingga sistem memberikan respon yang sesuai. Dengan adanya *Skenario use case*, pengembang maupun pengguna dapat memahami secara rinci bagaimana sistem HRIS dirancang untuk mendukung kebutuhan organisasi. Berikut ini merupakan contoh *Skenario use case* pada aplikasi HRIS studi kasus Perusahaan Penyedia Jasa Teknologi Informasi (IT).

1. Scenario Use Case Login

Admin super memiliki fleksibilitas untuk melakukan *login* melalui berbagai perangkat, baik melalui aplikasi web dan *mobile*. Sementara itu, *admin office* dan teknisi hanya diperbolehkan melakukan *login* pada satu perangkat, yaitu perangkat pertama yang terdaftar pada sistem. Apabila akun sudah pernah digunakan untuk *login* di suatu perangkat, maka akun tersebut tidak dapat digunakan kembali untuk *login* di perangkat lain sehingga keamanan sistem tetap terjaga.

Tabel 3.12 *Scenario Use Case Login*

Nama Use Case	<i>Login</i>
Deskripsi	Semua aktor yang tersimpan di dalam <i>database</i> aplikasi dapat melakukan <i>login</i>. Jika aktor salah memasukkan kata sandi sebanyak tiga kali berturut-turut, maka akun akan terkunci secara otomatis. Namun, apabila aktor berhasil <i>login</i> sebelum mencapai tiga kali percobaan gagal, jumlah percobaan tersebut akan direset menjadi nol sehingga perhitungan percobaan <i>login</i> dimulai kembali pada <i>login</i> berikutnya. Untuk aktor yang melakukan <i>login</i> melalui aplikasi mobile, perangkat yang digunakan saat <i>login</i> pertama kali akan otomatis terdaftar di dalam <i>database</i>. Mekanisme ini bertujuan agar aktor tersebut hanya dapat <i>login</i> menggunakan satu perangkat. Sementara itu, aktor yang hanya memiliki akses <i>login</i> melalui web tidak dapat <i>login</i> melalui aplikasi mobile. Adapun aktor yang memiliki akses <i>login</i> melalui web dan aplikasi dapat menggunakan kedua <i>platform</i> tersebut tanpa pembatasan perangkat.
Aktor	<i>Admin Super, Admin office</i> , dan Teknisi.
Kondisi Awal	Semua aktor berada di halaman <i>login</i> dan memiliki akun terdaftar.
Normal Scenario	
Aksi	Sistem

Aktor mengunjungi <i>link</i> website atau aplikasi.	Menampilkan halaman <i>login</i> .
Aktor memasukkan <i>email</i> dan <i>password</i> .	-
Aktor menekan tombol <i>login</i> .	Memverifikasi <i>email</i> dan <i>password</i> , jika benar menuju halaman <i>dashboard</i> , jika salah sistem akan menampilkan notifikasi gagal.

2. Scenario Use Case Logout

Logout adalah *use case* di mana *admin super*, *admin office*, dan teknisi dapat keluar dari sistem.

Tabel 3.13 Scenario Use Case Logout

Nama Use Case	Logout
Deskripsi	Semua aktor dapat melakukan <i>logout</i> untuk keluar dari aplikasi.
Aktor	<i>Admin Super</i> , <i>Admin Office</i> , dan Teknisi.
Kondisi Awal	Semua aktor sudah melakukan <i>login</i> .
Normal Scenario	
Aksi	Sistem
Aktor memilih dan menekan tombol <i>logout</i> .	Menghapus token dari <i>database</i> dan mengeluarkan aktor dari sistem menuju halaman <i>login</i> .

3. Scenario Use Case Manajemen Jabatan

Manajemen jabatan merupakan *use case* yang hanya bisa di akses oleh *admin super* untuk menambahkan, memperbarui, dan menghapus data jabatan.

Tabel 3.14 *Scenario Use Case* Manajemen Jabatan

Nama Use Case	Manajemen Jabatan
Deskripsi	Aktor dapat mengelola data jabatan dengan menambahkan, memperbarui, dan menghapus data jabatan.
Aktor	<i>Admin Super.</i>
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu jabatan.	Sistem menampilkan halaman daftar jabatan.
Aktor melihat data jabatan.	-
Alternative Scenario 1: Menambah Data Jabatan	
Aksi	Sistem
Aktor menekan tombol tambah jabatan.	Sistem menampilkan <i>form modal</i> tambah jabatan.
Aktor mengisi data jabatan pada <i>form</i> .	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem menyimpan data ke <i>database</i> dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Alternative Scenario 2: Mengedit Data Jabatan	
Aksi	Sistem
Aktor menekan tombol edit pada salah satu jabatan.	Sistem menampilkan <i>form modal</i> edit jabatan dengan data yang ada.
Aktor mengubah data jabatan sesuai kebutuhan.	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem memperbarui data jabatan dan menampilkan notifikasi

	sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Alternative Scenario 3: Menghapus Data Jabatan	
Aksi	Sistem
Aktor menekan tombol hapus pada salah satu jabatan.	Sistem menampilkan <i>modal</i> konfirmasi penghapusan.
Aktor menekan “Iya”.	Sistem menghapus data jabatan dan menampilkan notifikasi sukses.
Aktor menekan “Tidak”.	Sistem menutup <i>modal</i> konfirmasi tanpa perubahan data.

4. Scenario Use Case Manajemen Departemen

Manajemen departemen merupakan *use case* yang hanya bisa di akses oleh *admin super* untuk mengelola data departemen.

Tabel 3.15 Scenario Use Case Manajemen Departemen

Nama Use Case	Manajemen Departemen
Deskripsi	Aktor dapat mengelola data departemen dengan menambahkan, memperbarui, dan menghapus data departemen.
Aktor	<i>Admin Super</i> .
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu departemen.	Sistem menampilkan halaman daftar departemen.
Aktor melihat data departemen.	-
Alternative Scenario 1: Menambah Data Departemen	
Aksi	Sistem

Aktor menekan tombol tambah departemen.	Sistem menampilkan <i>form modal</i> tambah departemen.
Aktor mengisi data departemen pada <i>form</i> .	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem menyimpan data ke <i>database</i> dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan <i>notifikasi error</i> .
Alternative Scenario 2: Mengedit Data Departemen	
Aksi	Sistem
Aktor menekan tombol edit pada salah satu departemen.	Sistem menampilkan <i>form modal</i> edit departemen dengan data yang ada.
Aktor mengubah data departemen sesuai kebutuhan.	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem memperbarui data departemen dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan <i>notifikasi error</i> .
Alternative Scenario 3: Menghapus Data Departemen	
Aksi	Sistem
Aktor menekan tombol hapus pada salah satu departemen.	Sistem menampilkan <i>modal</i> konfirmasi penghapusan.
Aktor menekan “Iya”.	Sistem menghapus data departemen dan menampilkan notifikasi sukses.
Aktor menekan “Tidak”.	Sistem menutup <i>modal</i> konfirmasi tanpa perubahan data.

5. Scenario Use Case Manajemen Peran Hak Akses

Manajemen peran hak akses merupakan *use case* yang hanya bisa di akses oleh *admin super* untuk mengelola peran hak akses.

Tabel 3.16 *Scenario Use Case* Manajemen Peran Hak Akses

Nama Use Case	Peran Hak Akses
Deskripsi	Aktor dapat mengelola data peran hak akses dengan menambahkan, memperbarui, dan menghapus data peran hak akses.
Aktor	<i>Admin Super.</i>
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu peran hak akses.	Sistem menampilkan halaman daftar peran hak akses.
Aktor melihat data peran hak akses.	-
Alternative Scenario 1: Menambah Data Peran Hak Akses	
Aksi	Sistem
Aktor menekan tombol tambah peran hak akses.	Sistem menampilkan <i>form modal</i> tambah peran hak akses.
Aktor mengisi data peran hak akses pada <i>form</i> .	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem menyimpan data ke <i>database</i> dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Alternative Scenario 2: Mengedit Data Peran Hak Akses	
Aksi	Sistem
Aktor menekan tombol edit pada salah satu peran hak akses.	Sistem menampilkan <i>form modal</i> edit peran hak akses dengan data yang ada.

Aktor mengubah data peran hak akses sesuai kebutuhan.	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem memperbarui data peran hak akses dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Alternative Scenario 3: Menghapus Data Peran Hak Akses	
Aksi	Sistem
Aktor menekan tombol hapus pada salah satu peran hak akses.	Sistem menampilkan <i>modal</i> konfirmasi penghapusan.
Aktor menekan “Iya”.	Sistem menghapus data peran hak akses dan menampilkan notifikasi sukses.
Aktor menekan “Tidak”.	Sistem menutup <i>modal</i> konfirmasi tanpa perubahan data.

6. Scenario Use Case Manajemen Karyawan

Manajemen karyawan merupakan *use case* yang hanya bisa di akses oleh *admin super* untuk mengelola karyawan.

Tabel 3.17 Scenario Use Case Manajemen Karyawan

Nama Use Case	Karyawan
Deskripsi	Aktor dapat mengelola data karyawan dengan menambahkan, memperbarui, dan menghapus data karyawan.
Aktor	<i>Admin Super</i> .
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem. Sudah terdapat data departemen, jabatan, dan peran hak akses.
Normal Scenario	
Aksi	Sistem

Aktor memilih menu karyawan.	Sistem menampilkan halaman daftar karyawan.
Aktor melihat data karyawan.	-
Alternative Scenario 1: Menambah Data Karyawan	
Aksi	Sistem
Aktor menekan tombol tambah karyawan.	Sistem menampilkan <i>form modal</i> tambah karyawan.
Aktor mengisi data karyawan pada <i>form</i> .	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem menyimpan data ke <i>database</i> dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Alternative Scenario 2: Mengedit Data Karyawan	
Aksi	Sistem
Aktor menekan tombol edit pada salah satu karyawan.	Sistem menampilkan <i>form modal</i> edit karyawan dengan data yang ada.
Aktor mengubah data karyawan sesuai kebutuhan.	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem memperbarui data karyawan dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Alternative Scenario 3: Menghapus Data Karyawan	
Aksi	Sistem
Aktor menekan tombol hapus pada salah satu karyawan.	Sistem menampilkan <i>modal</i> konfirmasi penghapusan.
Aktor menekan “Iya”.	Sistem menghapus data karyawan dan menampilkan notifikasi sukses.
Aktor menekan “Tidak”.	Sistem menutup <i>modal</i> konfirmasi tanpa perubahan data.

7. Scenario Use Case Manajemen Kantor

Manajemen kantor merupakan *use case* yang hanya bisa di akses oleh *admin super* untuk mengolah data kantor.

Tabel 3.18 Scenario Use Case Manajemen Kantor

Nama Use Case	Manajemen Kantor
Deskripsi	Aktor dapat mengelola data kantor dengan menambahkan dan memperbarui data kantor.
Aktor	<i>Admin Super.</i>
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu kantor.	Sistem menampilkan halaman kantor.
Aktor melihat data kantor.	-
Alternative Scenario 1: Menambah Data Kantor	
Aksi	Sistem
Aktor menekan tombol tambah data kantor.	Sistem menampilkan <i>form modal</i> tambah data kantor.
Aktor mengisi data kantor pada <i>form</i> .	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem menyimpan data ke <i>database</i> dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Alternative Scenario 2: Mengedit Data Kantor	
Aksi	Sistem
Aktor menekan tombol Edit pada salah satu kantor.	Sistem menampilkan <i>form modal</i> edit dat.

Aktor mengubah data kantor sesuai kebutuhan.	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem memperbarui data kantor dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .

8. Scenario Use Case Manajemen Potongan Gaji

Manajemen potongan gaji merupakan *use case* bisa di akses oleh *admin super* dan *admin office* untuk mengolah data potongan gaji.

Tabel 3.19 Scenario Use Case Manajemen Potongan Gaji

Nama Use Case	Manajemen Potongan Gaji
Deskripsi	Aktor dapat mengelola data potongan dengan menambahkan, memperbarui, dan menghapus data potongan gaji.
Aktor	<i>Admin Super</i> dan <i>Admin Office</i> .
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu potongan gaji.	Sistem menampilkan halaman daftar potongan gaji.
Aktor melihat data potongan gaji.	-
Alternative Scenario 1: Menambah Data Potongan Gaji	
Aksi	Sistem
Aktor menekan tombol tambah potongan gaji.	Sistem menampilkan <i>form modal</i> tambah potongan gaji.
Aktor mengisi data potongan gaji pada <i>form</i> .	-

Aktor menekan tombol <i>submit</i> .	Jika valid, sistem menyimpan data ke <i>database</i> dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Alternative Scenario 2: Mengedit Data Potongan Gaji	
Aksi	Sistem
Aktor menekan tombol edit pada salah satu potongan gaji.	Sistem menampilkan <i>form modal</i> edit potongan gaji dengan data yang ada.
Aktor mengubah potongan gaji sesuai kebutuhan.	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem memperbarui data potongan gaji dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Alternative Scenario 3: Menghapus Data Potongan Gaji	
Aksi	Sistem
Aktor menekan tombol hapus pada salah satu potongan gaji.	Sistem menampilkan <i>modal</i> konfirmasi penghapusan.
Aktor menekan “Iya”.	Sistem menghapus data potongan gaji dan menampilkan notifikasi sukses.
Aktor menekan “Tidak”.	Sistem menutup <i>modal</i> konfirmasi tanpa perubahan data.

9. Scenario Use Case Log Aktivitas

Log aktivitas merupakan *use case* yang hanya bisa di akses oleh *admin super* untuk melihat *log* aktivitas setiap *user*.

Tabel 3.20 Scenario Use Case Manajemen Log Aktivitas

Nama Use Case	Log Aktivitas
Deskripsi	<i>Admin Super</i> dapat memantau <i>log</i> aktivitas karyawan yang terdaftar di dalam <i>database</i> .

Aktor	<i>Admin Super.</i>
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu <i>log</i> aktivitas.	Sistem menampilkan halaman daftar <i>log</i> aktivitas seluruh <i>user</i> .
Aktor melihat data <i>log</i> aktivitas.	-

10. Scenario Use Case Mengelola Reset

Mengelola reset merupakan *use case* yang hanya bisa di akses oleh *Admin Super* untuk melakukan penghapusan data secara masal.

Tabel 3.21 Scenario Use Case Mengelola Reset

Nama Use Case	Mengelola Reset
Deskripsi	Aktor dapat melakukan penghapusan data secara massal pada beberapa fitur seperti <i>log</i> aktivitas, cuti, lembur, tugas, dan gaji.
Aktor	<i>Admin Super</i>
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu reset massal.	Sistem menampilkan halaman reset massal.
Aktor melihat daftar fitur yang bisa dihapus massal.	-
Alternative Scenario 1: Menghapus Massal Data	
Aksi	Sistem

Aktor menekan tombol reset pada salah satu reset massal.	Sistem menampilkan <i>modal</i> data dengan <i>dropdown</i> berdasarkan bulan yang terdapat data beserta total data.
Aktor memilih salah satu data.	Sistem menampilkan <i>modal</i> konfirmasi penghapusan.
Aktor menekan tombol "iya".	Sistem menghapus data massal dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Aktor menekan batal.	Sistem menutup <i>modal</i> konfirmasi tanpa perubahan data.

11. Scenario Use Case Manajemen Gaji

Manajemen gaji merupakan *use case* yang hanya bisa di akses oleh *admin super* dan *admin office* untuk melakukan monitoring gaji dan melakukan *update* status gaji.

Tabel 3.22 Scenario Use Case Manajemen Gaji

Nama Use Case	Manajemen Gaji
Deskripsi	<i>Admin Super</i> dan <i>Admin Office</i> dapat melakukan <i>update</i> status gaji dan monitoring gaji karyawan per bulan.
Aktor	<i>Admin Super</i> dan <i>Admin Office</i> .
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu gaji.	Sistem menampilkan halaman menu gaji.
Aktor melihat daftar gaji dari setiap <i>user</i> yang terdaftar.	-
Alternative Scenario 1: Memperbarui Status Gaji	

Aksi	Sistem
Aktor menekan tombol status pada salah satu gaji.	Sistem menampilkan <i>dropdown</i> pada kolom status.
Aktor memilih status.	-
Aktor menekan salah satu jenis status.	Sistem memperbarui data status gaji pada <i>user</i> yang dipilih.

12. Scenario Use Case Export Excel Gaji

Use case export excel gaji merupakan *use case* yang bisa diakses oleh *Admin Super* dan *admin office* untuk mengambil data gaji dalam bentuk excel.

Tabel 3.23 Scenario Use Case Export Excel Gaji

Nama Use Case	Export Excel Gaji
Deskripsi	Aktor dapat melakukan <i>export</i> data tabel gaji dalam bentuk excel.
Aktor	<i>Admin Super</i> dan <i>Admin Office</i> .
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu gaji.	Sistem menampilkan halaman menu gaji.
Aktor menekan tombol <i>Export Excel</i> .	Sistem menampilkan pilihan rentang waktu gaji.
Aktor memilih rentang waktu gaji.	-
Aktor menekan tombol <i>export</i> .	Sistem menampilkan <i>modal</i> konfirmasi <i>export</i> gaji.
Aktor menekan tombol “Iya”.	Sistem menghasilkan file excel, mengunduh, dan menampilkan notifikasi sukses.

Aktor menekan tombol “Tidak”.	Sistem menutup <i>modal</i> konfirmasi <i>export gaji</i> .
-------------------------------	---

13. Scenario Use Case Manajemen Peningat

Manajemen pengingat merupakan *use case* yang hanya bisa di akses oleh *Admin Super* dan *admin office* untuk menambahkan, memperbarui, dan menghapus data pengingat.

Tabel 3.24 Scenario Use Case Manajemen Peningat

Nama Use Case	Manajemen Peningat
Deskripsi	Aktor dapat mengelola data pengingat dengan menambahkan, memperbarui, dan menghapus data pengingat. Setiap data yang terdaftar akan dibuatkan <i>email</i> oleh sistem dan dikirim H-7 kepada PIC data tersebut.
Aktor	<i>Admin Super</i> dan <i>Admin Office</i> .
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu pengingat.	Sistem menampilkan halaman daftar pengingat.
Aktor melihat data pengingat.	-
Alternative Scenario 1: Menambah Data Peningat	
Aksi	Sistem
Aktor menekan tombol tambah pengingat.	Sistem menampilkan <i>form modal</i> tambah pengingat
Aktor mengisi data pengingat pada <i>form</i> .	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem menyimpan data ke <i>database</i> dan menampilkan notifikasi

	sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Alternative Scenario 2: Mengedit Data Peningat	
Aksi	Sistem
Aktor menekan tombol edit pada salah satu peningat.	Sistem menampilkan <i>form modal</i> edit peningat dengan data yang ada.
Aktor mengubah data peningat sesuai kebutuhan.	-
Aktor menekan tombol <i>submit</i>	Jika valid, sistem memperbarui data peningat dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Alternative Scenario 3: Menghapus Data Peningat	
Aksi	Sistem
Aktor menekan tombol hapus pada salah satu peningat.	Sistem menampilkan <i>modal</i> konfirmasi penghapusan.
Aktor menekan “Iya”.	Sistem menghapus data peningat dan menampilkan notifikasi sukses.
Aktor menekan “Tidak”.	Sistem menutup <i>modal</i> konfirmasi tanpa perubahan data.

14. Scenario Use Case Melihat Daftar Absensi

Melihat daftar absensi merupakan *use case* yang diakses untuk *admin super*, *admin office*, dan teknisi.

Tabel 3.25 Scenario Use Case Melihat Absensi

Nama Use Case	Melihat Daftar Absensi
Deskripsi	Aktor dapat melihat seluruh daftar absensi kecuali teknisi dan <i>admin office</i> . Teknisi dan <i>admin office</i> hanya dapat melihat daftar absensi dirinya sendiri.
Aktor	<i>Admin Super</i> , <i>Admin Office</i> , dan Teknisi.

Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu absensi.	Sistem menampilkan halaman absensi.
Aktor melihat data absensi.	-

15. Scenario Use Case Melakukan Absensi

Melakukan absensi merupakan *use case* yang bisa di akses oleh *admin office* dan teknisi untuk melakukan *check in* dan *check out*.

Tabel 3.26 Scenario Use Case Melakukan Absensi

Nama Use Case	Melakukan Absensi
Deskripsi	Aktor dapat melakukan <i>Check in</i> dan <i>Check Out</i> dengan GPS dan <i>video</i> berdurasi 15 detik.
Aktor	<i>Admin Office</i> dan Teknisi.
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem. Data kantor sudah diatur oleh <i>Admin Super</i> .
Normal Scenario	
Aksi	Sistem
Aktor menekan tombol <i>Check In</i> .	Sistem menampilkan <i>form modal</i> Check In.
Aktori mengisi data <i>check in</i> pada <i>form</i> .	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem menyimpan data ke <i>database</i> dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Alternative Scenario 1: Check Out	
Aksi	Sistem

Aktor menekan tombol <i>Check Out</i> .	Sistem menampilkan <i>form modal Check Out</i> .
Aktor mengisi data <i>Check Out</i> pada <i>form</i> .	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem menyimpan data ke <i>database</i> dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .

16. Scenario Use Case Manajemen Tugas

Tugas merupakan *use case* yang bisa di akses *admin super* dan *admin office*. *Use case* tugas mencakup menambahkan, memperbarui, menghapus, mengupload lampiran, dan mengubah status tugas.

Tabel 3.27 *Scenario Use Case* Manajemen Tugas

Nama Use Case	Tugas
Deskripsi	Aktor dapat melihat seluruh daftar tugas, menambahkan, memperbarui, menghapus data tugas. Teknisi hanya dapat melihat daftar tugas yang di tugaskan ke dirinya sendiri.
Aktor	<i>Admin Super</i> dan <i>Admin Office</i>
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem. Sudah terdapat data karyawan dalam <i>database</i> .
Normal Scenario	
Aksi	Sistem
Aktor dan teknisi memilih menu tugas.	Sistem menampilkan halaman tugas.
Aktor dan teknisi melihat data tugas.	-
Alternative Scenario 1: Menambah Data Tugas	

Aksi	Sistem
Aktor menekan tombol tambah tugas.	Sistem menampilkan <i>form modal</i> tambah tugas
Aktor mengisi data tugas pada <i>form</i> tugas.	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem menyimpan data ke <i>database</i> dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Alternative Scenario 2: Mengedit Data Tugas	
Aksi	Sistem
Aktor menekan tombol Edit pada salah satu tugas.	Sistem menampilkan <i>form modal</i> edit tugas dengan data yang ada.
Aktor mengubah tugas sesuai kebutuhan.	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem memperbarui data tugas dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Alternative Scenario 3: Menghapus Data Tugas	
Aksi	Sistem
Aktor menekan tombol hapus pada salah satu tugas.	Sistem menampilkan <i>modal</i> konfirmasi penghapusan.
Aktor menekan “Iya”.	Sistem menghapus data tugas dan menampilkan notifikasi sukses.
Aktor menekan “Tidak”.	Sistem menutup <i>modal</i> konfirmasi tanpa perubahan data.

17. Scenario Use Case Upload Lampiran Tugas

Upload lampiran tugas merupakan *use case* yang hanya bisa di akses teknisi.

Tabel 3.28 *Scenario Use Case Upload Lampiran Tugas*

Nama Use Case	Upload Lampiran Tugas
Deskripsi	Aktor dapat melihat seluruh tugas yang diberikan kepada akunya serta melakukan <i>upload</i> terhadap tugas tersebut.
Aktor	Teknisi.
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem. Sudah terdapat data tugas yang di tugaskan untuk dirinya.
Normal Scenario	
Aksi	Sistem
Aktor menekan tombol tambah lampiran.	Sistem menampilkan <i>modal</i> upload bukti tugas.
Aktor memilih file sebagai lampiran pengerjaan tugas.	-
Teknisi menekan tombol <i>submit</i> .	Jika valid, sistem memperbarui data tugas dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan <i>notifikasi error</i> .

18. Scenario Use Case Pengajuan Lembur

Pengajuan Lembur merupakan *use case* yang bisa diakses oleh *admin office* dan teknisi dimana dapat melakukan pengajuan lembur.

Tabel 3.29 *Scenario Use Case Pengajuan Lembur*

Nama Use Case	Pengajuan Lembur
Deskripsi	Aktor hanya dapat mengajukan lembur baru hanya jika lembur sebelumnya sudah disetujui atau ditolak. Pada halaman lembur teknisi hanya dapat melihat daftar lemburnya sendiri

	sedangkan <i>admin office</i> dapat melihat seluruh daftar lembur.
Aktor	<i>Admin Office</i> dan Teknisi.
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu lembur.	Sistem menampilkan halaman lembur.
Aktor melihat data lembur.	-
Alternative Scenario 1: Mengajukan Lembur	
Aksi	Sistem
Aktor menekan tombol tambah lembur.	Sistem menampilkan <i>form modal</i> pengajuan lembur.
Aktor mengisi data lembur pada <i>form</i> lembur.	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem menyimpan data ke <i>database</i> dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .

19. Scenario Use Case Menyetujui Lembur Awal

Menyetujui lembur awal merupakan *use case* yang bisa di akses oleh *admin office* dimana dapat melakukan persetujuan lembur tahap awal.

Tabel 3.30 Scenario Use Case Menyetujui Lembur Awal

Nama Use Case	Menyetujui Lembur Awal
Deskripsi	Aktor dapat melakukan penolakan dan persetujuan. Jika aktor melakukan penolakan pada lembur data tersebut langsung terkirim ke <i>Admin Super</i> dalam status “Tertolak”. Jika Aktor menyetujui data lembur akan terkirim ke <i>Admin</i>

	<i>Super</i> dengan status “Proses”. Jika lembur sudah di “Tertolak” tidak dapat diedit menjadi “Disetujui”. Tetapi jika sudah “Disetujui” dapat diedit menjadi “Tertolak”.
Aktor	<i>Admin Office.</i>
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem. Sudah terdapat daftar lembur yang diajukan.
Normal Scenario	
Aksi	Sistem
Aktor menekan tombol setuju pada salah satu lembur.	Sistem menampilkan <i>modal</i> persetujuan lembur.
Aktor menekan tombol “Setuju”.	Sistem memperbarui data pada lembur <i>database</i> dan menampilkan notifikasi sukses.
Aktor menekan tombol “Batal”.	Sistem menutup <i>modal</i> persetujuan lembur.
Alternative Scenario 1: Menolak Lembur	
Aksi	Sistem
Aktor menekan tombol “Tolak” pada salah satu data.	Sistem menampilkan <i>modal</i> inputan catatan revisi untuk dikirimkan kepada pengaju lembur.
Aktor mengisi catatan revisi.	-
Aktor menekan tombol <i>submit</i> .	Memperbarui data lembur pada <i>database</i> dan menampilkan notifikasi sukses.
Aktor menekan tombol “Batal”.	Sistem menutup <i>modal</i> penolakan lembur.

20. Scenario Use Case Menyetujui Lembur Akhir

Menyetujui lembur akhir merupakan *use case* yang hanya bisa di akses oleh *admin super* dimana dapat melakukan persetujuan lembur tahap akhir.

Tabel 3.31 Scenario Use Case Menyetujui Lembur Akhir

Nama Use Case	Menyetujui Lembur Akhir
Deskripsi	Aktor dapat melakukan penolakan dan persetujuan. Jika aktor melakukan penolakan pada lembur data pada <i>Super Admin</i> dan <i>Admin Office</i> akan ikut berubah menjadi “Tertolak”. Jika Aktor menyetujui data lembur akan berubah menjadi “Disetujui” pada <i>Super Admin</i> dan <i>Admin Office</i> . Jika lembur sudah “Tertolak” tidak dapat diedit menjadi “Disetujui”. Tetapi jika sudah “Disetujui” dapat diedit menjadi “Tertolak”.
Aktor	<i>Admin Super</i> .
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem. Sudah terdapat daftar lembur yang diajukan dan sudah disetujui oleh <i>Admin Office</i> .
Normal Scenario	
Aksi	Sistem
Aktor menekan tombol setuju pada salah satu lembur.	Sistem menampilkan <i>modal</i> persetujuan lembur.
Aktor menekan tombol “Setuju”.	Memperbarui data pada lembur <i>database</i> dan menampilkan notifikasi sukses.
Aktor menekan tombol “Batal”.	Sistem menutup <i>modal</i> persetujuan lembur.

Alternative Scenario 2: Menolak Lembur	
Aksi	Sistem
Aktor menekan tombol “Tolak” pada salah satu data.	Sistem menampilkan <i>modal</i> inputan catatan revisi untuk dikirimkan kepada pengaju lembur.
Aktor mengisi catatan revisi.	-
Aktor menekan tombol <i>submit</i> .	Memperbarui data lembur pada <i>database</i> dan menampilkan notifikasi sukses.
Aktor menekan tombol “Batal”.	Sistem menutup <i>modal</i> penolakan lembur.

21. Scenario Use Case Pengajuan Cuti

Pengajuan Cuti merupakan *use case* yang bisa diakses oleh *admin office* dan teknisi dimana dapat melakukan pengajuan Cuti.

Tabel 3.32 Scenario Use Case Pengajuan Cuti

Nama Use Case	Pengajuan Cuti
Deskripsi	Aktor hanya dapat mengajukan cuti baru hanya jika cuti sebelumnya sudah disetujui atau ditolak. Pada halaman cuti teknisi hanya dapat melihat daftar cutinya sendiri sedangkan <i>admin office</i> dapat melihat seluruh daftar cuti.
Aktor	<i>Admin Office</i> dan Teknisi.
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu cuti.	Sistem menampilkan halaman cuti.
Aktor melihat data cuti.	-
Alternative Scenario 1: Mengajukan Cuti	
Aksi	Sistem

Aktor menekan tombol tambah cuti.	Sistem menampilkan <i>form modal</i> pengajuan cuti.
Aktor mengisi data cuti pada <i>form</i> cuti.	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem menyimpan data ke <i>database</i> dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .

22. Scenario Use Case Menyetujui Cuti Awal

Menyetujui cuti awal merupakan *use case* yang bisa di akses oleh *admin office* dimana dapat melakukan persetujuan cuti tahap awal.

Tabel 3.33 Scenario Use Case Menyetujui Cuti Awal

Nama Use Case	Menyetujui Cuti Awal
Deskripsi	Aktor dapat melakukan penolakan dan persetujuan. Jika aktor melakukan penolakan pada cuti data tersebut langsung terkirim ke <i>Admin Super</i> dalam status “Tertolak”. Jika Aktor menyetujui data cuti akan terkirim ke <i>Admin Super</i> dengan status “Proses”. Jika cuti sudah “Tertolak” tidak dapat diedit menjadi “Disetujui”. Tetapi jika sudah “Disetujui” dapat diedit menjadi “Tertolak”.
Aktor	<i>Admin Office</i> .
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem. Sudah terdapat daftar cuti yang diajukan.
Normal Scenario	
Aksi	Sistem

Aktor menekan tombol setuju pada salah satu cuti.	Sistem menampilkan <i>modal</i> persetujuan cuti.
Aktor menekan tombol “Setuju”.	Sistem memperbarui data pada cuti <i>database</i> dan menampilkan notifikasi sukses.
Aktor menekan tombol “Batal”.	Sistem menutup <i>modal</i> persetujuan cuti.
Alternative Scenario 1: Menolak Cuti	
Aktor menekan tombol “Tolak” pada salah satu data.	Sistem menampilkan <i>modal</i> inputan catatan revisi untuk dikirimkan kepada pengaju cuti.
Aktor mengisi catatan revisi.	-
Aktor menekan tombol <i>submit</i> .	Sistem memperbarui data cuti pada <i>database</i> dan menampilkan notifikasi sukses.
Aktor menekan tombol “Batal”.	Sistem menutup <i>modal</i> penolakan cuti.

23. Scenario Use Case Menyetujui Cuti Akhir

Menyetujui cuti akhir merupakan *use case* yang hanya bisa di akses oleh *admin super* dimana dapat melakukan persetujuan cuti tahap akhir.

Tabel 3.34 Scenario Use Case Menyetujui Cuti Akhir

Nama Use Case	Menyetujui Cuti Akhir
Deskripsi	Aktor dapat melakukan penolakan dan persetujuan. Jika aktor melakukan penolakan pada cuti data pada <i>Super Admin</i> dan <i>Admin Office</i> akan ikut berubah menjadi “Tertolak”. Jika Aktor menyetujui data cuti akan berubah menjadi “Disetujui” pada <i>Super Admin</i> dan <i>Admin Office</i> . Jika cuti sudah

	“Tertolak” tidak dapat diedit menjadi “Disetujui”. Tetapi jika sudah “Disetujui” dapat diedit menjadi “Tertolak”.
Aktor	<i>Admin Super.</i>
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem. Sudah terdapat daftar cuti yang diajukan dan sudah disetujui oleh <i>Admin Office</i> .
Normal Scenario	
Aksi	Sistem
Aktor menekan tombol setuju pada salah satu cuti.	Sistem menampilkan <i>modal</i> persetujuan cuti.
Aktor menekan tombol “Setuju”.	Sistem memperbarui data pada cuti <i>database</i> dan menampilkan notifikasi sukses.
Aktor menekan tombol “Batal”.	Sistem menutup <i>modal</i> persetujuan cuti.
Alternative Scenario 1: Menolak Cuti	
Aktor menekan tombol “Tolak” pada salah satu data.	Sistem menampilkan <i>modal</i> inputan catatan revisi untuk dikirimkan kepada pengaju cuti.
Aktor mengisi catatan revisi.	-
Aktor menekan tombol <i>submit</i> .	Sistem memperbarui data cuti pada <i>database</i> dan menampilkan notifikasi sukses.
Aktor menekan tombol “Batal”.	Sistem menutup <i>modal</i> penolakan cuti.

24. Scenario Use Case Mengelola Profile

Profil merupakan *use case* yang bisa di akses *admin super*, *admin office*, dan *teknisi*. *Use case* profil mencakup mengubah data diri *email* dan *password*.

Tabel 3.35 Scenario Use Case Mengelola Profile

Nama Use Case	Mengelola Profil
Deskripsi	Use case profil dapat di akses oleh <i>Admin Super</i> , <i>Admin Office</i> , dan <i>Teknisi</i> . Pada halaman ini akan menampilkan data diri dari akun dan dapat melakukan perubahan <i>email</i> dan <i>password</i> akun yang sedang di gunakan aktor.
Aktor	<i>Admin Super</i> , <i>Admin Office</i> , dan <i>Teknisi</i> .
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu profil.	Sistem menampilkan halaman profil.
Aktor melihat data profil.	-
Alternative Scenario 1: Memperbarui Email	
Aksi	Sistem
Aktor menekan tombol perbarui <i>email</i> .	Sistem menampilkan <i>form modal email</i> .
Aktor mengisi data pada <i>form email</i> .	-
Aktor menekan tombol “ <i>submit</i> ”.	Jika valid, sistem menyimpan data ke <i>database</i> dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Aktor menekan tombol “ <i>Batal</i> ”.	Sistem menutup <i>form modal email</i> .
Alternative Scenario 2: Memperbarui Password	

Aksi	Sistem
Aktor menekan tombol perbarui <i>password</i> .	Sistem menampilkan <i>form</i> perbarui <i>password</i> .
Aktor mengisi <i>form</i> perbarui <i>password</i> .	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem menyimpan data ke <i>database</i> dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .

25. Scenario Use Case Mengelola Pengaturan

Pengaturan merupakan *use case* yang bisa di akses *admin super*, *admin office*, dan teknisi. *Use case* ini mencakup pengaturan tema dan bahasa aplikasi.

Tabel 3.36 Scenario Use Case Mengelola Pengaturan

Nama Use Case	Mengelola Pengaturan
Deskripsi	Use case pengaturan dapat diakses oleh <i>Admin Super</i> , <i>Admin Office</i> , dan Teknisi. Use case ini mencakup dalam mengubah tema dan bahasa aplikasi.
Aktor	<i>Admin Super</i> , <i>Admin Office</i> , dan Teknisi.
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu pengaturan.	Sistem menampilkan halaman pengaturan.
Aktor melihat halaman pengaturan.	-
Alternative Scenario 1 : Memperbarui Tema	
Aksi	Sistem

Aktor menekan tombol tema “Terang”.	Sistem memperbarui tema aplikasi menjadi terang dan memperbarui <i>database</i> pengaturan.
Aktor menekan tombol tema “Gelap”.	Sistem memperbarui tema aplikasi menjadi gelap dan memperbarui <i>database</i> pengaturan.
Alternative Scenario 2 : Memperbarui Bahasa	
Aksi	Sistem
Aktor menekan tombol bahasa.	Sistem menampilkan <i>dropdown</i> bahasa.
Aktor memilih bahasa dari <i>dropdown</i> .	-
Aktor memilih bahasa “Inggris”.	Sistem mengubah bahasa menjadi inggris dan memperbarui <i>database</i> pengaturan.
Aktor memilih bahasa “Indonesia”.	Sistem mengubah bahasa menjadi indonesia dan memperbarui <i>database</i> pengaturan.

26. Scenario Use Case Lupa Kata Sandi

Lupa kata sandi merupakan *use case* yang bisa di akses *admin super*, *admin office*, dan teknisi. *Use case* lupa kata sandi merupakan *use case* untuk mengubah kata sandi pada halaman *login*.

Tabel 3.37 Scenario Use Case Lupa Kata Sandi

Nama Use Case	Lupa Kata Sandi
Deskripsi	<i>Use case</i> lupa kata sandi dapat di akses oleh <i>Admin Super</i> , <i>Admin Office</i> , dan Teknisi. Pada <i>use case</i> ini <i>user</i> akan dikirim <i>email</i> untuk mengubah kata sandi dari <i>email</i> yang diinput.
Aktor	<i>Admin Super</i> , <i>Admin Office</i> , dan Teknisi.

Kondisi Awal	Semua aktor berada di halaman <i>login</i> .
Normal Scenario	
Aksi	Sistem
Aktor menekan tombol lupa kata sandi.	Sistem menampilkan <i>form modal input email</i> .
Aktor mengisi <i>form modal email</i> .	-
Aktor menekan tombol <i>submit</i> .	Jika valid, sistem mengirim <i>email</i> inputan dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .
Aktor membuka pesan <i>email</i> .	-
Aktor mengklik tautan ganti kata sandi.	Sistem menampilkan <i>form</i> ganti kata sandi.
Aktor mengisi <i>form</i> kata sandi baru.	-
Aktor menekan <i>submit</i> .	Jika valid, sistem menyimpan data ke <i>database</i> dan menampilkan notifikasi sukses. Jika tidak valid, sistem menampilkan notifikasi <i>error</i> .

27. Scenario Use Case Membuka Akun Karyawan

Membuka akun karyawan merupakan *use case* yang bisa di akses *admin super*. *Use case* membuka akun karyawan merupakan *use case* untuk membuka akun karyawna yang terkunci saat gagal *login* berturut-turut 3 kali.

Tabel 3.38 Scenario Use Case Membuka Akun Karyawan

Nama Use Case	Membuka Akun Karyawan
Deskripsi	<i>Use case</i> membuka akun karyawan dapat di akses oleh <i>Admin Super</i> . Pada <i>use case</i> ini aktor akan melakukan proses

	pembukaan akun karyawan yang terkunci di karenakan gagal <i>login</i> berturut-turut sebanyak 3 kali.
Aktor	<i>Admin Super.</i>
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem. Terdapat akun karyawan yang sudah terkunci.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu buka akun.	Sistem menampilkan halaman buka akun.
Aktor melihat halaman buka akun.	-
Alternative Scenario 1: Membuka Akun	
Aktor menekan tombol buka akun pada salah satu data.	Sistem menampilkan inputan konfirmasi buka akun.
Aktor mengisi inputan konfirmasi buka akun.	-
Aktor menekan tombol konfirmasi buka.	Sistem memperbarui data akun pada <i>database</i> dan menampilkan notifikasi sukses.
Aktor menekan tombol "Batal".	Sistem menutup <i>modal</i> buka akun.

28. Scenario Use Case Menghapus device Karyawan

Menghapus *device* karyawan merupakan *use case* yang bisa di akses *admin super*. *Use case* menghapus *device* karyawan merupakan *use case* untuk menghapus *device* id yang terdaftar pada *database*.

Tabel 3.39 Scenario Use Case Menghapus Device Karyawan

Nama Use Case	Menghapus device Karyawan
---------------	---------------------------

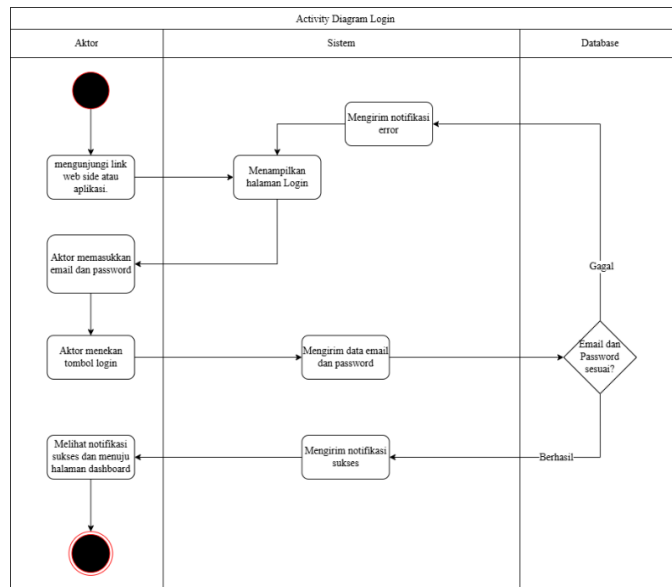
Deskripsi	<i>Use case</i> menghapus <i>device</i> karyawan dapat di akses oleh <i>Admin Super</i> . Pada <i>use case</i> ini aktor akan melakukan proses penghapusan akun karyawan yang terdaftar di <i>database</i> jika terdapat karyawan yang ingin ganti <i>device</i> .
Aktor	<i>Admin Super</i> .
Kondisi Awal	Aktor sudah berhasil <i>login</i> ke dalam sistem. Terdapat akun karyawan yang sudah terdaftar.
Normal Scenario	
Aksi	Sistem
Aktor memilih menu hapus <i>device</i> .	Sistem menampilkan halaman hapus <i>device</i> .
Aktor melihat halaman buka akun.	-
Alternative Scenario 1: Menghapus device Karyawan	
Aktor menekan tombol hapus <i>device</i> pada salah satu data.	Sistem menampilkan <i>modal</i> konfirmasi hapus <i>device</i> .
Aktor menekan tombol Konfirmasi.	Sistem memperbarui data <i>device</i> pada <i>database</i> dan menampilkan notifikasi sukses.
Aktor menekan tombol "Batal".	Sistem menutup <i>modal</i> hapus <i>device</i> .

3.2.5 Activity Diagram

Berikut merupakan *Activity* diagram aplikasi HRIS studi kasus Perusahaan Penyedia Jasa Teknologi Informasi (IT).

1. Activity Diagram Login

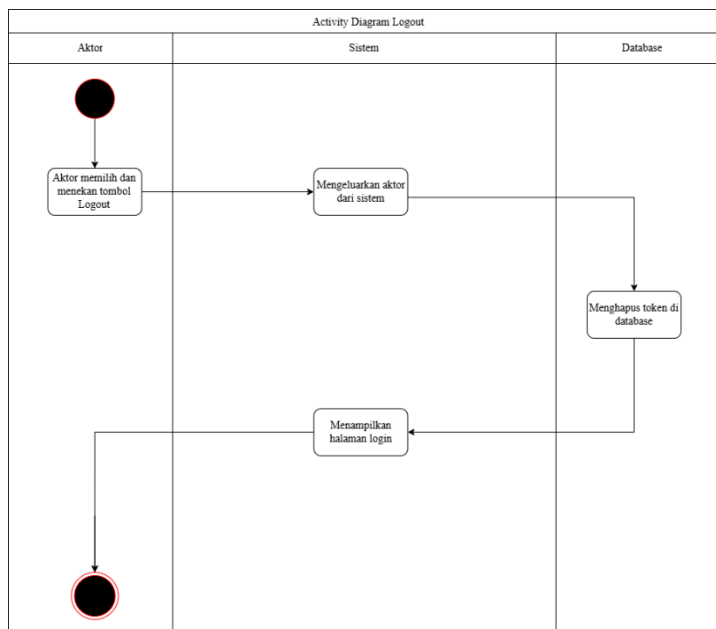
Berikut merupakan *Activity* diagram normal dari *scenario use case login*.



Gambar 3.4 Activity Diagram *Login Normal Scenario*

2. Activity Diagram Logout

Berikut merupakan Activity diagram normal dari *scenario use case logout*.

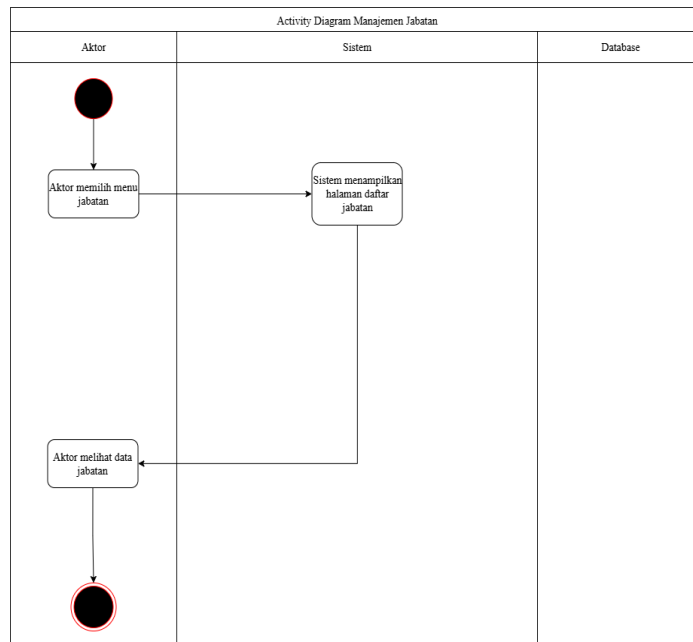


Gambar 3.5 Activity Diagram *Logout Normal Scenario*

3. Activity Diagram Manajemen Jabatan

- Normal

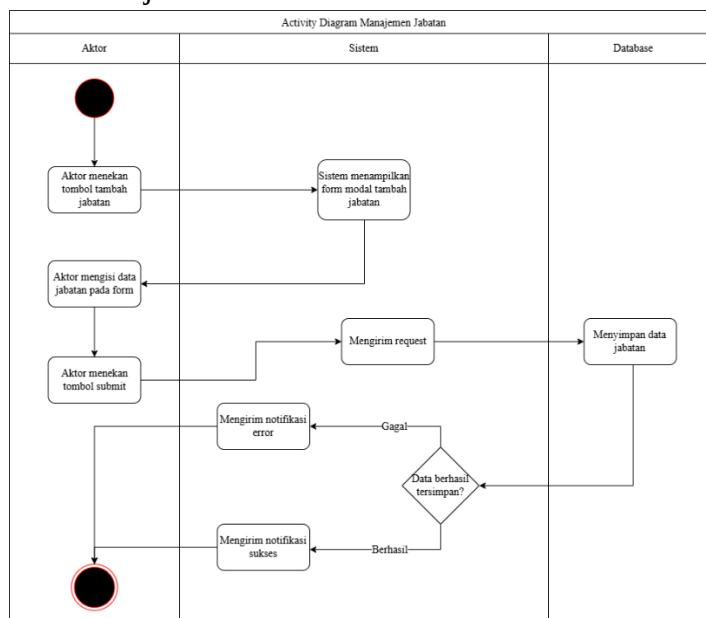
Berikut merupakan Activity diagram normal dari *scenario use case manajemen jabatan*.



Gambar 3.6 Activity Diagram Manajemen Jabatan Normal Scenario

- Alternative Scenario 1

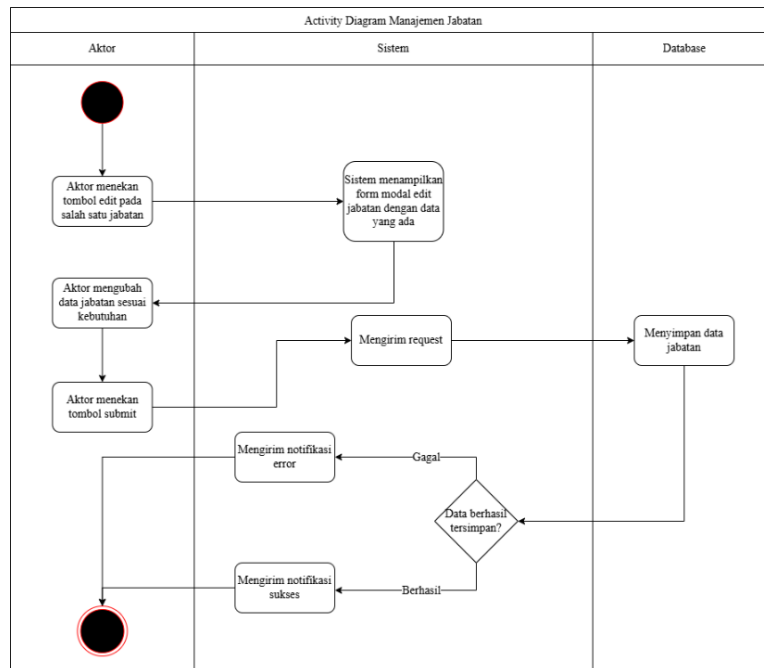
Berikut merupakan Activity diagram alternative 1 dari scenario use case menambah jabatan.



Gambar 3.7 Activity Diagram Manajemen Jabatan Alternative Scenario 1

- Alternative Scenario 2

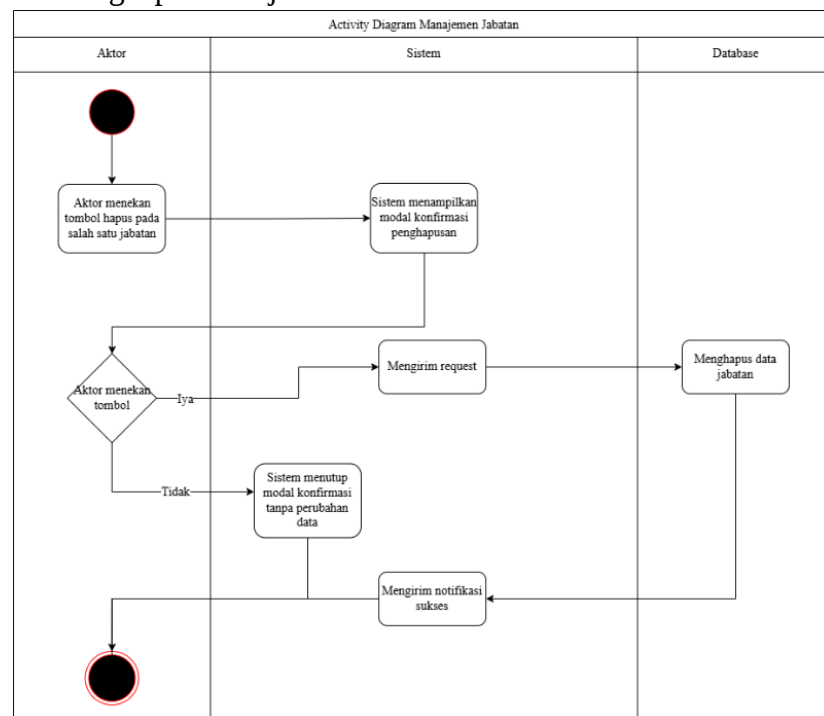
Berikut merupakan Activity diagram alternative 2 dari scenario use case edit Jabatan



Gambar 3.8 Activity Diagram Manajemen Jabatan *Alternative Scenario 2*

- *Alternative Scenario 3*

Berikut merupakan Activity diagram *alternative 3* dari *scenario use case* menghapus data jabatan.

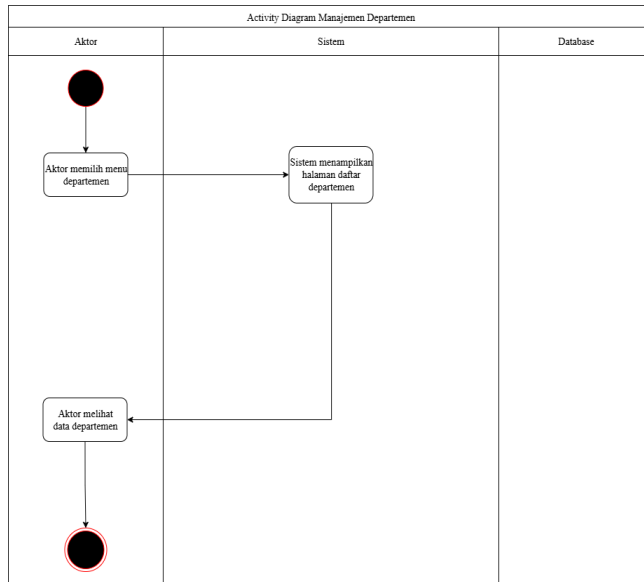


Gambar 3.9 Activity Diagram Manajemen Jabatan *Alternative Scenario 3*

4. Activity Diagram Manajemen Departemen

- Normal

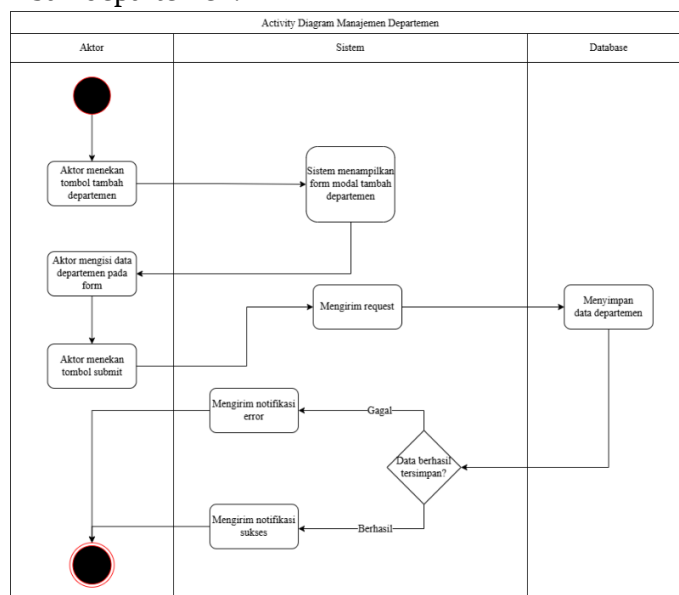
Berikut merupakan Activity diagram normal dari *scenario use case* manajemen departemen.



Gambar 3.10 Activity Diagram Manajemen Departemen Normal Scenario

- Alternative Scenario 1

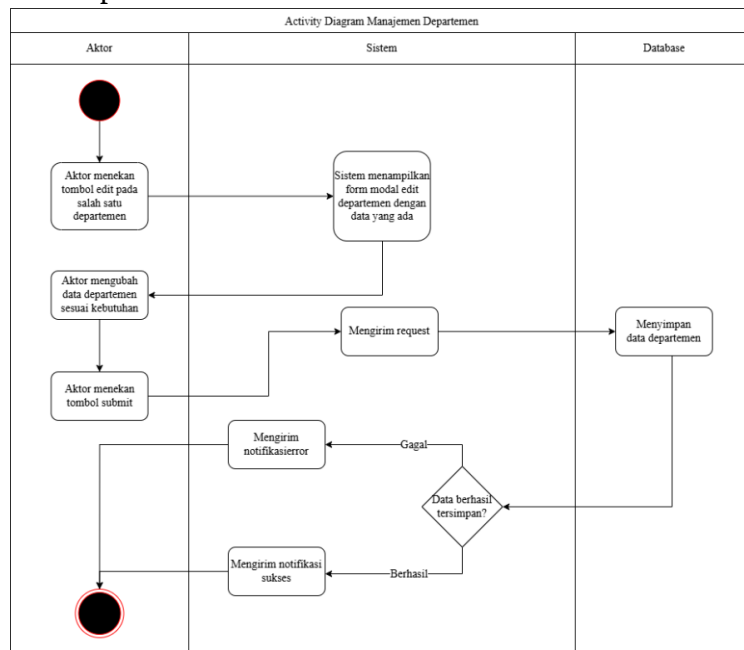
Berikut merupakan Activity diagram *alternative 1* dari *scenario use case* tambah departemen.



Gambar 3.11 Activity Diagram Manajemen Departemen Alternative Scenario 1

- *Alternative Scenario 2*

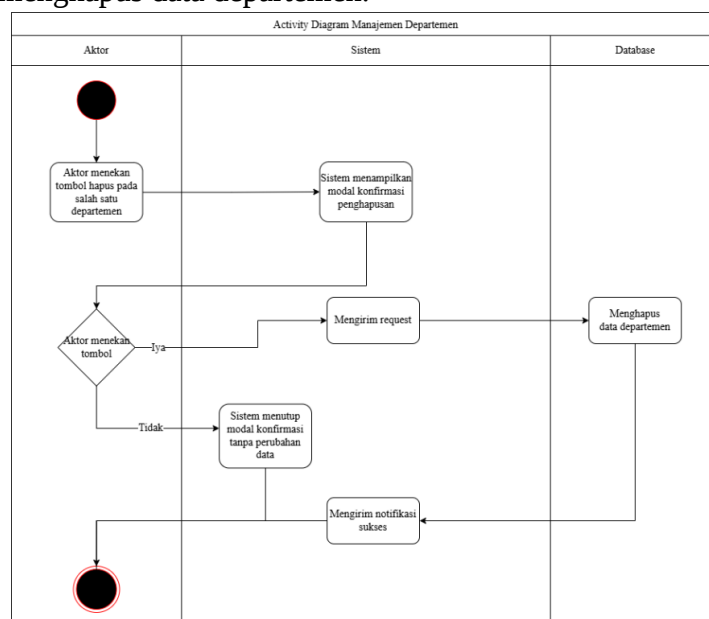
Berikut merupakan *Activity diagram alternative 2* dari *scenario use case edit departemen*.



Gambar 3.12 *Activity Diagram Manajemen Departemen Alternative Scenario 2*

- *Alternative Scenario 3*

Berikut merupakan *Activity diagram alternative 3* dari *scenario use case menghapus data departemen*.

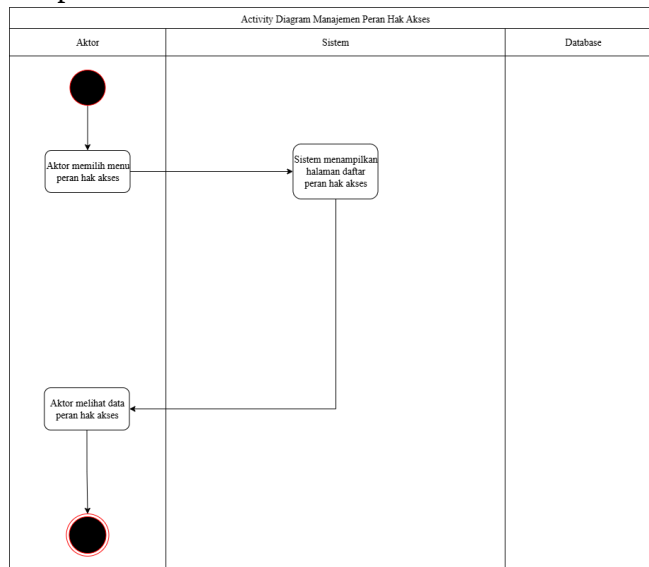


Gambar 3.13 *Activity Diagram Manajemen Departemen Alternative Scenario 3*

5. Activity Diagram Manajemen Peran Hak Akses

- Normal

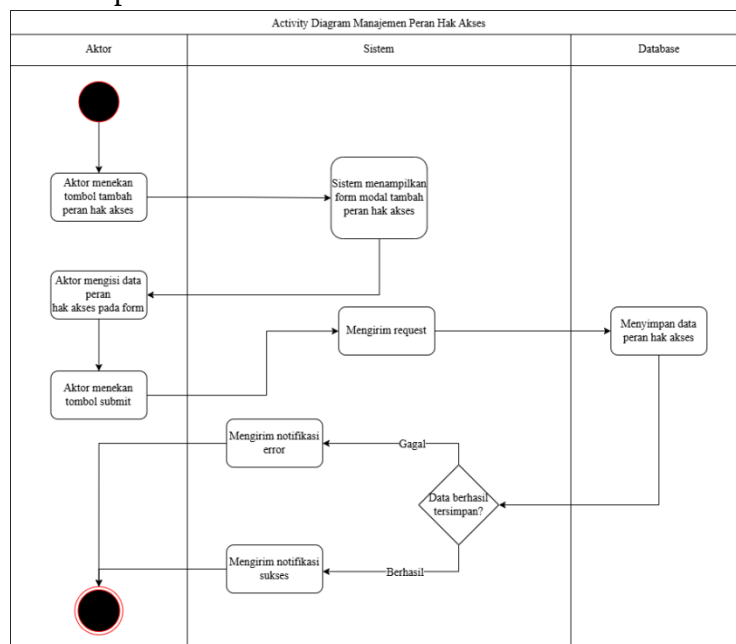
Berikut merupakan Activity diagram normal dari *scenario use case* manajemen peran hak akses.



Gambar 3.14 Activity Diagram Manajemen Peran Hak Akses Normal Scenario

- Alternative Scenario 1

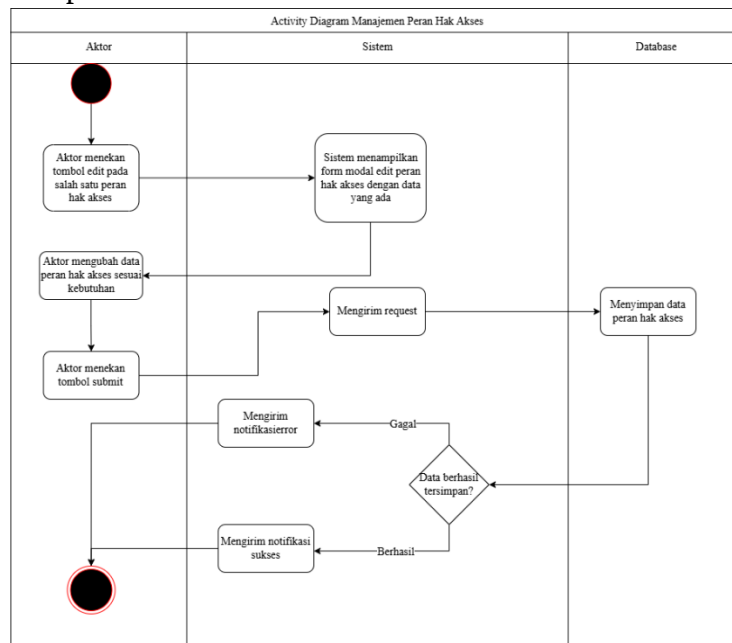
Berikut merupakan Activity diagram alternative 1 dari *scenario use case* tambah peran hak akses.



Gambar 3.15 Activity Diagram Manajemen Peran Hak Akses Normal Alternative Scenario 1

- *Alternative Scenario 2*

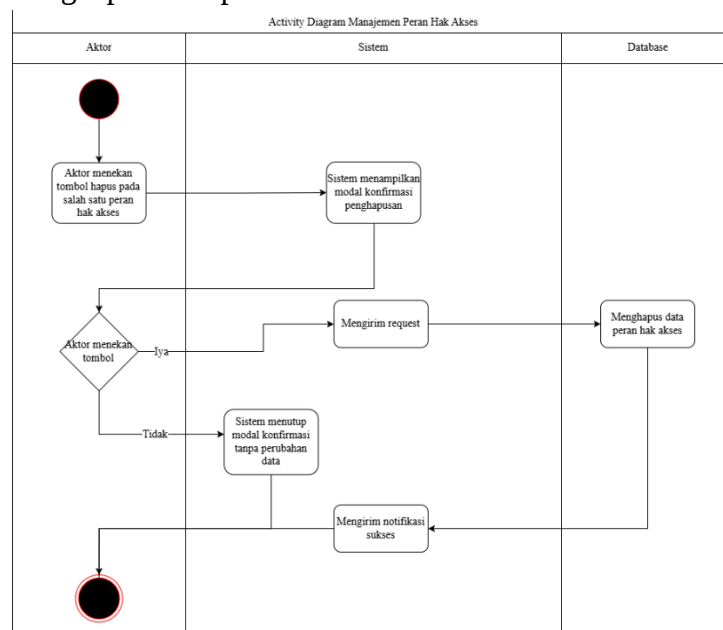
Berikut merupakan *Activity diagram alternative 2* dari *scenario use case edit peran hak akses*.



Gambar 3.16 *Activity Diagram Manajemen Peran Hak Akses Normal Alternative Scenario 2*

- *Alternative Scenario 3*

Berikut merupakan *Activity diagram alternative 3* dari *scenario use case menghapus data peran hak akses*.

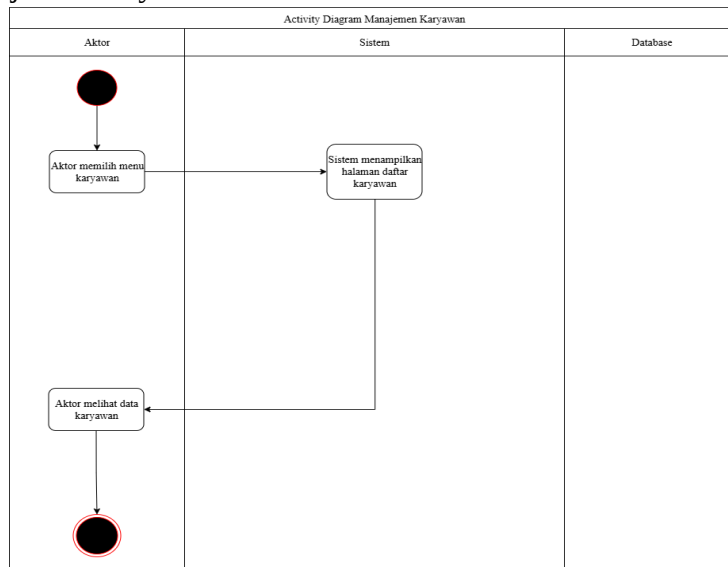


Gambar 3.17 *Activity Diagram Manajemen Peran Hak Akses Normal Alternative Scenario 3*

6. Activity Diagram Manajemen Karyawan

- Normal

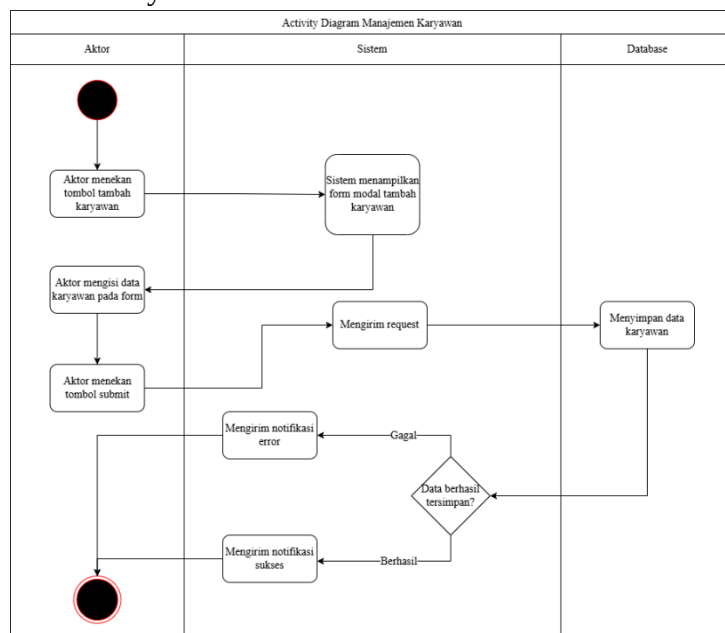
Berikut merupakan *Activity diagram normal* dari *scenario use case* manajemen karyawan.



Gambar 3.18 Activity Diagram Manajemen Karyawan Normal Scenario

- Alternative Scenario 1

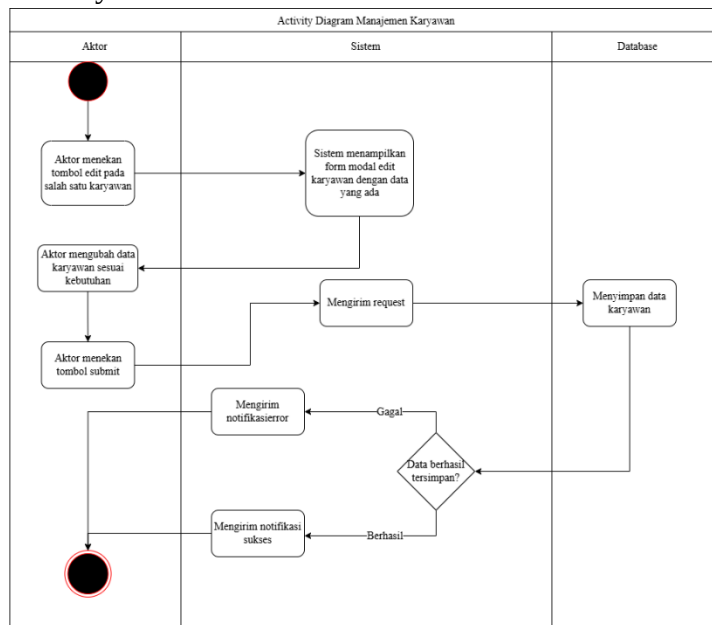
Berikut merupakan *Activity diagram alternative 1* dari *scenario use case* tambah karyawan



Gambar 3.19 Activity Diagram Manajemen Karyawan Alternative Scenario 1

- *Alternative Scenario 2*

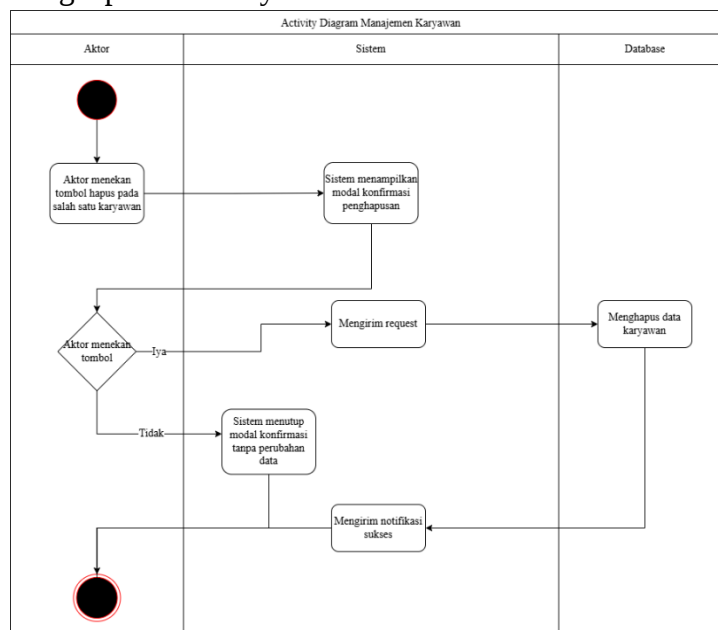
Berikut merupakan *Activity diagram alternative 2* dari *scenario use case edit karyawan*.



Gambar 3.20 *Activity Diagram Manajemen Karyawan Alternative Scenario 2*

- *Alternative Scenario 3*

Berikut merupakan *Activity diagram alternative 3* dari *scenario use case menghapus data karyawan*.

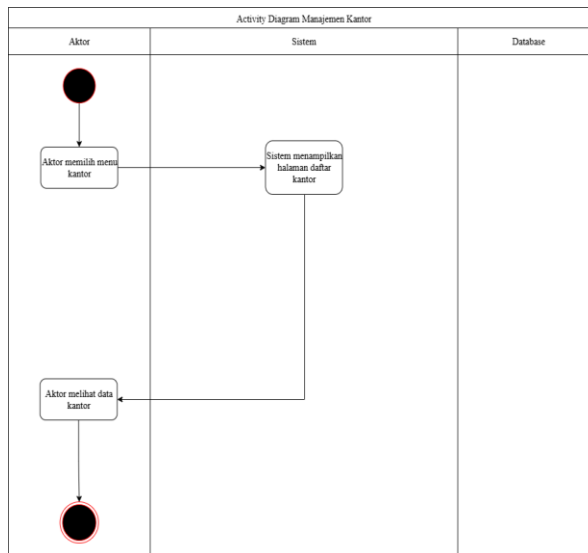


Gambar 3.21 *Activity Diagram Manajemen Karyawan Alternative Scenario 3*

7. Activity Diagram Manajemen Kantor

- Normal

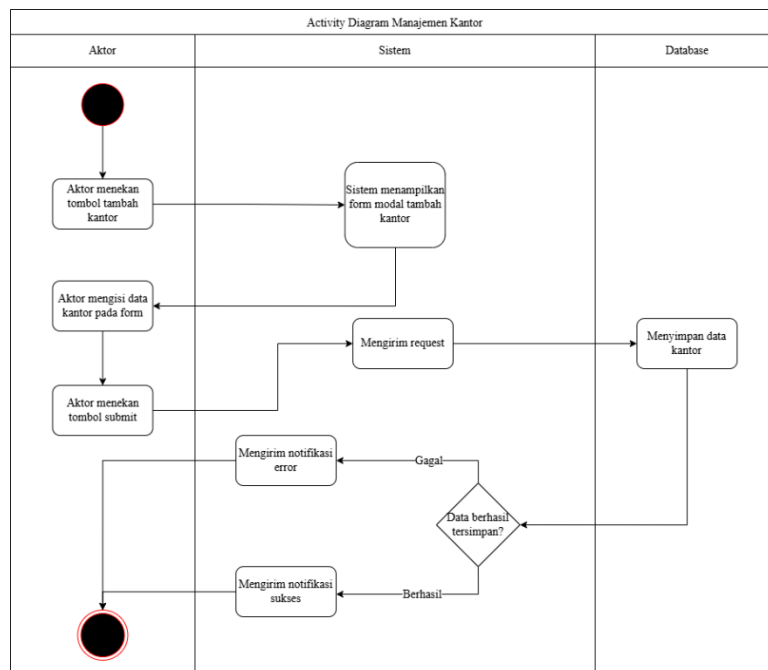
Berikut merupakan Activity diagram normal dari *scenario use case* manajemen kantor.



Gambar 3.22 Activity Diagram Manajemen Kantor Normal Scenario

- Alternative Scenario 1

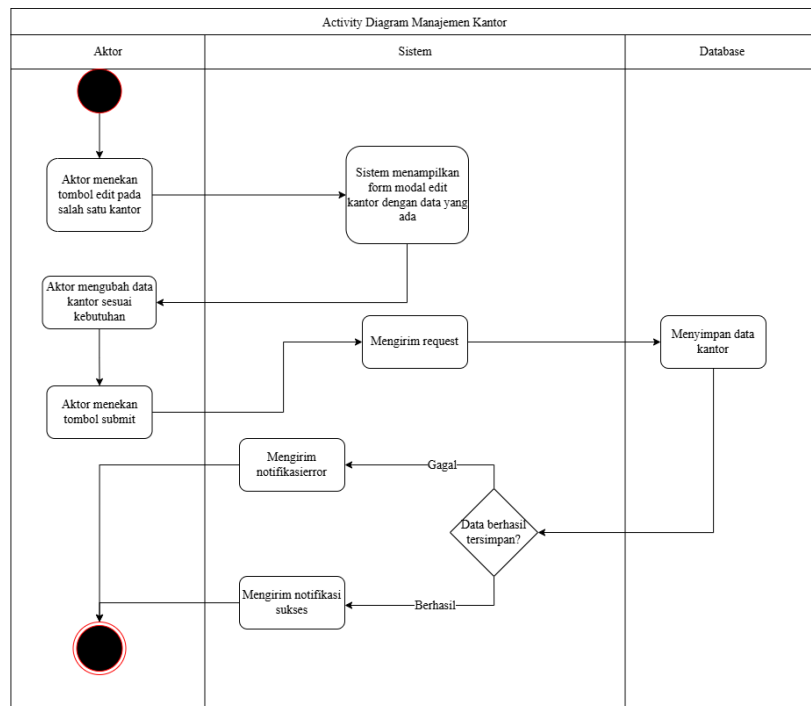
Berikut merupakan Activity diagram alternative 1 dari *scenario use case* tambah kantor.



Gambar 3.23 Activity Diagram Manajemen Kantor Alternative Scenario 1

- *Alternative Scenario 2*

Berikut merupakan *Activity diagram alternative 2* dari *scenario use case edit kantor*.

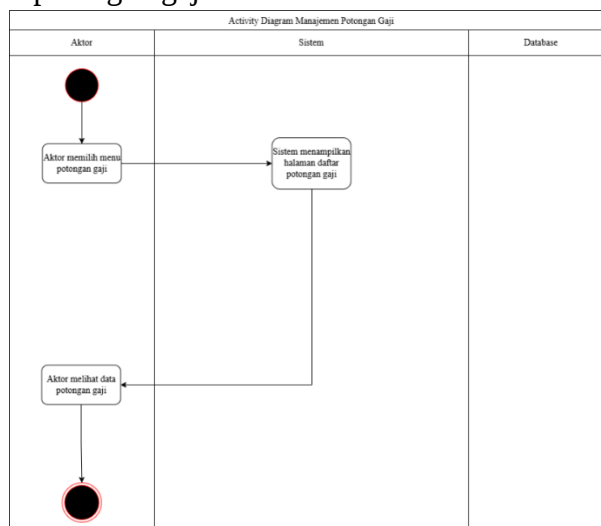


Gambar 3.24 Activity Diagram Manajemen Kantor *Alternative Scenario 2*

8. Activity Diagram Manajemen Potongan Gaji

- Normal

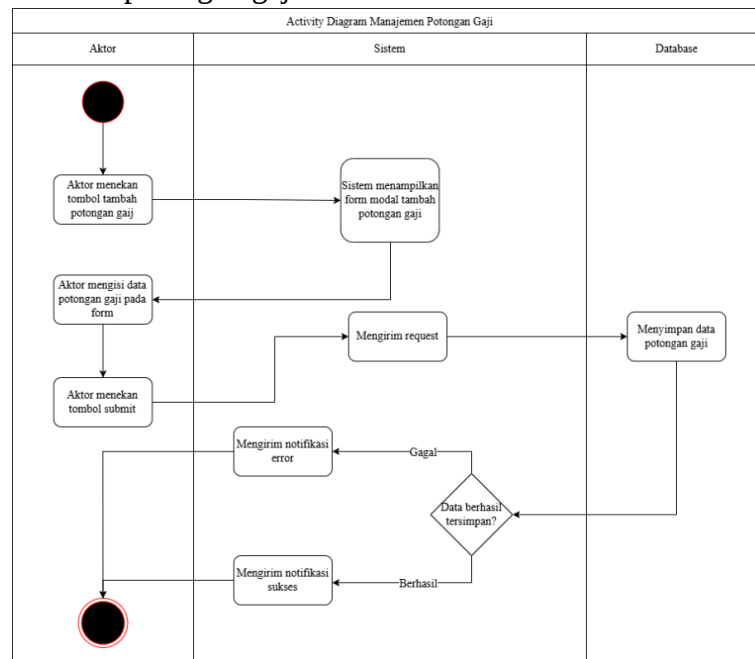
Berikut merupakan *Activity diagram normal* dari *scenario use case manajemen potongan gaji*.



Gambar 3.25 Activity Diagram Manajemen Potongan Gaji Normal *Scenario*

- *Alternative Scenario 1*

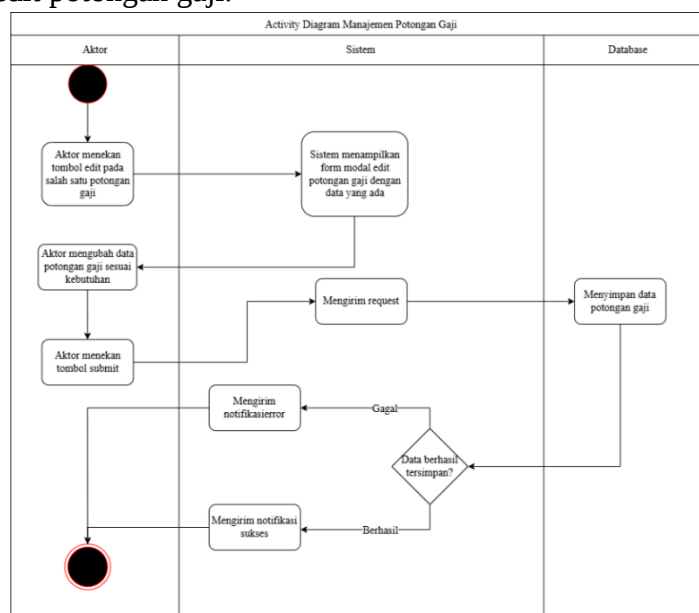
Berikut merupakan *Activity diagram alternative 1* dari *scenario use case* tambah potongan gaji.



Gambar 3.26 *Activity Diagram Manajemen Potongan Gaji Alternative Scenario 1*

- *Alternative Scenario 2*

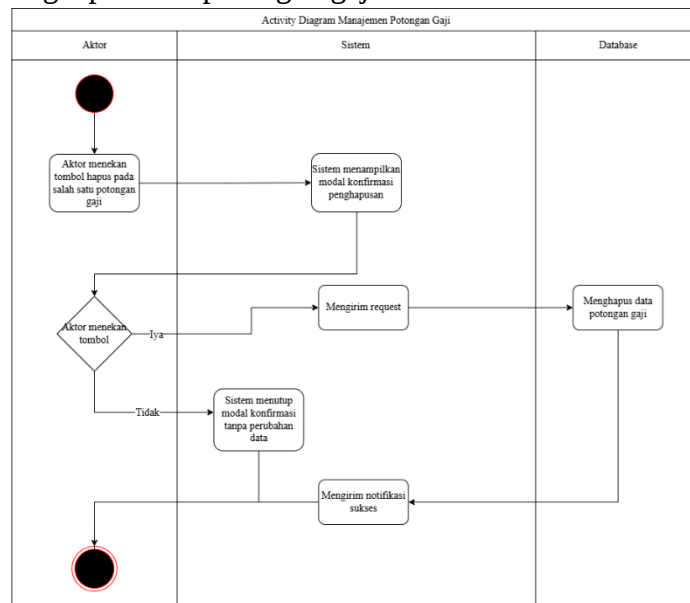
Berikut merupakan *Activity diagram alternative 2* dari *scenario use case* edit potongan gaji.



Gambar 3.27 *Activity Diagram Manajemen Potongan Gaji Alternative Scenario 2*

- *Alternative Scenario 3*

Berikut merupakan *Activity diagram alternative 3* dari *scenario use case* menghapus data potongan gaji.

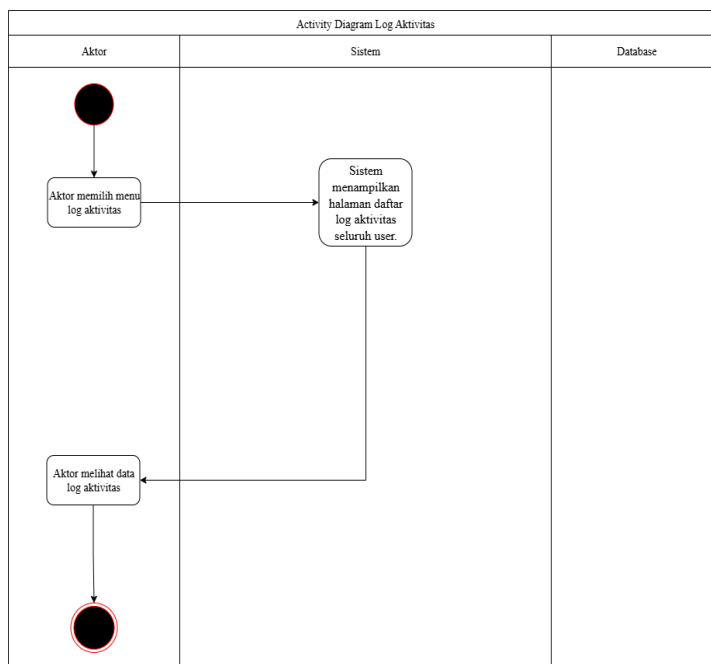


Gambar 3.28 Activity Diagram Manajemen Potongan Gaji
Alternative Scenario 3

9. Activity Diagram Log Aktivitas

- Normal

Berikut merupakan *Activity diagram normal* dari *scenario use case* log aktivitas.

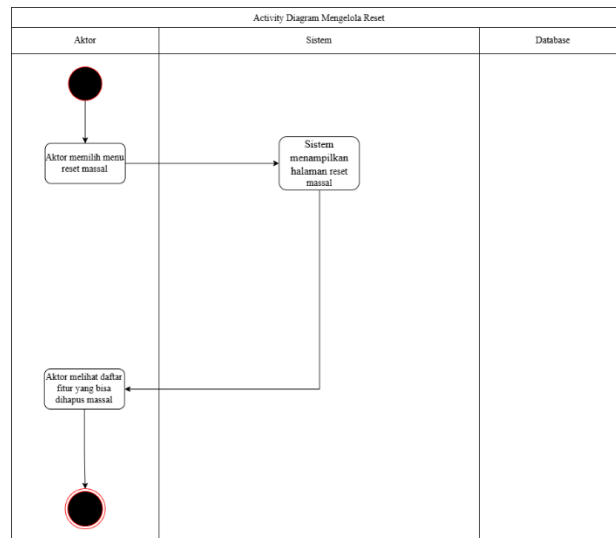


Gambar 3.29 Activity Diagram Log Aktivitas Normal Scenario

10. Activity Diagram Mengelola Reset

- Normal

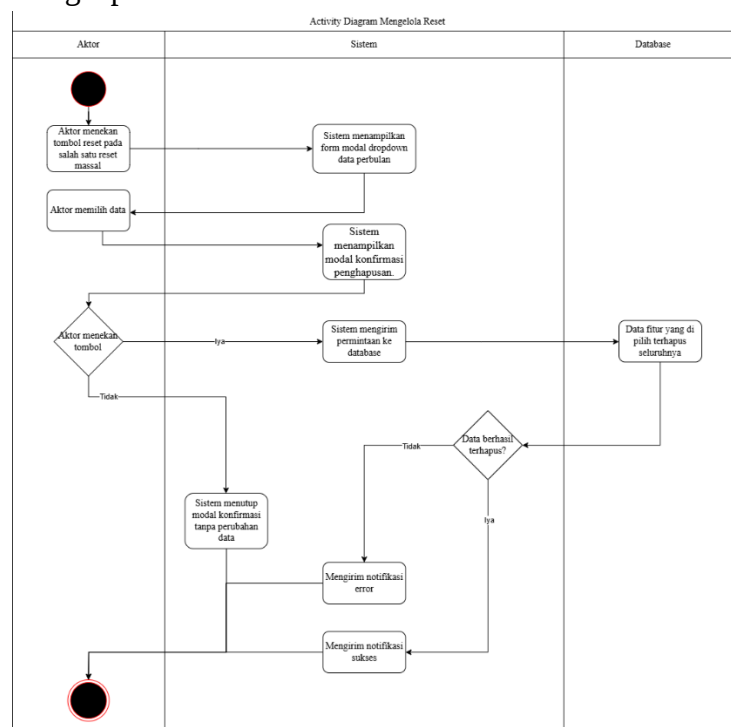
Berikut merupakan Activity diagram normal dari *scenario use case* mengelola reset.



Gambar 3.30 Activity Diagram Mengelola Reset Normal Scenario

- Alternative Scenario 1

Berikut merupakan Activity diagram alternative 1 dari *scenario use case* menghapus massal data.

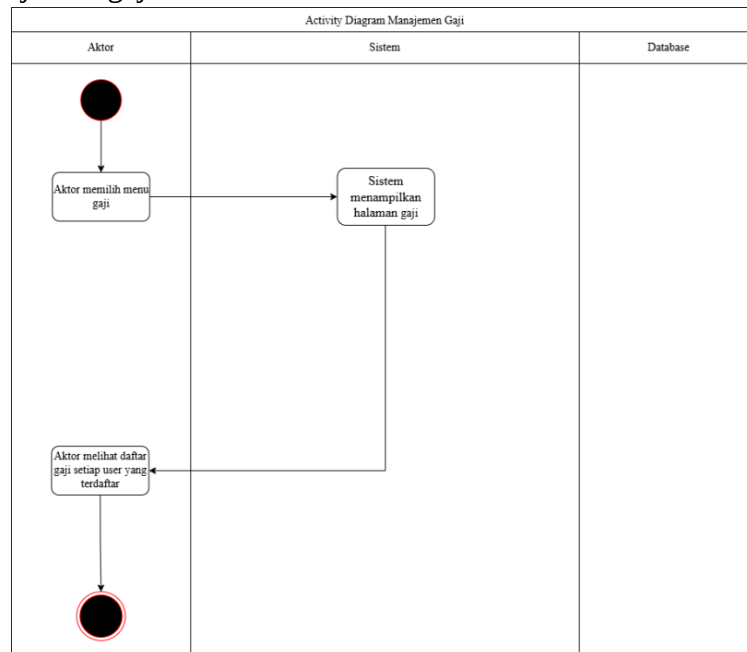


Gambar 3.31 Activity Diagram Mengelola Reset Alternative Scenario 1

11. Activity Diagram Manajemen Gaji

- Normal

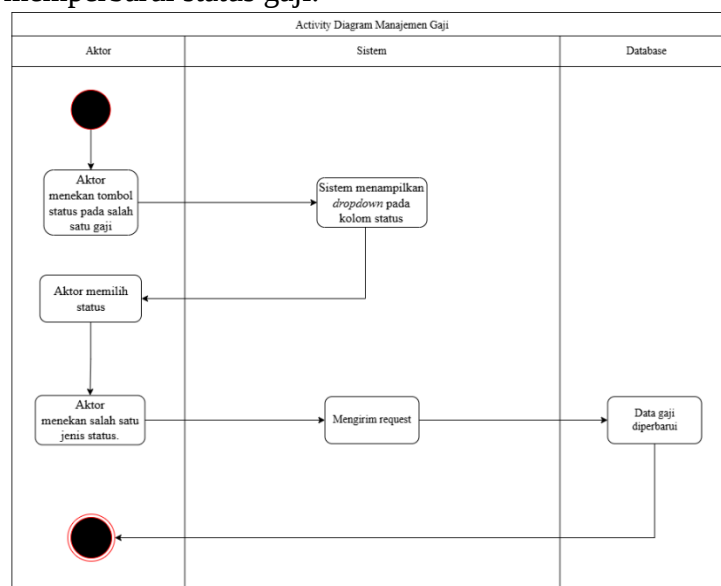
Berikut merupakan *Activity* diagram normal dari *scenario use case* manajemen gaji.



Gambar 3.32 Activity Diagram Manajemen Gaji Normal Scenario

- Alternative Scenario 1

Berikut merupakan *Activity* diagram alternative 1 dari *scenario use case* memperbarui status gaji.

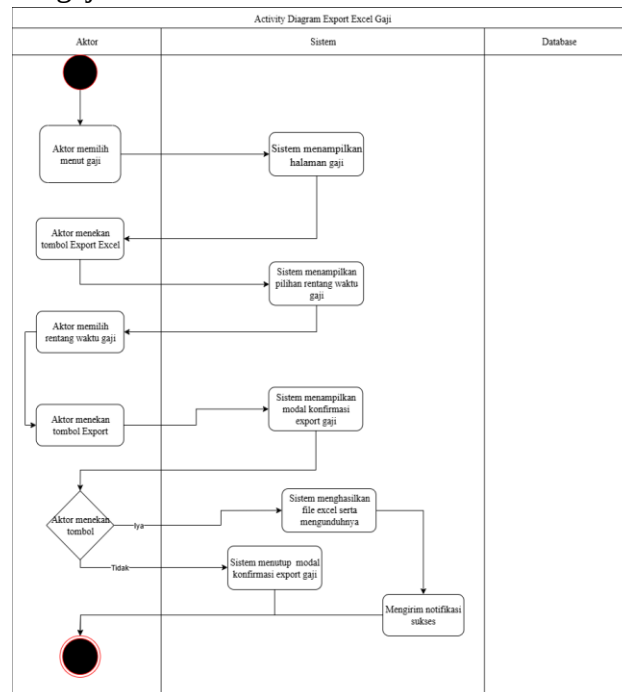


Gambar 3.33 Activity Diagram Manajemen Gaji Alternative Scenario 1

12. Activity Diagram *Export Excel Gaji*

- Normal

Berikut merupakan *Activity* diagram normal dari *scenario use case export excel gaji*.

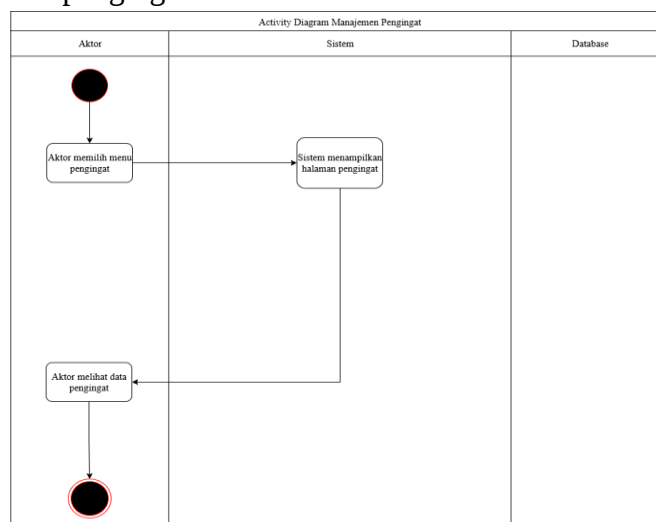


Gambar 3.34 Activity Diagram *Export Excel Gaji Normal Scenario*

13. Activity Diagram *Manajemen Pengingat*

- Normal

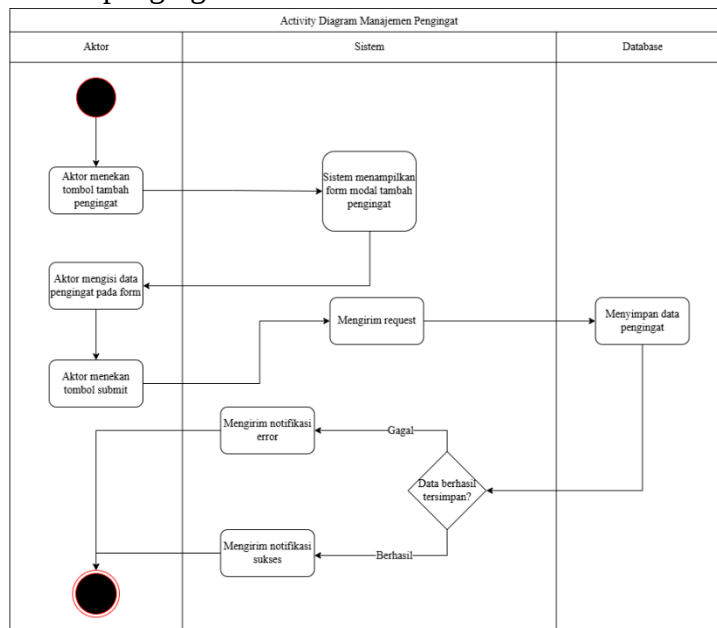
Berikut merupakan *Activity* diagram normal dari *scenario use case manajemen pengingat*.



Gambar 3.35 Activity Diagram *Manajemen Pengingat Normal Scenario*

- *Alternative Scenario 1*

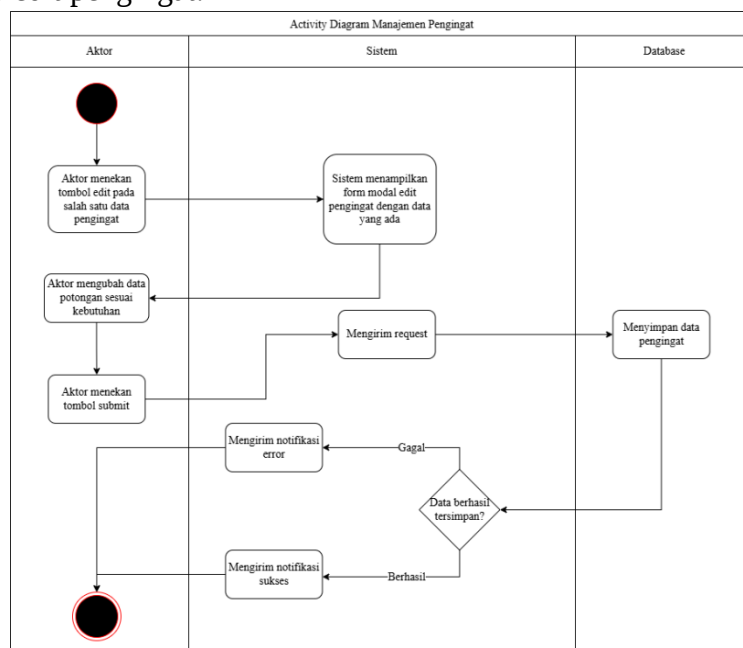
Berikut merupakan *Activity diagram alternative 1* dari *scenario use case tambah pengingat*.



Gambar 3.36 *Activity Diagram Manajemen Peningat Alternative Scenario 1*

- *Alternative Scenario 2*

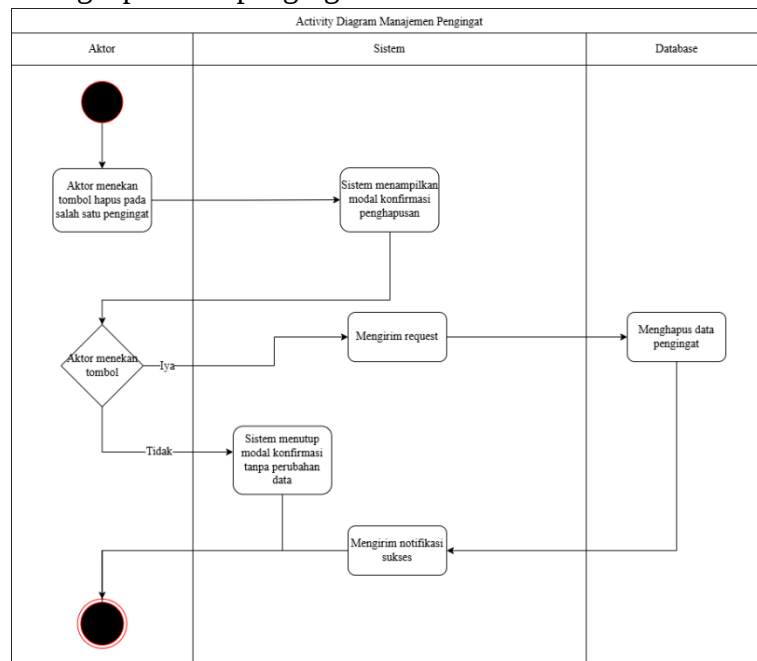
Berikut merupakan *Activity diagram alternative 2* dari *scenario use case edit pengingat*.



Gambar 3.37 *Activity Diagram Manajemen Peningat Alternative Scenario 1*

- *Alternative Scenario 3*

Berikut merupakan *Activity diagram alternative 3* dari *scenario use case* menghapus data pengingat.

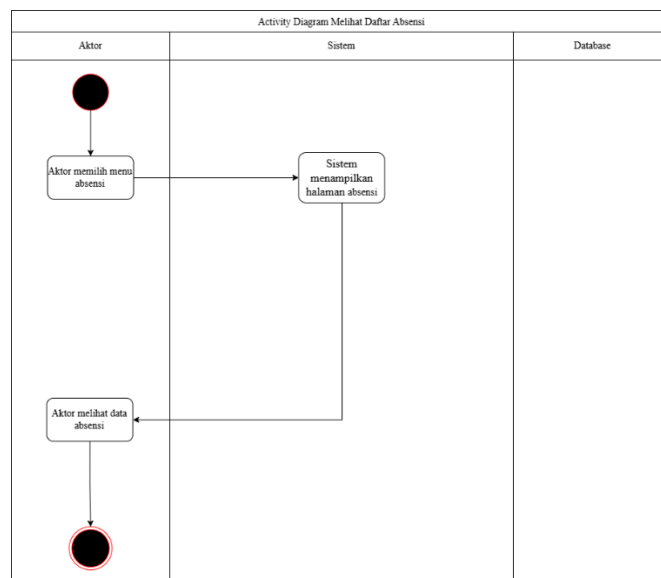


Gambar 3.38 Activity Diagram Manajemen Peningait *Alternative Scenario 3*

14. Activity Diagram Melihat Daftar Absensi

- Normal

Berikut merupakan *Activity diagram normal* dari *scenario use case* melihat daftar absensi.

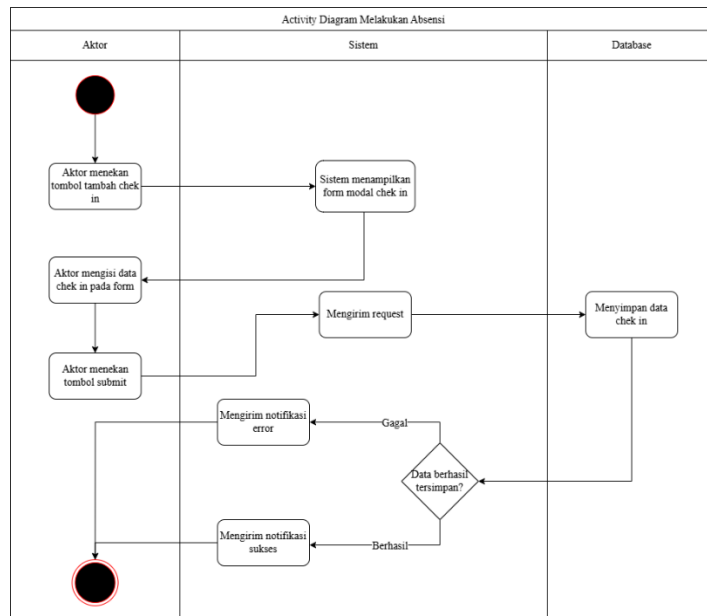


Gambar 3.39 Activity Diagram Melihat Daftar Absensi Normal *Scenario*

15. Activity Diagram Melakukan Absensi

- Normal

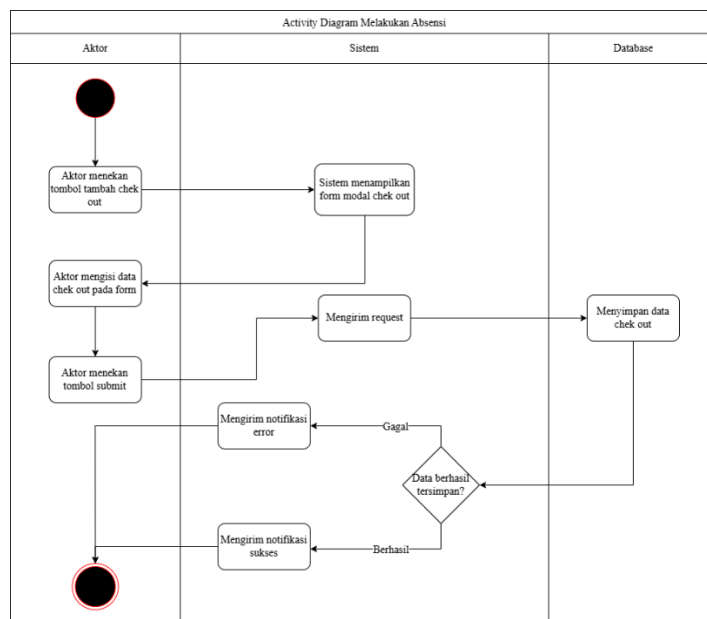
Berikut merupakan Activity diagram normal dari *scenario use case* melakukan *check in*.



Gambar 3.40 Activity Diagram Melakukan Absensi Normal Scenario

- Alternative Scenario 1

Berikut merupakan Activity diagram alternative 1 dari *scenario use case* tambah melakukan *check out*.

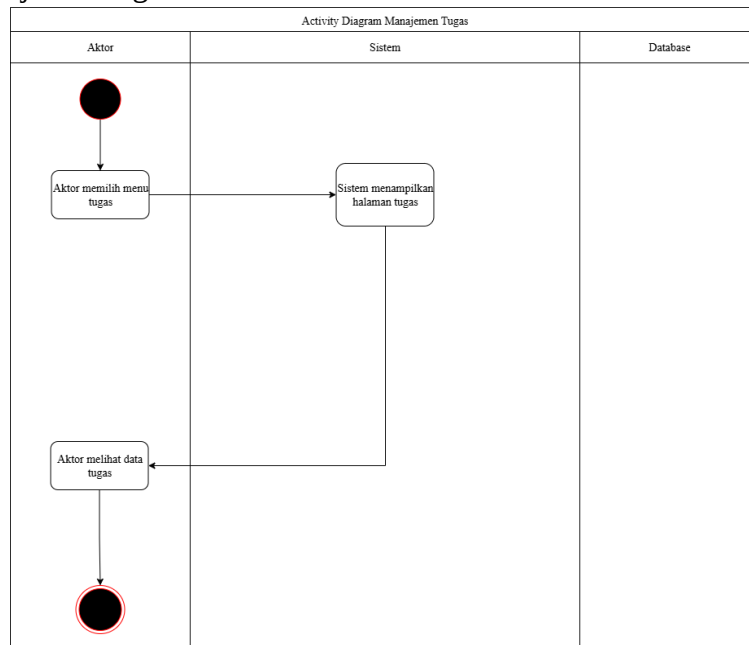


Gambar 3.41 Activity Diagram Melakukan Absensi Alternative Scenario 1

16. Activity Diagram Manajemen Tugas

- Normal

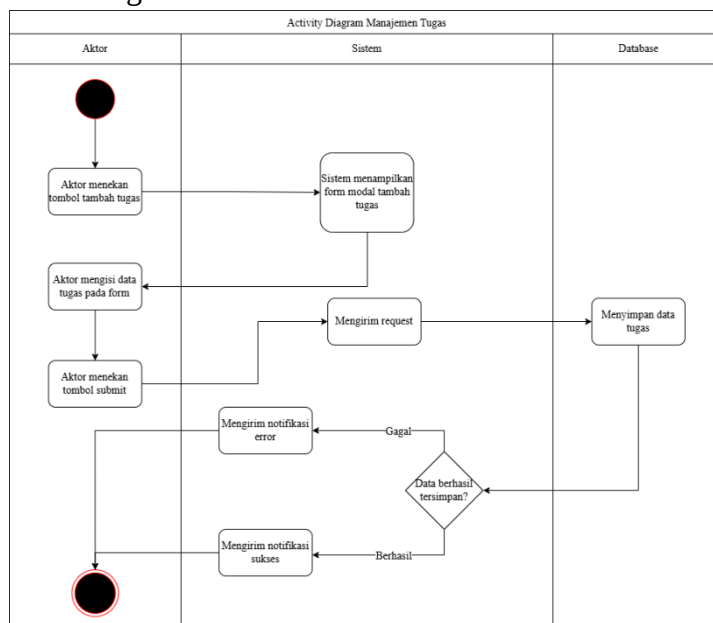
Berikut merupakan Activity diagram normal dari *scenario use case* manajemen tugas.



Gambar 3.42 Activity Diagram Manajemen Tugas Normal Scenario

- Alternative Scenario 1

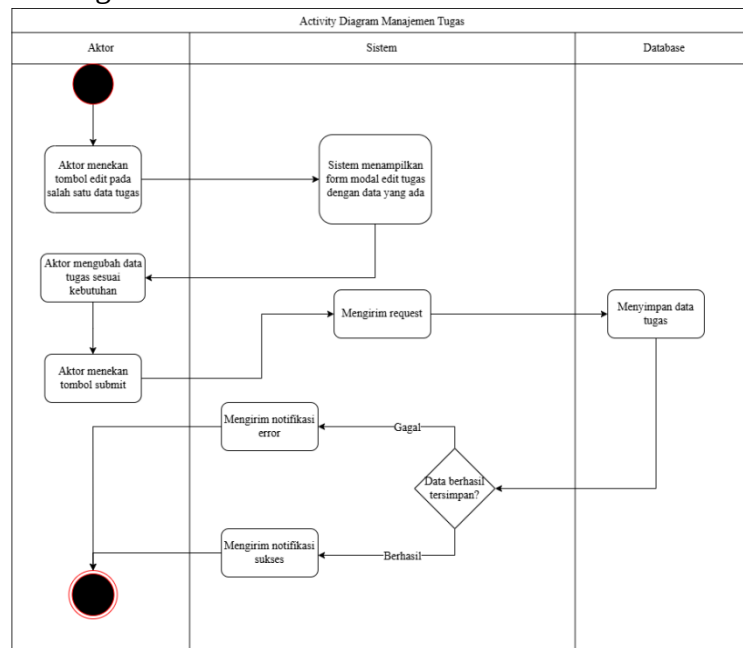
Berikut merupakan Activity diagram alternative 1 dari *scenario use case* tambah tugas.



Gambar 3.43 Activity Diagram Manajemen Tugas Alternative Scenario 1

- *Alternative Scenario 2*

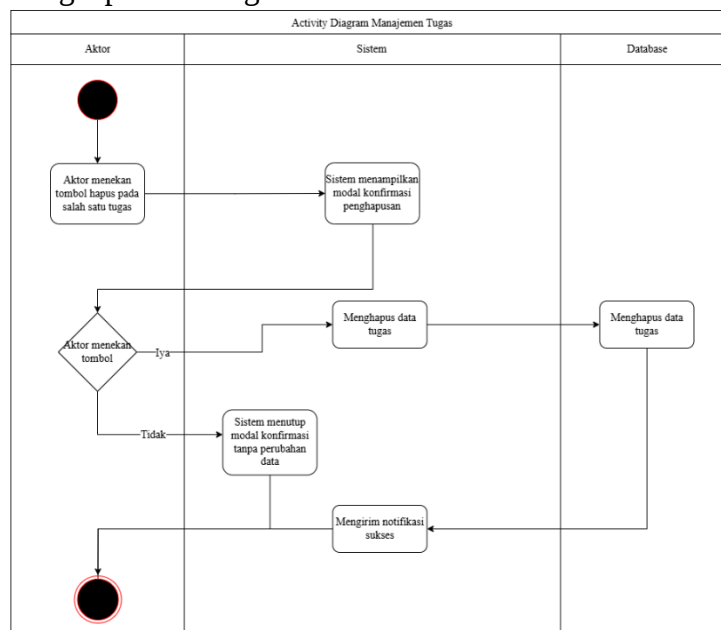
Berikut merupakan *Activity diagram alternative 2* dari *scenario use case edit tugas*.



Gambar 3.44 Activity Diagram Manajemen Tugas *Alternative Scenario 2*

- *Alternative Scenario 3*

Berikut merupakan *Activity diagram alternative 3* dari *scenario use case menghapus data tugas*.

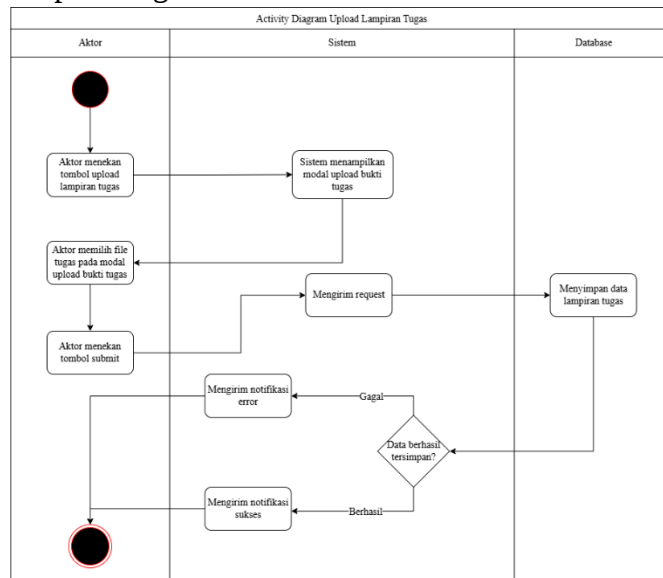


Gambar 3.45 Activity Diagram Manajemen Tugas *Alternative Scenario 3*

17. Activity Diagram Upload Lampiran Tugas

- Normal

Berikut merupakan Activity diagram normal dari *scenario use case upload lampiran tugas*

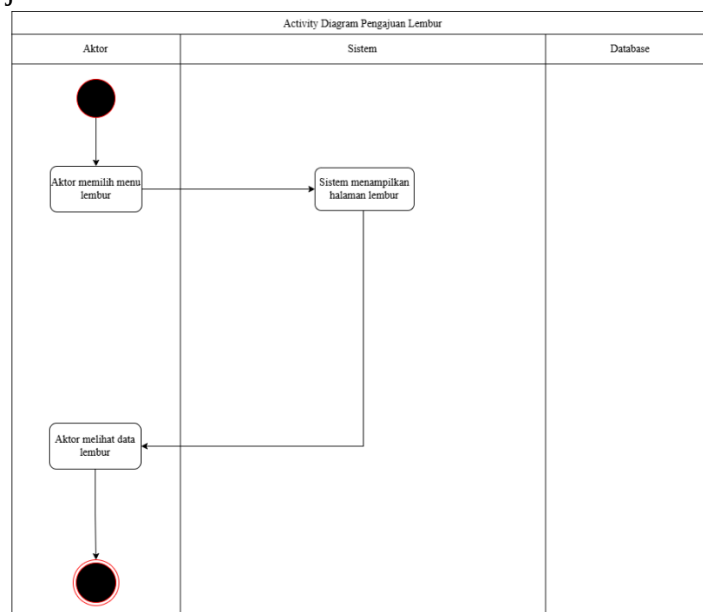


Gambar 3.46 Activity Diagram Upload Lampiran Tugas Normal Scenario

18. Activity Diagram Pengajuan Lembur

- Normal

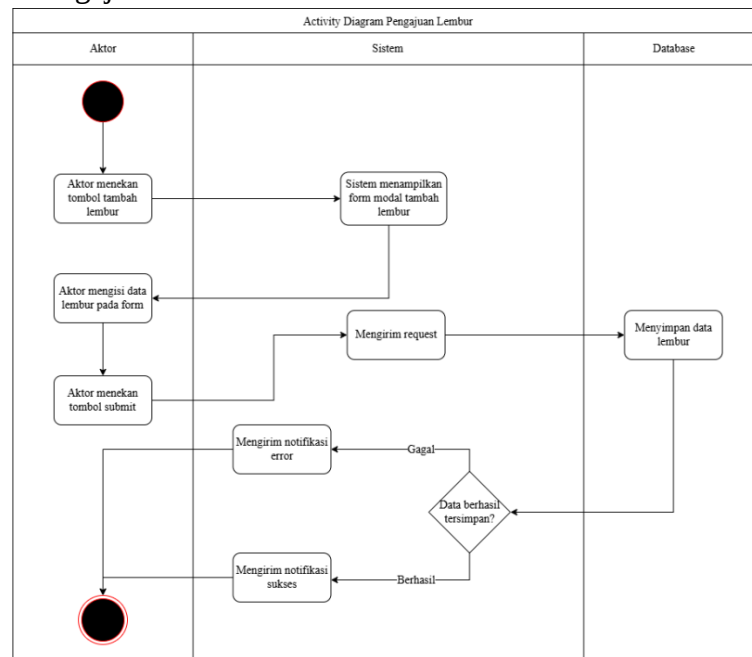
Berikut merupakan Activity diagram normal dari *scenario use case pengajuan lembur*.



Gambar 3.47 Activity Diagram Pengajuan Lembur Normal Scenario

- *Alternative Scenario 1*

Berikut merupakan *Activity diagram alternative 1* dari *scenario use case* mengajukan lembur.

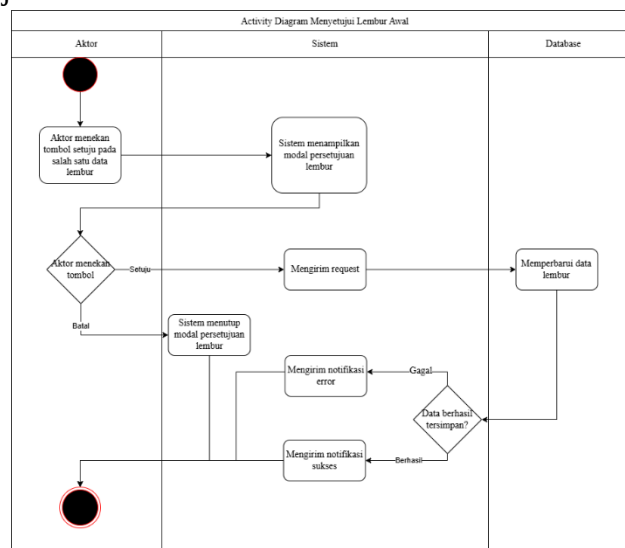


Gambar 3.48 Activity Diagram Pengajuan Lembur *Alternative Scenario 1*

19. Activity Diagram Menyetujui Lembur Awal

- Normal

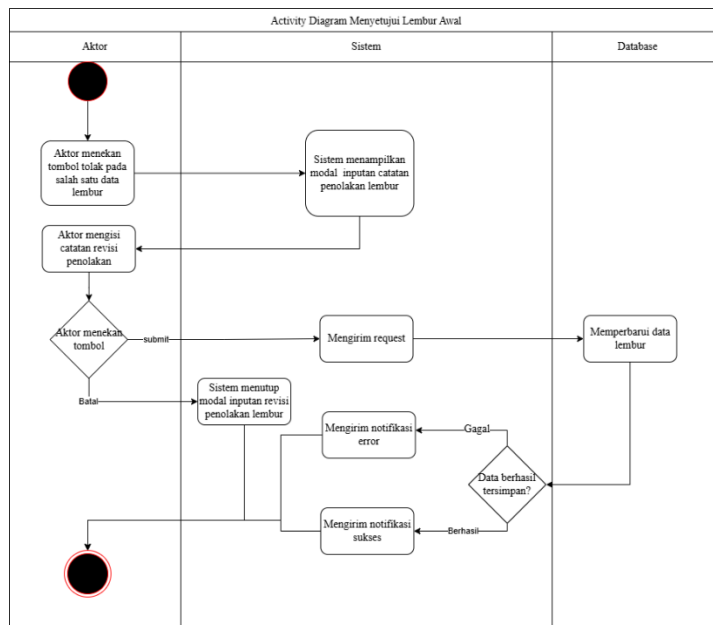
Berikut merupakan *Activity diagram normal* dari *scenario use case* menyetujui lembur awal



Gambar 3.49 Activity Diagram Menyetujui Lembur Awal Normal *Scenario*

- *Alternative Scenario 1*

Berikut merupakan *Activity diagram alternative 1* dari *scenario use case* menolak lembur awal.

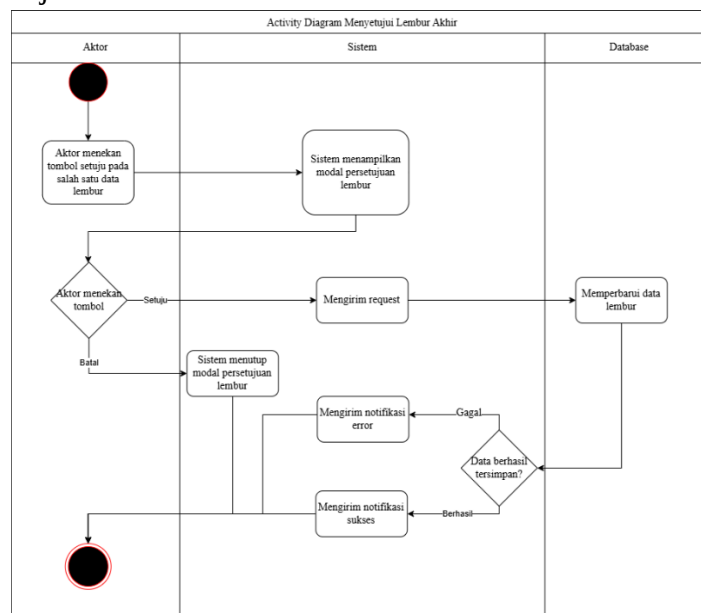


Gambar 3.50 Activity Diagram Menyetujui Lembur Awal
Alternative Scenario 1

20. Activity Diagram Menyetujui Lembur Akhir

- Normal

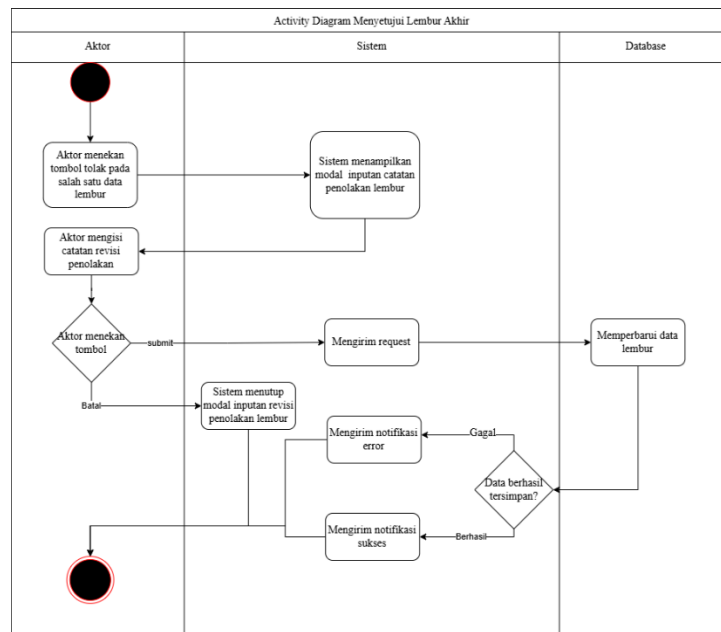
Berikut merupakan *Activity diagram normal* dari *scenario use case* menyetujui lembur akhir.



Gambar 3.51 Activity Diagram Menyetujui Lembur Akhir Normal
Scenario

- *Alternative Scenario 1*

Berikut merupakan *Activity diagram alternative 1* dari *scenario use case* menolak lembur akhir.

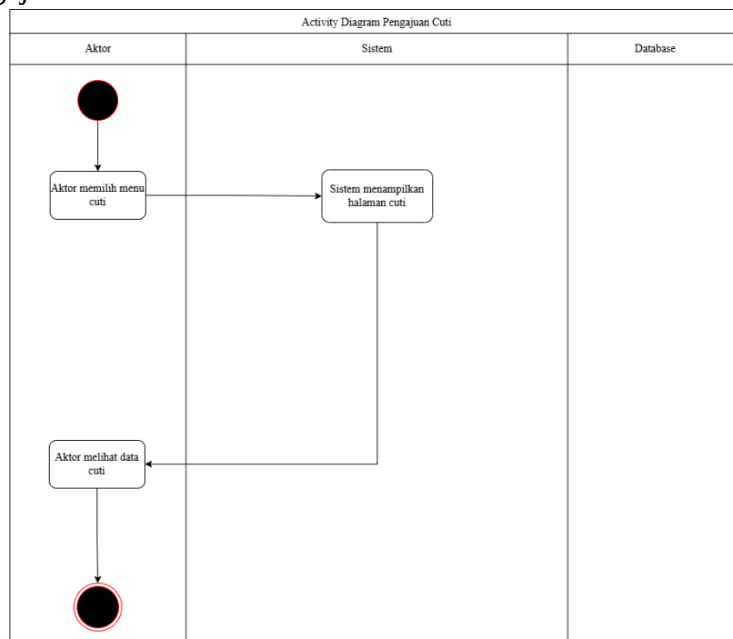


Gambar 3.52 Activity Diagram Menyetujui Lembur Akhir
Alternative Scenario 1

21. Activity Diagram Pengajuan Cuti

- Normal

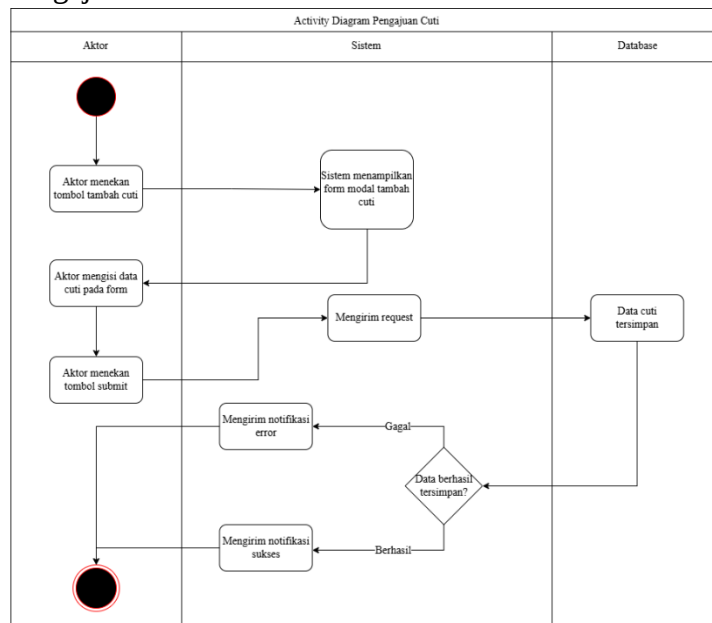
Berikut merupakan *Activity diagram normal* dari *scenario use case* pengajuan cuti.



Gambar 3.53 Activity Diagram Pengajuan Cuti Normal Scenario

- *Alternative Scenario 1*

Berikut merupakan *Activity diagram* alternative 1 dari *scenario use case* mengajukan cuti.

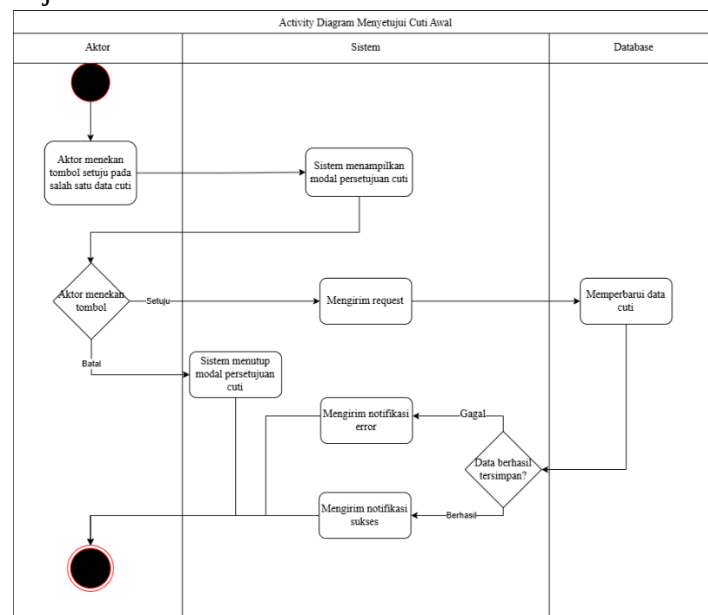


Gambar 3.54 *Activity Diagram* Pengajuan Cuti *Alternative Scenario 1*

22. *Activity Diagram* Menyetujui Cuti Awal

- Normal

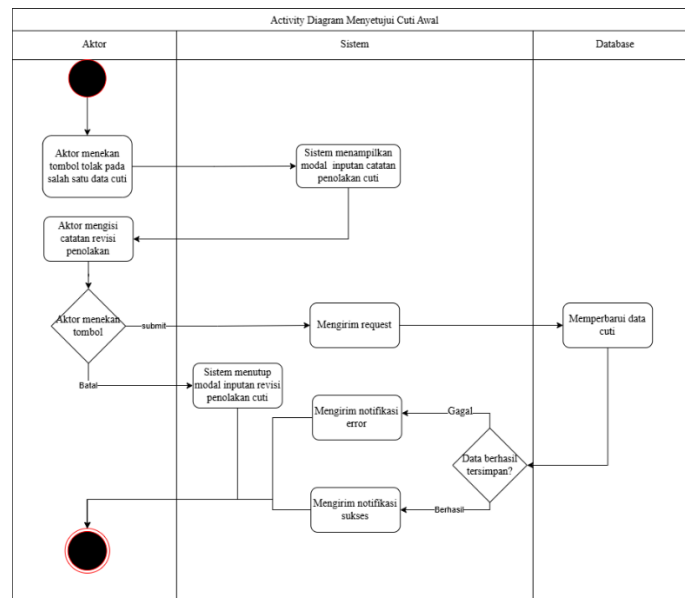
Berikut merupakan *Activity diagram* normal dari *scenario use case* menyetujui cuti awal.



Gambar 3.55 *Activity Diagram* Menyetujui Cuti Awal Normal *Scenario*

- *Alternative Scenario 1*

Berikut merupakan *Activity diagram* alternative 1 dari *scenario use case* menolak cuti awal.

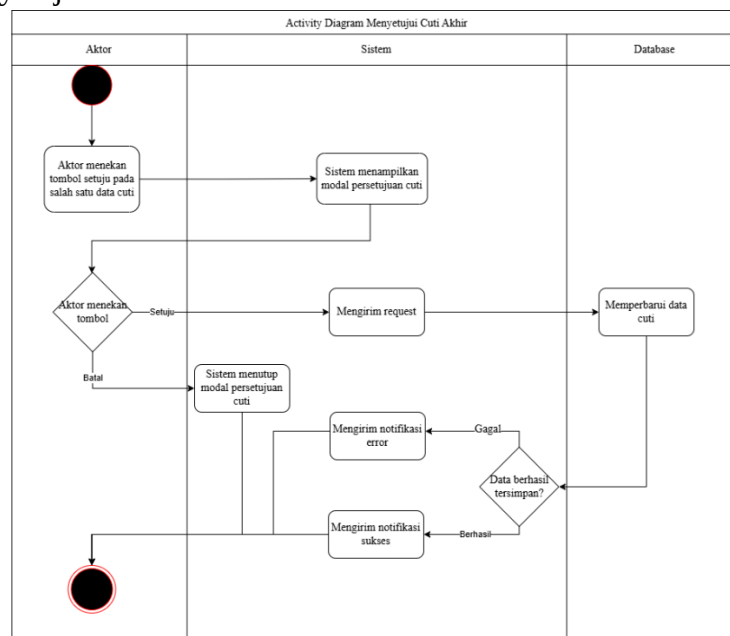


Gambar 3.56 *Activity Diagram Menyetujui Cuti Awal Alternative Scenario 1*

23. *Activity Diagram Menyetujui Cuti Akhir*

- Normal

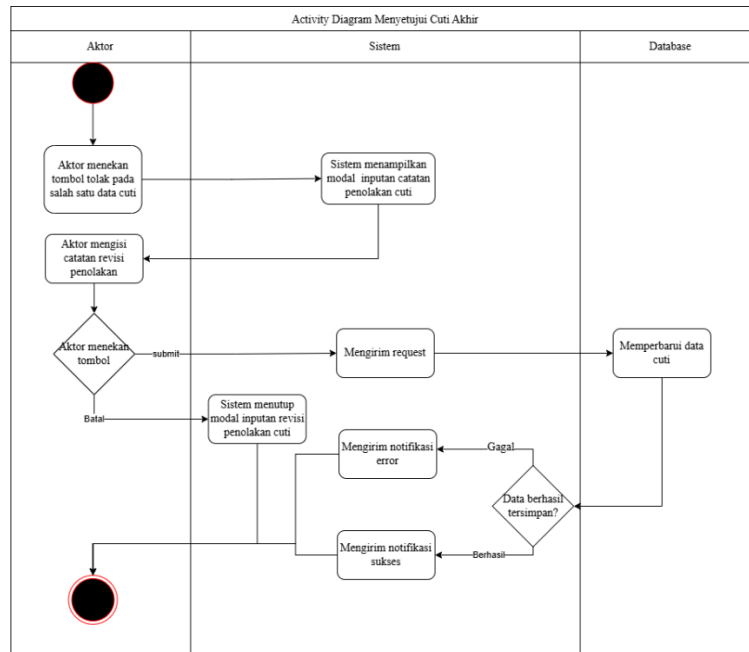
Berikut merupakan *Activity diagram* normal dari *scenario use case* menyetujui cuti akhir



Gambar 3.57 *Activity Diagram Menyetujui Cuti Akhir Normal Scenario*

- *Alternative Scenario 1*

Berikut merupakan *Activity diagram* alternative 1 dari *scenario use case* menolak cuti akhir.

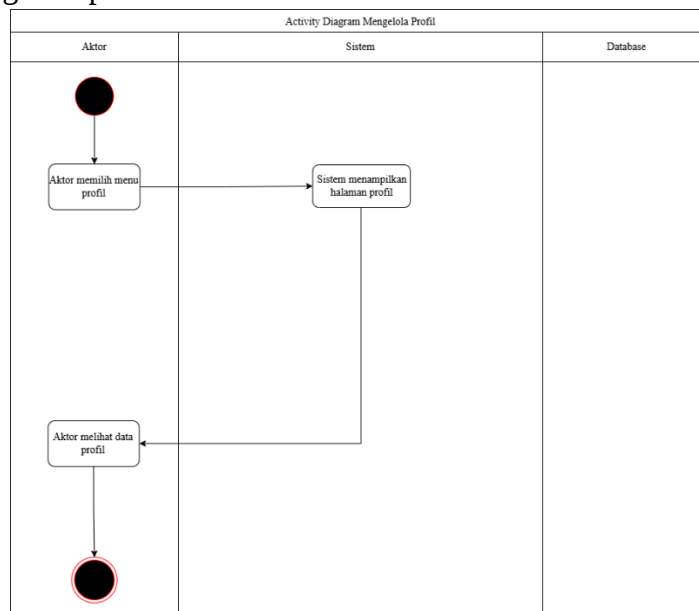


Gambar 3.58 *Activity Diagram Menyetujui Cuti Akhir Alternative Scenario 1*

24. *Activity Diagram Mengelola Profile*

- Normal

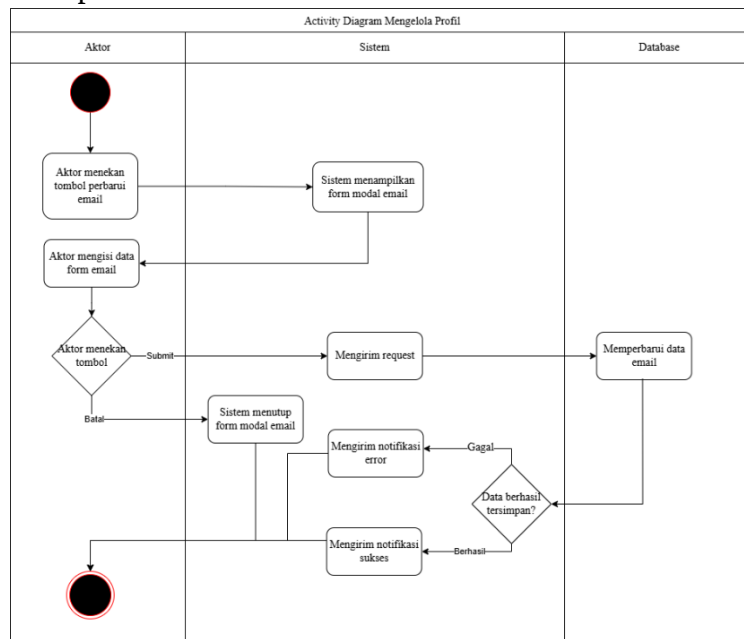
Berikut merupakan *Activity diagram* normal dari *scenario use case* mengelola profil.



Gambar 3.59 *Activity Diagram Mengelola Profile Normal Scenario*

- *Alternative Scenario 1*

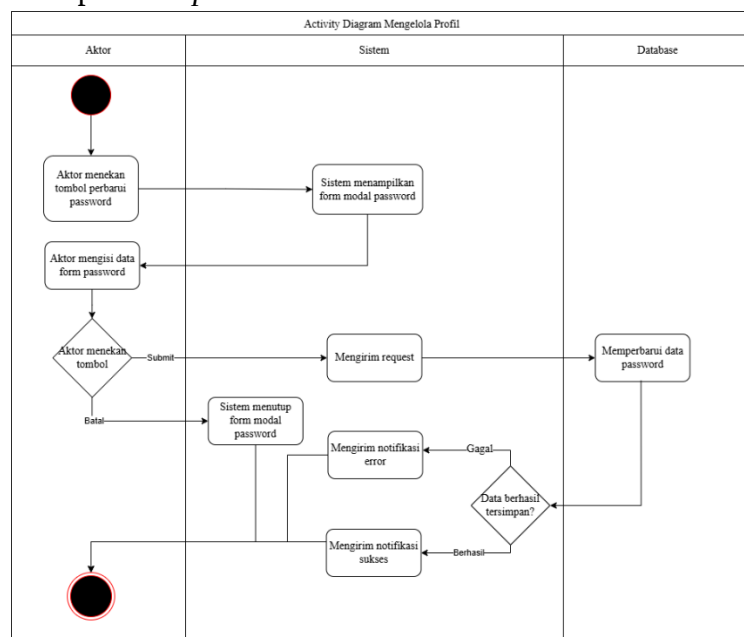
Berikut merupakan Activity diagram alternative 1 dari *scenario use case memperbarui email*.



Gambar 3.60 Activity Diagram Mengelola Profile Alternative Scenario 1

- *Alternative Scenario 2*

Berikut merupakan Activity diagram alternative 2 dari *scenario use case memperbarui password*.

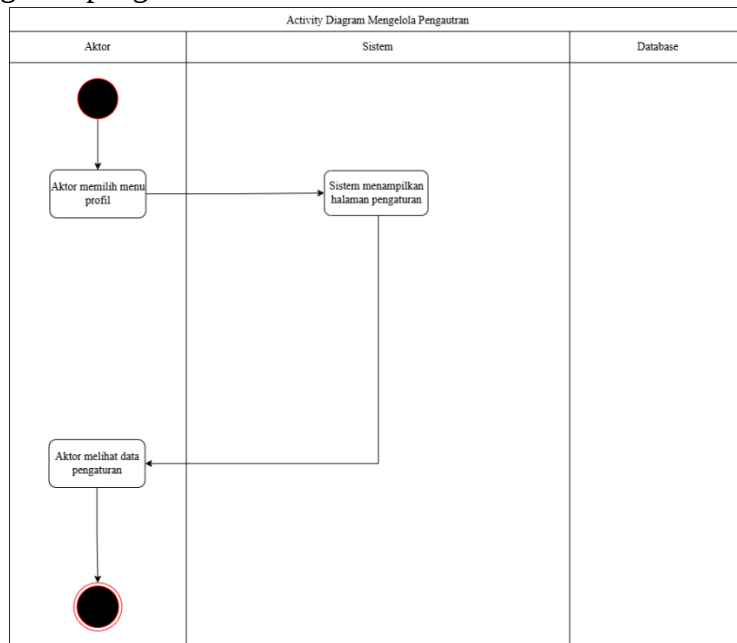


Gambar 3.61 Activity Diagram Mengelola Profile Alternative Scenario 2

25. Activity Diagram Mengelola Pengaturan

- Normal

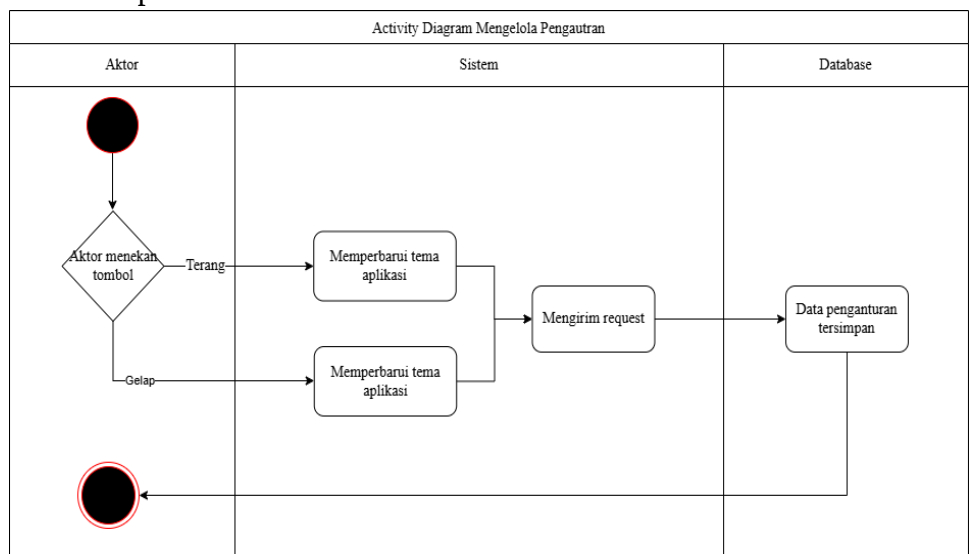
Berikut merupakan Activity diagram normal dari *scenario use case* mengelola pengaturan



Gambar 3.62 Activity Diagram Mengelola Pengaturan Normal
Scenario

- *Alternative Scenario 1*

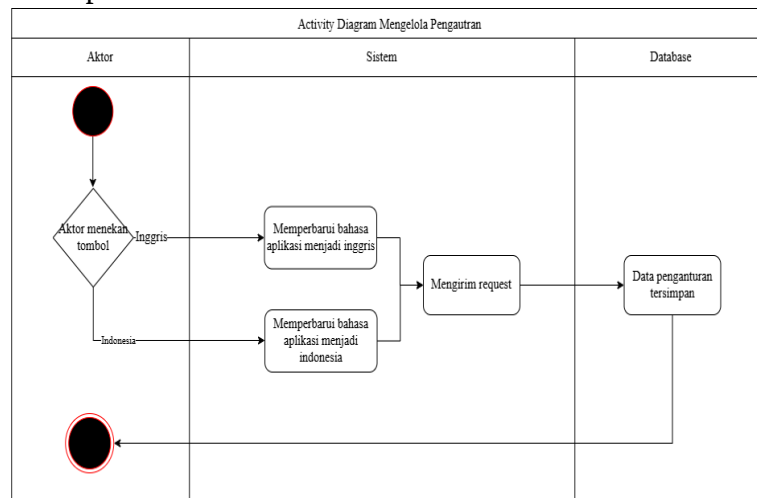
Berikut merupakan Activity diagram alternative 1 dari *scenario use case* memperbarui tema.



Gambar 3.63 Activity Diagram Mengelola Pengaturan *Alternative Scenario 1*

- *Alternative Scenario 2*

Berikut merupakan *Activity diagram alternative 2* dari *scenario use case* memperbarui bahasa.

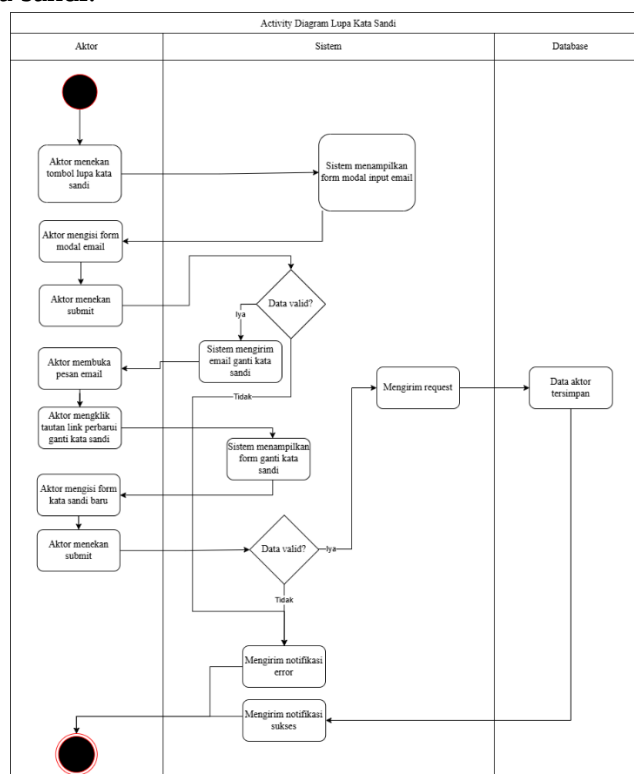


Gambar 3.64 Activity Diagram Mengelola Pengaturan *Alternative Scenario 2*

26. Activity Diagram Lupa Kata Sandi

- Normal

Berikut merupakan *Activity diagram normal* dari *scenario use case* lupa kata sandi.

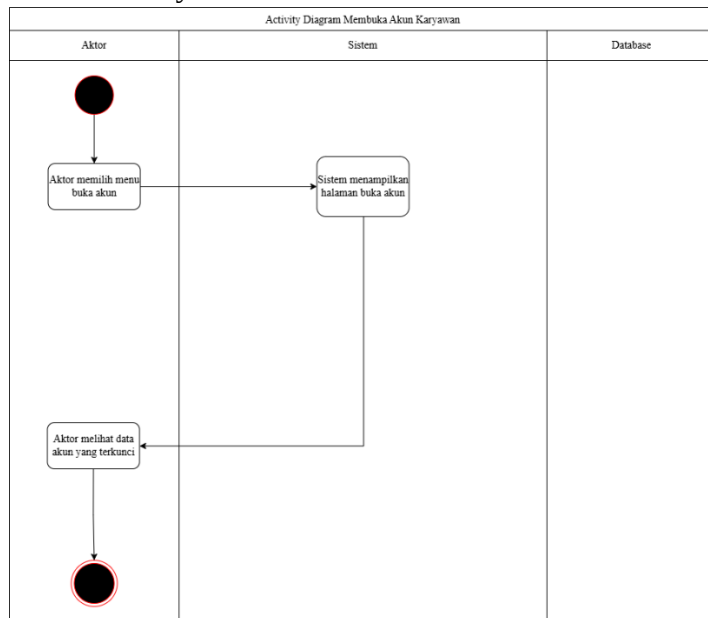


Gambar 3.65 Activity Diagram Lupa Kata Sandi Normal *Scenario*

27. Activity Diagram Membuka Akun Karyawan

- Normal

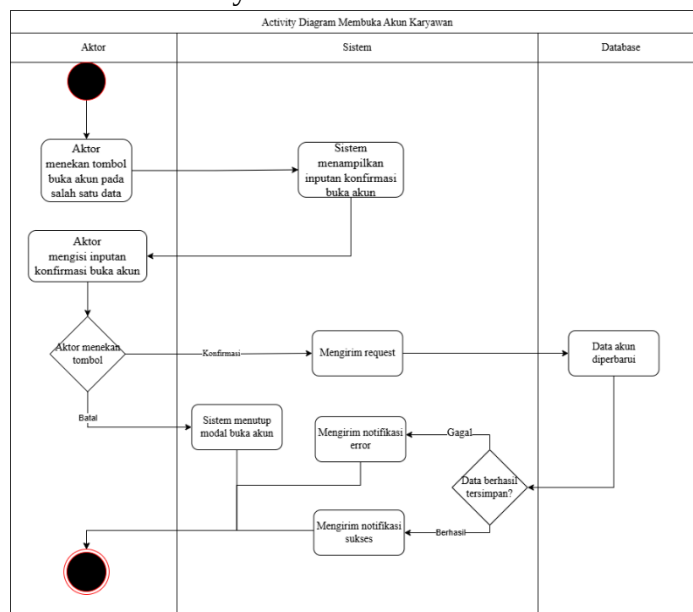
Berikut merupakan *Activity diagram normal* dari *scenario use case* membuka akun karyawan.



Gambar 3.66 Activity Diagram Membuka Akun Karyawan Normal Scenario

- Alternative Scenario 1

Berikut merupakan *Activity diagram alternative 1* dari *scenario use case* membuka akun karyawan.

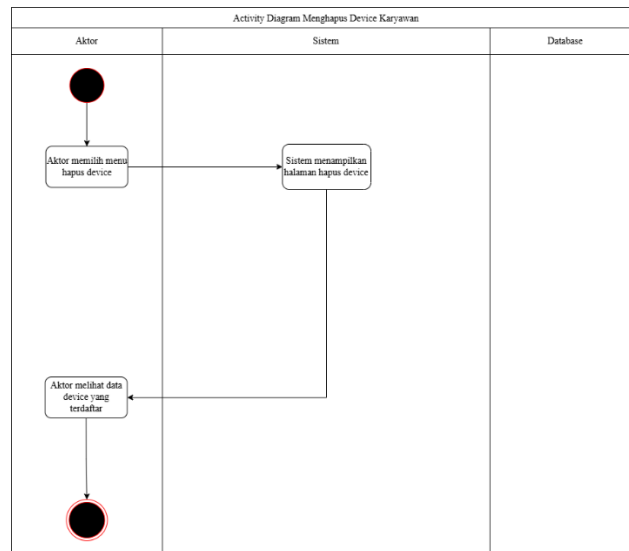


Gambar 3.67 Activity Diagram Membuka Akun Karyawan Alternative Scenario 1

28. Activity Diagram Menghapus device Karyawan

- Normal

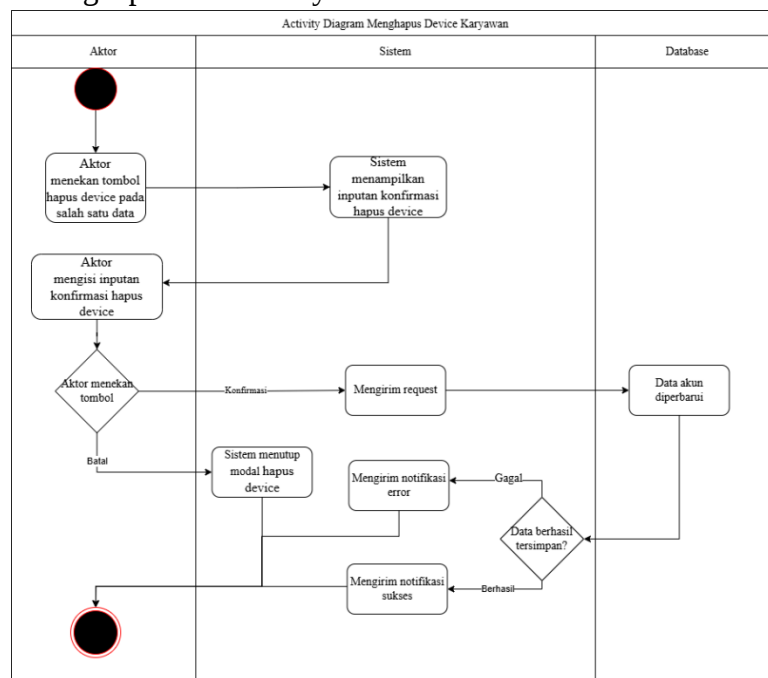
Berikut merupakan Activity diagram normal dari *scenario use case* lupa kata sandi.



Gambar 3.68 Activity Diagram Menghapus Device Karyawan Normal Scenario

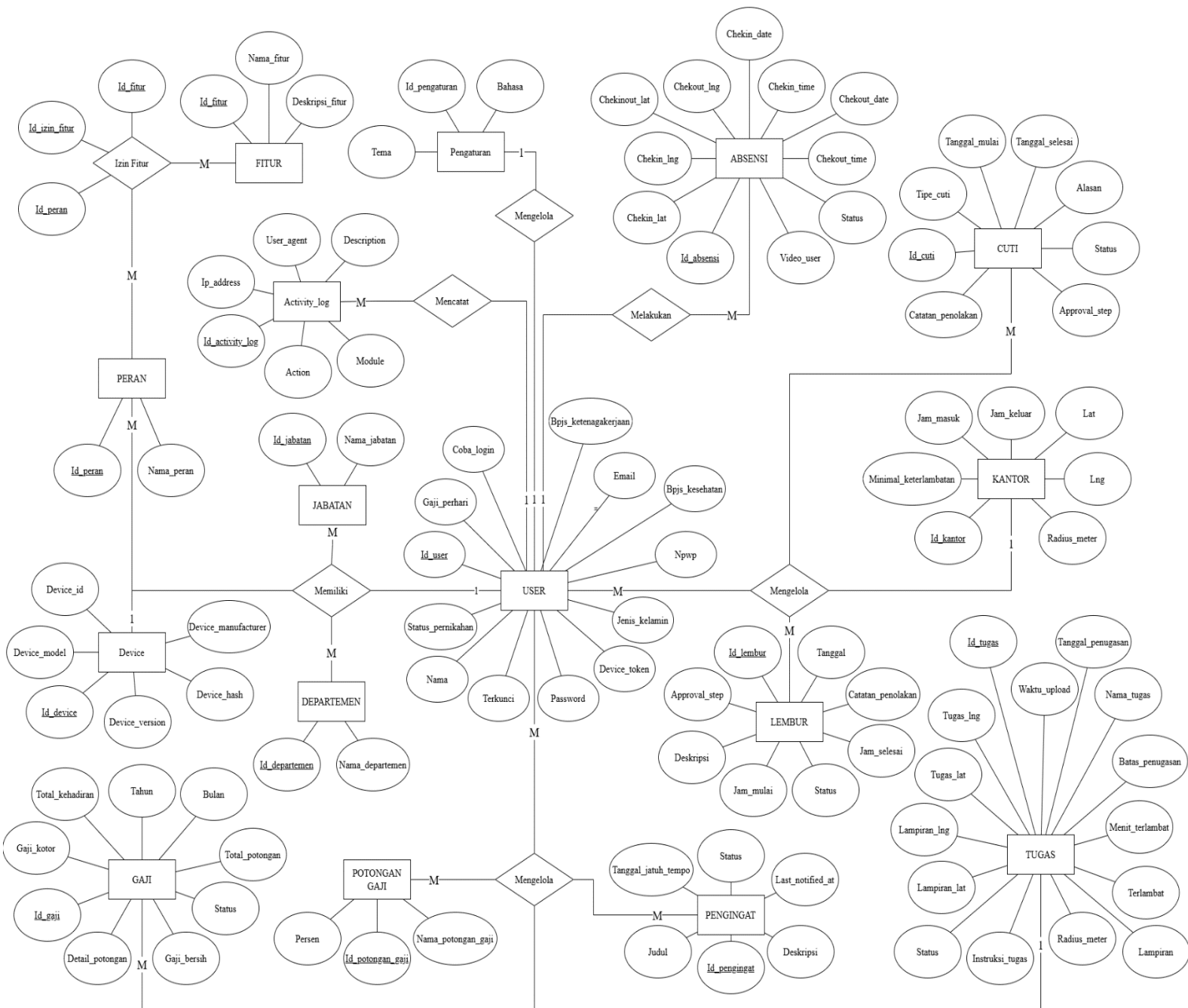
- Alternative Scenario 1

Berikut merupakan Activity diagram alternative 1 dari *scenario use case* menghapus device karyawan.



Gambar 3.69 Activity Diagram Menghapus Device Karyawan Alternative Scenario 1

3.2.6 ER Diagram



Gambar 3.70 ER Diagram

Berikut merupakan penjelasan singkat dari ER Diagram di atas:

1. *User*

Entitas *user* menyimpan seluruh informasi utama mengenai pengguna aplikasi, seperti nama, *email*, *password*, bpjs kesehatan, bpjs

ketenagakerjaan, npwp, status pernikahan, jenis kelamin, gaji per hari, token perangkat serta data pendukung lain untuk kebutuhan fungsional aplikasi. *User* menjadi pusat dari hampir seluruh proses karena *user* dapat melakukan absensi, mengajukan cuti, mengelola lembur, menerima tugas, mendapatkan pengingat, dan menghasilkan data gaji. Setiap *user* juga tercatat aktivitasnya melalui *Activity log* dan terhubung dengan satu perangkat yang digunakan untuk keamanan.

2. Departemen

Entitas departemen berfungsi untuk mengelompokkan *user* berdasarkan unit kerja atau divisi mereka dalam organisasi. Setiap *user* berada pada satu departemen tertentu, sehingga departemen membantu mengatur struktur organisasi dan pemetaan karyawan dalam perusahaan.

3. Jabatan

Entitas jabatan menyimpan data mengenai posisi kerja *user* dalam perusahaan. Setiap *user* memiliki satu jabatan, yang akan berpengaruh pada peran, struktur kepemimpinan, serta hubungan kerja di dalam sistem. Jabatan juga menjadi referensi dalam pengembangan fitur atau kebijakan tertentu.

4. Peran

Entitas peran digunakan untuk mengatur hak akses setiap *user* dalam aplikasi. *User* hanya memiliki satu peran, dan setiap peran dapat memiliki akses ke beberapa fitur aplikasi. Sistem *role-based access control* ini memastikan bahwa *user* hanya bisa mengakses fungsi yang sesuai dengan perannya dan jika di masa mendatang manajer ingin memperbarui suatu peran tidak perlu membuka kodingan.

5. Fitur

Entitas fitur berisi daftar fitur atau modul aplikasi yang dapat digunakan dalam sistem. Setiap fitur memiliki nama dan deskripsi, dan fitur ini dihubungkan dengan peran untuk menentukan *user* mana yang memiliki akses terhadap fitur tertentu.

6. Absensi

Entitas absensi mencatat kehadiran harian *user* dengan *detail* seperti koordinat *check-in/check-out*, waktu masuk dan keluar, tanggal, dan status kehadiran. Setiap *user* dapat memiliki banyak catatan absensi, dan sistem absensi ini dipengaruhi oleh pengaturan kantor seperti radius lokasi dan jam kerja.

7. Cuti

Entitas cuti mengelola pengajuan cuti *user* yang berisi jenis cuti, alasan, tanggal mulai dan selesai, status, serta tingkat persetujuan. Setiap *user* dapat mengajukan banyak cuti, dan setiap permohonan dapat melalui proses persetujuan bertahap atau ditolak dengan catatan penjelasan.

8. Lembur

Entitas lembur menyimpan data pengajuan lembur *user*, termasuk tanggal, jam mulai, jam selesai, deskripsi lembur, status persetujuan, dan catatan penolakan. Seorang *user* dapat memiliki banyak lembur, dan setiap lembur melalui proses persetujuan bertahap atau ditolak dengan catatan penjelasan.

9. Tugas

Entitas tugas mengelola penugasan yang diberikan kepada *user*, seperti nama tugas, instruksi, lokasi tugas, batas waktu, status penyelesaian, dan lampiran bukti. Setiap *user* dapat menerima banyak tugas, dan sistem tugas dilengkapi koordinat lokasi dan waktu upload untuk memastikan keaslian pengerjaan.

10. Kantor

Entitas kantor berisi pengaturan operasional kantor, seperti jam masuk, jam pulang, radius absensi, lokasi (lat/lng), dan batas minimal keterlambatan.

11. Peningat

Entitas pengingat menyediakan fungsi pengingat bagi *user* terkait aktivitas atau tenggat waktu tertentu. Setiap pengingat memiliki judul,

deskripsi, dan waktu pemberitahuan, dan satu *user* dapat memiliki banyak pengingat.

12. *Device*

Entitas *device* mencatat informasi perangkat yang digunakan oleh *user* untuk *login*. Informasi yang disimpan meliputi ID perangkat, model, versi, dan pabrikan. Setiap *user* terhubung dengan satu perangkat sebagai langkah menyesuaikan kebutuhan *client* yang ingin satu *user* hanya bisa *login* satu *device*.

13. *Activity Log*

Entitas *Activity Log* menyimpan catatan aktivitas *user* dalam aplikasi, termasuk modul yang diakses, tindakan yang dilakukan, IP address, serta *user-agent*. Catatan ini berguna untuk audit, keamanan, dan pelacakan aktivitas pengguna.

14. Gaji

Entitas gaji menyimpan data perhitungan gaji bulanan *user*, seperti gaji kotor, total potongan, gaji bersih, status gaji, dan periode bulan/tahun. Setiap *user* dapat memiliki banyak catatan gaji untuk setiap periode penggajian.

15. Potongan Gaji

Entitas potongan gaji mendefinisikan berbagai jenis potongan yang dapat muncul dalam perhitungan gaji, misalnya potongan keterlambatan atau absensi. Potongan ini terkait dengan entitas gaji dan memiliki *Persentase* atau nominal sesuai aturan perusahaan.

16. Pengaturan

Entitas pengaturan menyimpan data pengaturan *user*, seperti tema dan bahasa. Setiap *user* hanya dapat mengelola data pengaturannya sendiri.

17. Izin Fitur

Relasi izin fitur merupakan relasi *many-to-many* dari entitas peran dan fitur yang pada saat implementasi akan menjadi tabel pivot baru yang mencatat setiap fitur dari tabel peran.

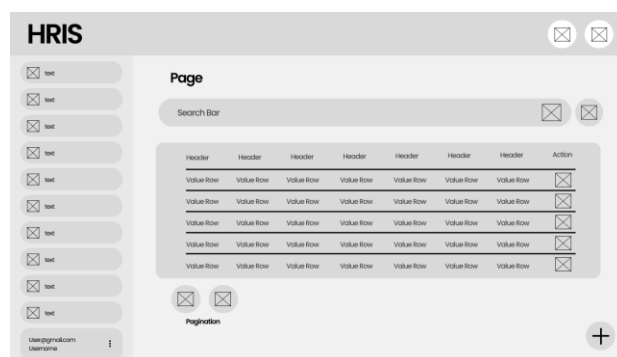
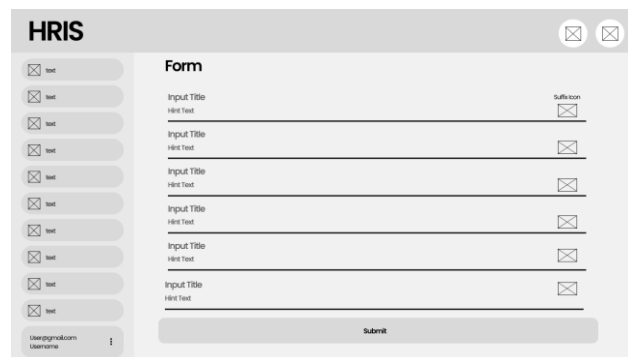
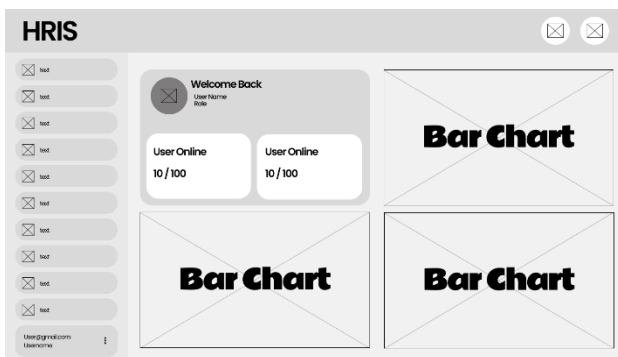
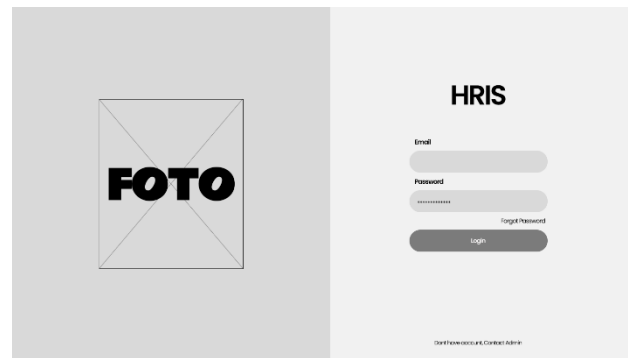
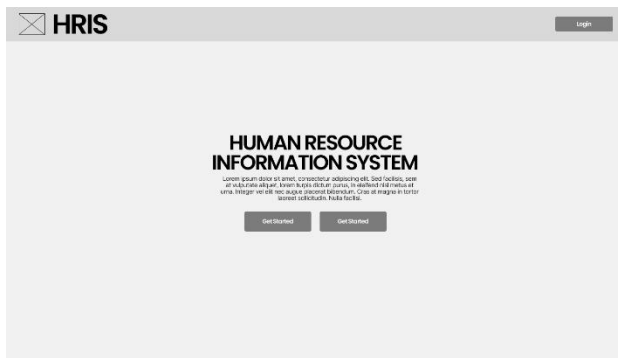
3.2.7 Perancangan Antarmuka (Wireframe)

1. Mobile



Gambar 3.71 Wireframe Mobile

2. Web



Gambar 3.72 Wireframe Web

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Hasil Implementasi

Proses implementasi pada proyek akhir ini dilakukan setelah tahap analisis dan perancangan sistem selesai dilaksanakan. Implementasi dilakukan dengan menyiapkan *framework* serta lingkungan pengembangan yang diperlukan, baik untuk sisi *backend* maupun *frontend*.

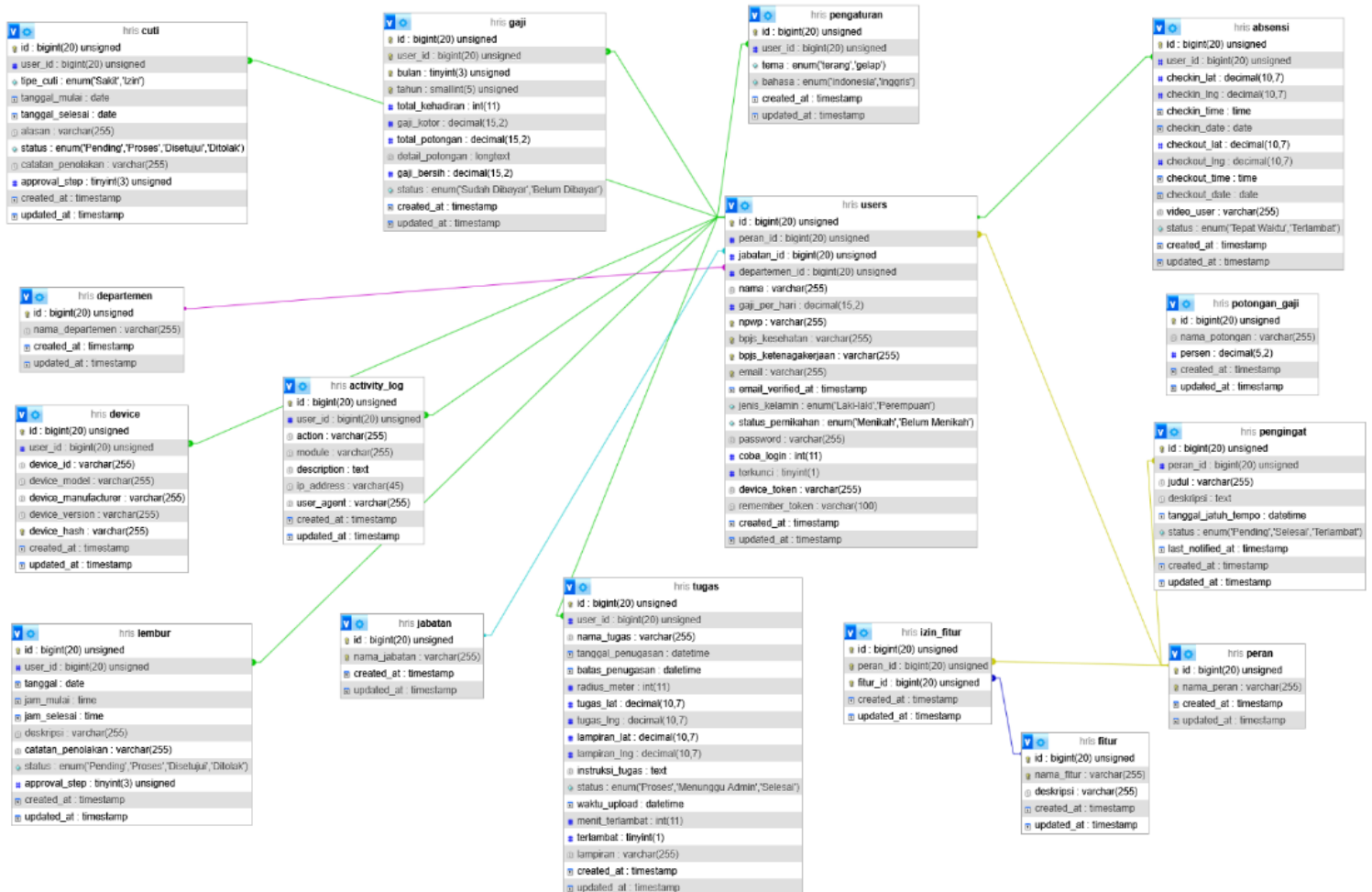
Pada sisi *backend*, *Laravel* digunakan untuk membangun API yang menjadi penghubung antara aplikasi dan *database*. Tahap awal dilakukan dengan membuat *migration* untuk mendefinisikan struktur tabel pada *database*. Dengan adanya *migration*, proses perubahan struktur tabel dapat dilakukan secara mudah dan konsisten. *Database* yang digunakan adalah MySQL, yang dijalankan menggunakan XAMPP sebagai *server* lokal. Pengujian API dilakukan menggunakan *postman* untuk memastikan setiap *route* dan *endpoint* sudah memberikan respons sesuai dengan kebutuhan fungsional. Setelah hasil *testing* pada *postman* sesuai, tahap berikutnya adalah mengimplementasikan API tersebut ke dalam aplikasi *Flutter*.

Pada sisi *frontend* *Flutter*, proses implementasi dilakukan dengan menyesuaikan *model* dan *service* berdasarkan hasil *response* API dari *laravel*. *Model* digunakan untuk memetakan struktur data dari API, sedangkan *service* digunakan untuk melakukan komunikasi dengan sisi *Laravel*.

Pada proyek ini, dilakukan *remote* kepada *server* lokal kantor dan melakukan *setup* pada *server*, baik instalasi komponen MySQL sebagai *database* maupun Apache sebagai *web server*. Pada saat *setup* port, IP privat *server* dikonfigurasi dengan port 444 sebagai akses *frontend*, port 80 untuk HTTP *backend*, dan port 8080 untuk HTTPS *backend*. Pada sisi *backend*, sengaja dibuka dua port (HTTP dan HTTPS) untuk mengatasi konflik *mixed content* pada saat akses *web* dan aplikasi. Untuk *web*, URL API *backend* dibuat dengan port 8080 HTTPS agar sesuai dengan URL 444 *web*, sedangkan untuk aplikasi dihubungkan dengan URL

API port 80 HTTP. Sistem operasi yang digunakan adalah Lubuntu dan proses *remote* dilakukan menggunakan SSH

4.1.1 Implementasi Basis data



Gambar 4.1 Implementasi Basis Data

4.1.2 Implementasi API

Setiap respons API menggunakan format JSON yang memuat *message*, token, dan data agar mudah dibaca serta diintegrasikan dengan *framework* Flutter. pengembangan backend, saya juga menggunakan berbagai HTTP *method* seperti *GET*, *POST*, *PUT*, *PATCH*, dan *DELETE* sesuai dengan fungsi masing-masing endpoint. Setiap method digunakan untuk operasi yang berbeda, misalnya *POST* untuk proses *login* dan *input* data baru, *GET* untuk pengambilan data, *PUT/PATCH* untuk proses pembaruan data, serta *DELETE* untuk penghapusan data. Berikut

merupakan *response* API dari serta endpoint yang digunakan untuk memenuhi kebutuhan fungsional sistem:

Tabel 4.1 Dokumentasi Implementasi API

No	Endpoint	Methods	Screen Shoot Respon
1	/api/login	POST	<pre> { "message": "Login berhasil", "token": "1 nKvtDw5tVPZWAIT5SQP187uVY8j5NYCswvKc8nn02978c64", "data": { "id": 1, "peran_id": 1, "jabatan_id": 3, "departemen_id": 1, "nama": "User Super", "gaji_per_hari": 500000, "npwp": null, "bpjs_kesehatan": null, "bpjs_ketenagakerjaan": null, "email": "usersuper@gmail.com", "email_verified_at": null, "jenis_kelamin": "Laki-laki", "status_pernikahan": "Menikah", "coba_login": 0, "terkunci": 0, "device_token": null, "created_at": "2025-11-19T07:13:23.000000Z", "updated_at": "2025-11-19T07:13:23.000000Z", "peran": { "id": 1, "nama_peran": "Admin Super", "created_at": "2025-11-19T07:13:22.000000Z", "updated_at": "2025-11-19T07:13:22.000000Z", "fitur": [</pre>
2	/api/email	PUT	<pre> { "message": "Email berhasil diperbarui", "data": { "email": "grey@gmail.com" } } </pre>
3	/api/logout	POST	<pre> { "message": "Logout berhasil" } </pre>
4	/api/me	GET	<pre> { "message": "User ditemukan", "data": { "id": 1, "peran_id": 1, "jabatan_id": 3, "departemen_id": 1, "nama": "User Super", "gaji_per_hari": "500000.00", "npwp": null, "bpjs_kesehatan": null, "bpjs_ketenagakerjaan": null, "email": "usersuper@gmail.com", "email_verified_at": null, "jenis_kelamin": "Laki-laki", "status_pernikahan": "Menikah", "coba_login": 0, "terkunci": 0, </pre>
5	/api/change-password	POST	<pre> { "message": "Password berhasil diperbarui" } </pre>

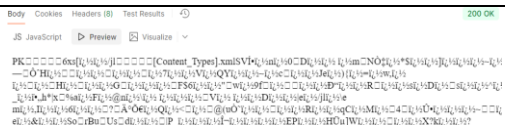
6	/api/lembur	GET	<pre> { "message": "Data lembur berhasil diambil", "data": [{ "id": 1, "user_id": 4, "tanggal": "2025-08-07", "jam_mulai": "18:00:00", "jam_selesai": "21:00:00", "deskripsi": "Lembur menyelesaikan laporan akhir bulan", "catatan_penolakan": null, "status": "Pending", "approval_step": 0, "created_at": "2025-11-19T07:29:04.000000Z", "updated_at": "2025-11-19T07:29:04.000000Z", "keterangan_status": "Menunggu diproses tahap awal", "user": { "id": 4, </pre>
7	/api/lembur	POST	<pre> { "message": "Pengajuan lembur berhasil dikirim", "data": { "user_id": 4, "tanggal": "2025-08-07", "jam_mulai": "18:00", "jam_selesai": "21:00", "deskripsi": "Lembur menyelesaikan laporan akhir bulan", "status": "Pending", "updated_at": "2025-11-19T07:29:04.000000Z", "created_at": "2025-11-19T07:29:04.000000Z", "id": 1, "keterangan_status": "Menunggu diproses tahap awal", "user": { "id": 4, "peran_id": 2, "jabatan_id": 5, "departemen_id": 2, </pre>
8	/api/lembur/{id}/approve	PUT	<pre> { "message": "Lembur disetujui tahap awal", "step": 1, "status": "Proses", "data": { "id": 1, "user_id": 4, "tanggal": "2025-08-07", "jam_mulai": "18:00:00", "jam_selesai": "21:00:00", "deskripsi": "Lembur menyelesaikan laporan akhir bulan", "catatan_penolakan": null, "status": "Proses", "approval_step": 1, "created_at": "2025-11-19T07:29:04.000000Z", "updated_at": "2025-11-19T07:31:23.000000Z", "keterangan_status": "Menunggu persetujuan final", "user": { </pre>
9	/api/lembur/{id}/decline	PUT	<pre> { "message": "Lembur ditolak dengan catatan revisi", "step": 3, "status": "Ditolak", "catatan_penolakan": "kamu pemalas", "data": { "id": 1, "user_id": 4, "tanggal": "2025-08-07", "jam_mulai": "18:00:00", "jam_selesai": "21:00:00", "deskripsi": "Lembur menyelesaikan laporan akhir bulan", "catatan_penolakan": "kamu pemalas", "status": "Ditolak", "approval_step": 3, "created_at": "2025-11-19T07:29:04.000000Z", "updated_at": "2025-11-19T07:32:24.000000Z", "keterangan_status": "Lembur Anda ditolak, silahkan ajukan lembur kembali", </pre>
10	/api/cuti	GET	<pre> { "message": "Data cuti berhasil diambil", "data": [{ "id": 1, "user_id": 4, "tipe_cuti": "Sakit", "tanggal_mulai": "2025-08-25", "tanggal_selesai": "2025-08-27", "alasan": "malas", "status": "Pending", "catatan_penolakan": null, "approval_step": 0, "created_at": "2025-11-19T07:34:26.000000Z", "updated_at": "2025-11-19T07:34:26.000000Z", "keterangan_status": "Menunggu diproses tahap awal", "user": { "id": 4, </pre>

11	/api/cuti	POST	<pre>{ "message": "Pengajuan cuti berhasil dikirim", "data": { "user_id": 4, "tipe_cuti": "sakit", "tanggal_mulai": "2025-08-25", "tanggal_selesai": "2025-08-27", "alasan": "malas", "status": "Pending", "updated_at": "2025-11-19T07:34:26.000000Z", "created_at": "2025-11-19T07:34:26.000000Z", "id": 1, "keterangan_status": "Menunggu diproses tahap awal", "user": { "id": 4, "peran_id": 2, "jabatan_id": 5, "departemen_id": 2, </pre>
12	/api/cuti/(id)/approve	PUT	<pre>{ "message": "Cuti disetujui tahap awal", "step": 1, "status": "Proses", "data": { "id": 1, "user_id": 4, "tipe_cuti": "Sakit", "tanggal_mulai": "2025-08-25", "tanggal_selesai": "2025-08-27", "alasan": "malas", "status": "Proses", "catatan_penolakan": null, "approval_step": 1, "created_at": "2025-11-19T07:34:26.000000Z", "updated_at": "2025-11-19T07:36:31.000000Z", "keterangan_status": "Menunggu persetujuan final", "user": { </pre>
13	/api/cuti/(id)/decline	PUT	<pre>{ "message": "Cuti ditolak dengan catatan revisi", "step": 3, "status": "Ditolak", "catatan_penolakan": "kamu pemalas", "data": { "id": 1, "user_id": 4, "tipe_cuti": "Sakit", "tanggal_mulai": "2025-08-25", "tanggal_selesai": "2025-08-27", "alasan": "malas", "status": "Ditolak", "catatan_penolakan": "kamu pemalas", "approval_step": 3, "created_at": "2025-11-19T07:34:26.000000Z", "updated_at": "2025-11-19T07:37:05.000000Z", "keterangan_status": "Cuti Anda ditolak, silahkan ajukan cuti kembali", </pre>
14	/api/tugas	GET	<pre>{ "message": "Data tugas berhasil diambil", "data": [{ "id": 1, "user_id": 7, "nama_tugas": "Membuat laporan bulanan tes it", "tanggal_penugasan": "2025-10-13T01:00:00.000000Z", "batas_penugasan": "2025-10-13T10:00:00.000000Z", "radius_meter": 100, "tugas_lat": "1.1249480", "tugas_lng": "104.0290258", "lampiran_lat": null, "lampiran_lng": null, "instruksi_tugas": "asek lole", "status": "Proses", "waktu_upload": null, "menit_terlambat": null, </pre>
15	/api/tugas	POST	<pre>{ "message": "Tugas berhasil dibuat", "data": { "user_id": 7, "nama_tugas": "Membuat laporan bulanan tes it", "tanggal_penugasan": "2025-10-13T01:00:00.000000Z", "batas_penugasan": "2025-10-13T10:00:00.000000Z", "instruksi_tugas": "asek lole", "tugas_lat": "1.1249480", "tugas_lng": "104.0290258", "radius_meter": "100", "status": "Proses", "updated_at": "2025-11-19T07:39:08.000000Z", "created_at": "2025-11-19T07:39:08.000000Z", "id": 1, "user": { "id": 7, "peran_id": 8, </pre>

16	/api/tugas/(id)	PUT	<pre> { "message": "Tugas berhasil diperbarui", "data": { "id": 2, "user_id": 3, "nama_tugas": "Membuat laporan bulanan tes it", "tanggal_penugasan": "2025-10-13T01:00:00.000000Z", "batas_penugasan": "2025-10-13T10:00:00.000000Z", "radius_meter": 100, "tugas_lat": "1.1249480", "tugas_lng": "104.0290258", "lampiran_lat": null, "lampiran_lng": null, "instruksi_tugas": "bal bla edit", "status": "Proses", "waktu_upload": null, "menit_terlambat": null, "terlambat": null, } } </pre>
17	/api/tugas/(id)	DELETE	<pre> { "message": "Tugas berhasil dihapus" } </pre>
18	/api/tugas/(id)/upload-file	POST	<pre> { "message": "Lampiran berhasil diupload!", "data": { "id": 2, "user_id": 3, "nama_tugas": "Membuat laporan bulanan tes it", "tanggal_penugasan": "2025-10-13T01:00:00.000000Z", "batas_penugasan": "2025-10-13T10:00:00.000000Z", "radius_meter": 100, "tugas_lat": "1.1249480", "tugas_lng": "104.0290258", "lampiran_lat": "1.1249480", "lampiran_lng": "104.0290258", "instruksi_tugas": "bal bla edit", "status": "Menunggu Admin", "waktu_upload": "2025-11-19T07:43:36.000000Z", "menit_terlambat": 53143.608919649996, "terlambat": true, } } </pre>
19	/api/departemen	GET	<pre> { "message": "Data departemen berhasil diambil", "data": [{ "id": 1, "nama_departemen": "IT", "created_at": "2025-11-19T07:28:03.000000Z", "updated_at": "2025-11-19T07:28:03.000000Z" }, { "id": 2, "nama_departemen": "Office", "created_at": "2025-11-19T07:28:03.000000Z", "updated_at": "2025-11-19T07:28:03.000000Z" }, { "id": 3, "nama_departemen": "Maintenance", }] } </pre>
20	/api/departemen	POST	<pre> { "message": "Departemen berhasil dibuat", "data": { "nama_departemen": "testing", "created_at": "2025-11-19T07:47:49.000000Z", "updated_at": "2025-11-19T07:47:49.000000Z", "id": 5 } } </pre>
21	/api/departemen/(id)	PUT	<pre> { "message": "Departemen berhasil diperbarui", "data": { "id": 5, "nama_departemen": "edit", "created_at": "2025-11-19T07:47:49.000000Z", "updated_at": "2025-11-19T07:48:29.000000Z" } } </pre>
22	/api/departemen/(id)	DELETE	<pre> { "message": "Departemen berhasil dihapus" } </pre>

23	/api/peran	GET	<pre> { "message": "Data peran berhasil diambil", "data": [{ "id": 1, "nama_peran": "Admin Super", "created_at": "2025-11-19T07:28:03.000000Z", "updated_at": "2025-11-19T07:28:03.000000Z", "fitur": [{ "id": 1, "nama_fitur": "web", "deskripsi": "User hanya dapat login melalui aplikasi web", "created_at": "2025-11-19T07:28:03.000000Z", "updated_at": "2025-11-19T07:28:03.000000Z", "pivot": { "peran_id": 1, "fitur_id": 1 } }] }] } </pre>
24	/api/peran	POST	<pre> { "message": "Peran berhasil dibuat", "data": { "nama_peran": "TESTING", "updated_at": "2025-11-19T07:51:26.000000Z", "created_at": "2025-11-19T07:51:26.000000Z", "id": 10, "fitur": [{ "id": 1, "nama_fitur": "web", "deskripsi": "User hanya dapat login melalui aplikasi web", "created_at": "2025-11-19T07:28:03.000000Z", "updated_at": "2025-11-19T07:28:03.000000Z", "pivot": { "peran_id": 10, "fitur_id": 1 } }] } } </pre>
25	/api/peran/(id)	PUT	<pre> { "message": "Peran berhasil diperbarui", "data": { "id": 4, "nama_peran": "edit testing", "created_at": "2025-11-19T07:28:03.000000Z", "updated_at": "2025-11-19T07:51:51.000000Z", "fitur": [{ "id": 1, "nama_fitur": "web", "deskripsi": "User hanya dapat login melalui aplikasi web", "created_at": "2025-11-19T07:28:03.000000Z", "updated_at": "2025-11-19T07:28:03.000000Z", "pivot": { "peran_id": 4, "fitur_id": 1 } }] } } </pre>
26	/api/peran/(id)	DELETE	<pre> { "message": "Peran berhasil dihapus" } </pre>
27	/api/jabatan	GET	<pre> { "message": "Data jabatan berhasil diambil", "data": [{ "id": 1, "nama_jabatan": "Sales Executive", "created_at": "2025-11-19T07:28:03.000000Z", "updated_at": "2025-11-19T07:28:03.000000Z" }, { "id": 2, "nama_jabatan": "Telemarketing", "created_at": "2025-11-19T07:28:03.000000Z", "updated_at": "2025-11-19T07:28:03.000000Z" }, { "id": 3, "nama_jabatan": "General Manager", }] } </pre>
28	/api/jabatan	POST	<pre> { "message": "Jabatan berhasil dibuat", "data": { "nama_jabatan": "kepala departemen makan", "updated_at": "2025-11-19T07:54:30.000000Z", "created_at": "2025-11-19T07:54:30.000000Z", "id": 11 } } </pre>
29	/api/jabatan/(id)	PUT	<pre> { "message": "Jabatan berhasil diperbarui", "data": { "id": 11, "nama_jabatan": "edit departemen", "created_at": "2025-11-19T07:54:30.000000Z", "updated_at": "2025-11-19T07:54:58.000000Z" } } </pre>
30	/api/jabatan/(id)	DELETE	<pre> { "message": "Jabatan berhasil dihapus" } </pre>

31	/api/user	GET	<pre> { "message": "Data user berhasil diambil", "data": [{ "id": 1, "peran_id": 1, "jabatan_id": 3, "departemen_id": 1, "nama": "User Super", "gaji_per_hari": 500000, "npwp": null, "bpjs_kesehatan": null, "bpjs_ketenagakerjaan": null, "email": "usersuper@gmail.com", "email_verified_at": null, "jenis_kelamin": "Laki-laki", "status_pernikahan": "Menikah", "coba_login": 0, }] } </pre>
32	/api/user	POST	<pre> { "message": "Karyawan berhasil dibuat", "data": { "nama": "TESTING TAMBAH", "email": "testing@gmail.com", "peran_id": "1", "jabatan_id": "1", "departemen_id": "1", "gaji_per_hari": 123456, "npwp": "123456", "bpjs_kesehatan": "123456", "bpjs_ketenagakerjaan": "123456", "jenis_kelamin": "Perempuan", "status_pernikahan": "Menikah", "updated_at": "2025-11-19T07:58:19.000000Z", "created_at": "2025-11-19T07:58:19.000000Z", "id": 14 } } </pre>
33	/api/user/(id)	PUT	<pre> { "message": "User berhasil diperbarui", "data": { "id": 2, "peran_id": "1", "jabatan_id": "1", "departemen_id": "1", "nama": "tes 12312312312312", "gaji_per_hari": "500000.00", "npwp": "123", "bpjs_kesehatan": "123", "bpjs_ketenagakerjaan": "123", "email": "ksi.admin@gmail.com", "email_verified_at": null, "jenis_kelamin": "Perempuan", "status_pernikahan": "Menikah", "coba_login": 0, "terkunci": 0, } } </pre>
34	/api/user/(id)	DELETE	<pre> { "message": "User berhasil dihapus" } </pre>
35	/api/gaji	GET	<pre> { "message": "Data gaji bulan ini berhasil dihitung", "data": { "id": 1, "user": { "id": 1, "nama": "User Supes" }, "total_kehadiran": 0, "gaji_kotor": 0, "gaji_per_hari": "500000.00", "total lembur": 0, "potongan": [], "total_potongan": 0, "gaji_bersih": 0, "status": null, "bulan": 11, } } </pre>
36	/api/gaji/(id)/status	PUT	<pre> { "message": "Status gaji berhasil diupdate", "data": { "id": 3, "user_id": 3, "bulan": 11, "tahun": 2025, "total_kehadiran": 0, "gaji_kotor": "0.00", "total_potongan": "0.00", "detail_potongan": [], "gaji_bersih": "0.00", "status": "Sudah Dibayar", "created_at": "2025-11-19T07:59:31.000000Z", "updated_at": "2025-11-19T08:01:05.000000Z", "user": { "id": 3, "peran_id": 1, } } } </pre>

37	/api/gaji/export	GET	
38	/api/potongan_gaji	GET	<pre>{ "message": "Data potongan gaji berhasil diambil", "data": [{ "id": 1, "nama_potongan": "BPJS Kesehatan", "persen": "12.00", "created_at": "2025-11-19T08:09:28.000000Z", "updated_at": "2025-11-19T08:09:28.000000Z" }] }</pre>
39	/api/potongan_gaji	POST	<pre>{ "message": "Potongan berhasil ditambahkan", "data": { "nama_potongan": "BPJS Kesehatan", "persen": 12, "updated_at": "2025-11-19T08:09:28.000000Z", "created_at": "2025-11-19T08:09:28.000000Z", "id": 1 } }</pre>
40	/api/potongan_gaji/(id)	PUT	<pre>{ "message": "Potongan berhasil diperbarui", "data": { "id": 1, "nama_potongan": "BPJS Kesehatan edit", "persen": "12.00", "created_at": "2025-11-19T08:09:28.000000Z", "updated_at": "2025-11-19T08:10:36.000000Z" } }</pre>
41	/api/potongan_gaji/(id)	DELETE	<pre>{ "message": "Potongan berhasil dihapus" }</pre>
42	/api/kantor	GET	<pre>{ "message": "Data kantor berhasil diambil", "data": { "id": 1, "jam_masuk": "08:00", "jam_keluar": "17:00", "minimal_keterlambatan": 10, "lat": "1.1250277", "lng": "104.0290078", "radius_meter": 100, "created_at": "2025-11-19T07:28:08.000000Z", "updated_at": "2025-11-19T07:28:08.000000Z" } }</pre>
43	/api/kantor	POST	<pre>{ "message": "Data kantor berhasil diperbarui", "data": { "id": 1, "jam_masuk": "08:30", "jam_keluar": "12:30", "minimal_keterlambatan": 101, "lat": -6.2, "lng": 106.816666, "radius_meter": 1010, "created_at": "2025-11-19T07:28:08.000000Z", "updated_at": "2025-11-19T08:18:42.000000Z" } }</pre>

44	/api/absensi/checkin	POST	<pre> { "message": "Check-in berhasil", "data": { "user_id": 4, "checkin_lat": "-6.2000000", "checkin_lng": "104.0290258", "checkin_date": "2025-08-20", "checkin_time": "08:15:00", "video_user": null, "status": "Tepat Waktu", "updated_at": "2025-11-19T08:25:19.000000Z", "created_at": "2025-11-19T08:25:19.000000Z", "id": 1, "user": { "id": 4, "peran_id": 2, "jabatan_id": 5, "departemen_id": 2, </pre>
45	/api/absensi/checkout	POST	<pre> { "message": "Check-out berhasil", "data": { "id": 1, "user_id": 4, "checkin_lat": "-6.2000000", "checkin_lng": "104.0290258", "checkin_time": "08:15:00", "checkin_date": "2025-08-20", "checkout_lat": "1.1249480", "checkout_lng": "104.0290258", "checkout_time": "08:15:00", "checkout_date": "2025-08-20", </pre>
46	/api/absensi	GET	<pre> { "message": "Data absensi berhasil diambil", "data": [{ "id": 1, "user_id": 4, "checkin_lat": "-6.2000000", "checkin_lng": "104.0290258", "checkin_time": "08:15:00", "checkin_date": "2025-08-20", "checkout_lat": "1.1249480", "checkout_lng": "104.0290258", "checkout_time": "08:15:00", "checkout_date": "2025-08-20", "video_user": null, "status": "Tepat Waktu", "created_at": "2025-11-19T08:25:19.000000Z", "updated_at": "2025-11-19T08:30:27.000000Z", </pre>
47	/api/log	GET	<pre> { "message": "Daftar Activity Log", "data": [{ "id": 38, "user": "User Super", "action": "Login", "module": "User", "description": "User Super (usersuper@gmail.com) berhasil login", "changes": null, "created_at": "19-11-2025 15:34:08" }, { "id": 37, "user": "Desy", "action": "Check-out", "module": "Absensi", "description": "Karyawan Desy melakukan check-out", </pre>
48	/api/pengingat	GET	<pre> { "message": "Data pengingat berhasil diambil", "data": [{ "id": 1, "judul": "Meeting Bulanan", "deskripsi": "Meeting evaluasi project", "tanggal_jatuh_tempo": "05-09-2025 23:59:59", "status": "Pending", "PIC": "Admin Super", "sisa_hari": "74 hari yang lalu", "sisa_jam": null, "relative": "74 hari yang lalu" }] } </pre>

49	/api/pengingat	POST	<pre> { "message": "Pengingat berhasil dibuat", "data": { "peran_id": 1, "judul": "Meeting Bulanan", "deskripsi": "Meeting evaluasi project", "tanggal_jatuh_tempo": "2025-09-05T16:59:59.000000Z", "updated_at": "2025-11-19T08:35:33.000000Z", "created_at": "2025-11-19T08:35:33.000000Z", "id": 1, "peran": { "id": 1, "nama_peran": "Admin Super", "created_at": "2025-11-19T07:28:03.000000Z", "updated_at": "2025-11-19T07:28:03.000000Z" } } } </pre>
50	/api/pengingat/(id)	PUT	<pre> { "message": "Pengingat berhasil diperbarui", "data": { "id": 1, "peran_id": 1, "judul": "Meeting Bulanan", "deskripsi": "Meeting evaluasi project", "tanggal_jatuh_tempo": "2025-09-03T16:59:59.000000Z", "status": "Selesai", "last_notified_at": null, "created_at": "2025-11-19T08:35:33.000000Z", "updated_at": "2025-11-19T08:39:28.000000Z" } } </pre>
51	/api/pengingat/(id)	DELETE	<pre> { "message": "Pengingat berhasil dihapus" } </pre>
52	/api/danger/cuti/months	GET	<pre> [{ "tahun": 2025, "bulan": 11, "jumlah": 1 }] </pre>
53	/api/danger/lembur/months	GET	<pre> [{ "tahun": 2025, "bulan": 11, "jumlah": 1 }] </pre>
54	/api/danger/tugas/months	GET	<pre> [{ "tahun": 2025, "bulan": 11, "jumlah": 1 }] </pre>
55	/api/danger/log/months	GET	<pre> [{ "tahun": 2025, "bulan": 11, "jumlah": 41 }] </pre>

56	/api/danger/absensi/months	GET	<pre>[{ "tahun": 2025, "bulan": 11, "jumlah": 1 }]</pre>
57	/api/danger/gaji/months	GET	<pre>[{ "tahun": 2025, "bulan": 11, "jumlah": 13 }]</pre>
58	/api/danger/cuti/reset	POST	<pre>{ "message": "Sebanyak 1 data cuti bulan 11 tahun 2025 berhasil dihapus" }</pre>
59	/api/danger/lembur/reset	POST	<pre>{ "message": "Sebanyak 1 data lembur bulan 11 tahun 2025 berhasil dihapus" }</pre>
60	/api/danger/tugas/reset	POST	<pre>{ "message": "Sebanyak 1 data tugas + semua file lampiran berhasil dihapus." }</pre>
61	/api/danger/log/reset	POST	<pre>{ "message": "Sebanyak 51 log aktivitas bulan 11 tahun 2025 berhasil dihapus" }</pre>
62	/api/danger/absensi/reset	POST	<pre>{ "message": "Sebanyak 1 data absensi bulan 11 tahun 2025 berhasil dihapus" }</pre>
63	/api/danger/gaji/reset	POST	<pre>{ "message": "Sebanyak 1 data gaji bulan 11 tahun 2025 berhasil dihapus" }</pre>
64	/api/akun/terkunci	GET	<pre>{ "message": "Daftar akun terkunci", "data": [{ "id": 4, "peran_id": 2, "jabatan_id": 5, "departemen_id": 2, "nama": "Desy", "gaji_per_hari": "200000.00", "npwp": null, "bpjs_kesehatan": null, "bpjs_ketenagakerjaan": null, "email": "desy@gmail.com", "email_verified_at": null, "jenis_kelamin": "Perempuan", "status_pernikahan": "Belum Menikah", "coba_login": 2 }] }</pre>
65	/api/akun/(id)/terkunci	POST	<pre>{ "message": "Akun Desy berhasil dibuka kembali", "data": { "id": 4, "peran_id": 2, "jabatan_id": 5, "departemen_id": 2, "nama": "Desy", "gaji_per_hari": "200000.00", "npwp": null, "bpjs_kesehatan": null, "bpjs_ketenagakerjaan": null, "email": "desy@gmail.com", "email_verified_at": null, "jenis_kelamin": "Perempuan", "status_pernikahan": "Belum Menikah", "coba_login": 0, "terkunci": false, } }</pre>

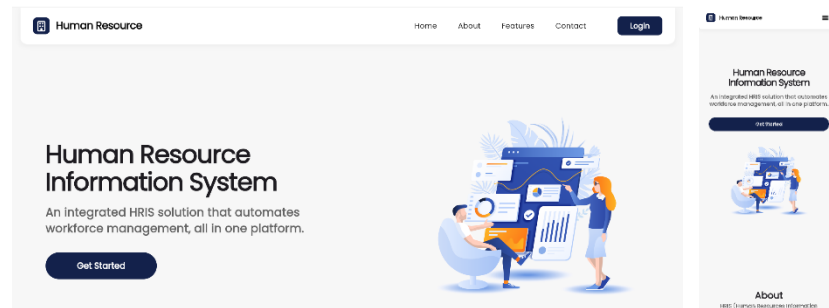
66	/api/device	GET	<pre> { "message": "Daftar semua device terdaftar", "data": [{ "id": 1, "user_id": 7, "device_id": "BADWwda12sqs21", "device_model": "Oppo", "device_manufacturer": "KWQ_123", "device_version": "awd21", "device_hash": "aawdadq23ex9yex9q83exhqwiuedawdqw", "created_at": null, "updated_at": null, "user": { "id": 7, "peran_id": 8, "jabatan_id": 7, "departemen_id": 1, </pre>
67	/api/device/(id)/reset	POST	<pre> { "message": "Device untuk akun User Super berhasil direset. User bisa login dari device baru." } </pre>
68	/api/pengaturan	GET	<pre> { "message": "Pengaturan berhasil diambil", "data": { "user_id": 1, "tema": "gelap", "bahasa": "indonesia", "updated_at": "2025-11-19T09:13:15.000000Z", "created_at": "2025-11-19T09:13:15.000000Z", "id": 1 } } </pre>
69	/api/pengaturan	POST	<pre> { "message": "Pengaturan berhasil diperbarui", "data": { "id": 1, "user_id": 1, "tema": "terang", "bahasa": "indonesia", "created_at": "2025-11-19T09:13:15.000000Z", "updated_at": "2025-11-19T09:13:54.000000Z" } } </pre>
70	/api/forgot-password	POST	<pre> { "status": "success", "message": "Link reset password sudah dikirim ke email Anda." } </pre>

4.1.3 Implementasi Pada Sisi UI

Pada subbab ini akan menjelaskan implementasi pada sisi antarmuka pengguna (UI), termasuk tampilan, respons terhadap aksi pengguna, serta keterangan pada setiap halaman. Berikut merupakan hasil implementasi UI tersebut:

1. Lending Page

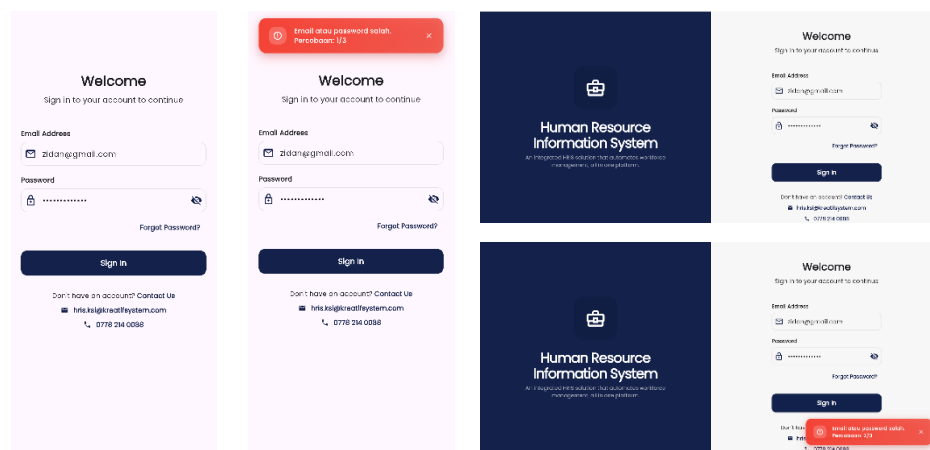
Lending page merupakan halaman awal yang akan di kunjungi oleh *user* saat membuka web maupun aplikasi.



Gambar 4.2 Implementasi *Lending Page*

2. Login

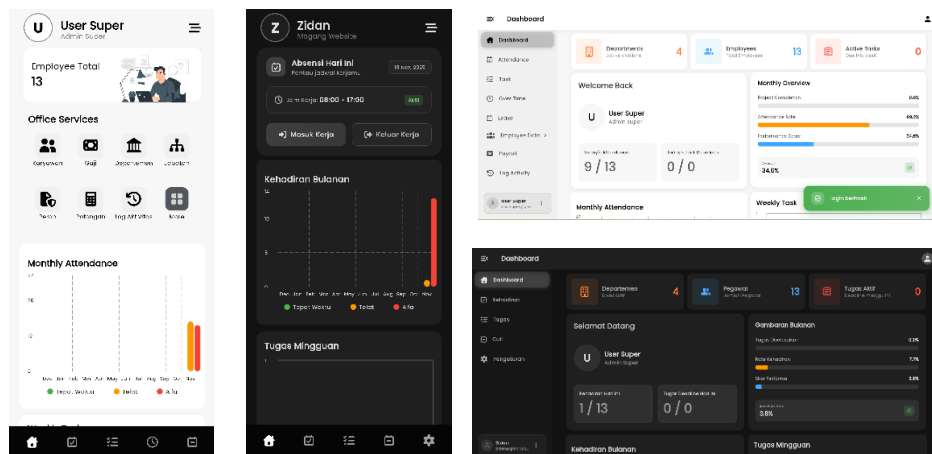
Jika *user* salah memasukkan kata sandi sebanyak tiga kali berturut-turut, maka akun akan terkunci secara otomatis. Namun, apabila aktor berhasil *login* sebelum mencapai tiga kali percobaan gagal, jumlah percobaan tersebut akan direset menjadi nol sehingga perhitungan percobaan *login* dimulai kembali pada *login* berikutnya. Untuk aktor yang melakukan *login* melalui aplikasi mobile, perangkat yang digunakan saat *login* pertama kali akan otomatis terdaftar di dalam *database*. Mekanisme ini bertujuan agar aktor tersebut hanya dapat *login* menggunakan satu perangkat. Sementara itu, aktor yang hanya memiliki akses *login* melalui web tidak dapat *login* melalui aplikasi mobile. Adapun aktor yang memiliki akses *login* melalui web dan aplikasi dapat menggunakan kedua *platform*.



Gambar 4.3 Implementasi *Login*

3. Dashboard

Saat *user* memiliki fitur absensi *check-in* dan *check-out*, pada halaman *dashboard* akan otomatis muncul tombol *check-in* dan *check-out*. Berikut juga ditampilkan *dashboard* dengan tema gelap dan terang. Pada halaman *dashboard* juga terdapat grafik yang akan otomatis menyesuaikan berdasarkan fitur *user*. Saat *user* memiliki fitur melihat semua tugas, akan muncul seluruh data grafik tugas dari karyawan yang terdaftar. Sedangkan jika *user* hanya memiliki fitur melihat tugas sendiri, grafik yang ditampilkan hanya tugas untuk dirinya sendiri. Begitu pula dengan fitur absensi, saat *user* memiliki fitur melihat seluruh absensi, grafik akan otomatis menampilkan seluruh data absensi karyawan. Sedangkan saat *user* hanya memiliki fitur melihat absensi sendiri, grafik yang ditampilkan hanya untuk akun *user* tersebut.

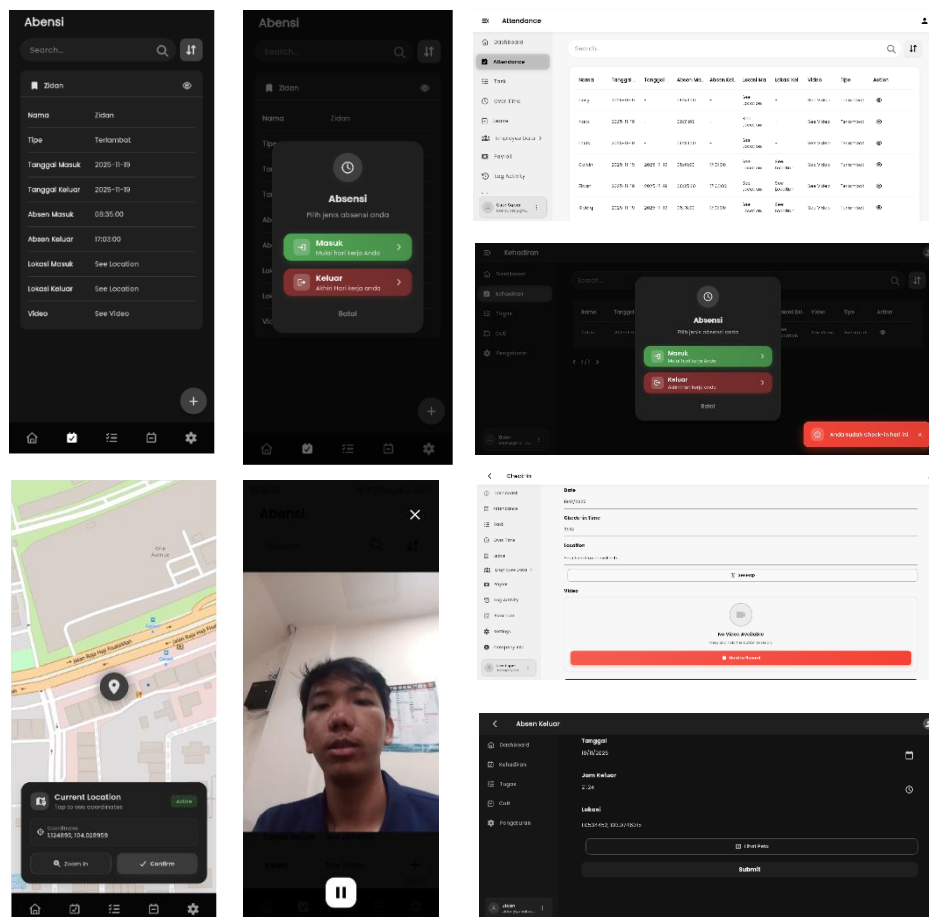


Gambar 4.4 Implementasi *Dashboard*

4. Kehadiran

Pada *page* absensi, *user* yang memiliki fitur *check-in* dan *check-out* akan otomatis memunculkan tombol tambah pada halaman absensi. Saat tombol diklik, akan muncul opsi *check-in* atau *check-out*. *User* dapat melihat lokasi *check-in*, *check-out*, presensi sebelumnya, dan lokasi saat ini. *User* yang memiliki fitur melihat semua absensi pada *page* absensi akan menampilkan seluruh daftar absensi karyawan, sedangkan *user* yang hanya

memiliki fitur melihat absensi sendiri akan menampilkan daftar absensinya sendiri. *User* hanya dapat melakukan *check-in* di area sekitar kantor, dan sistem akan otomatis *generate* lokasi *user* saat itu. Saat *submit*, data akan terekam dan *user* dapat melakukan *record* ulang serta melihat hasil *record* sebelum melakukan *submit*. Saat *check-out*, *user* juga akan otomatis di-*generate* lokasinya dan dapat melakukan *check-out* di berbagai lokasi.

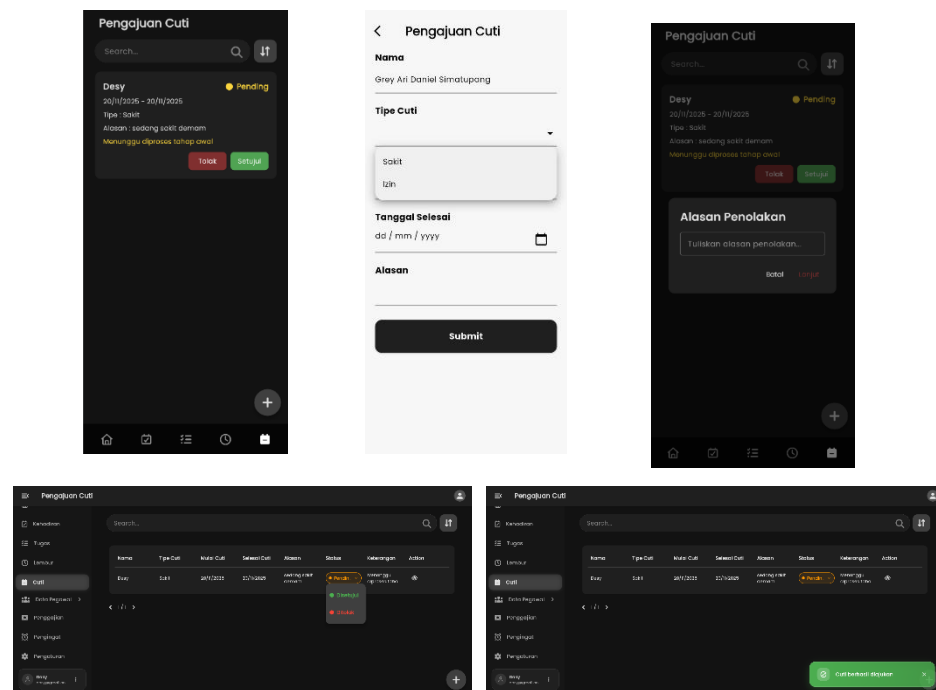


Gambar 4.5 Implementasi Kehadiran

5. Cuti

Pada halaman cuti, *user* yang memiliki fitur lihat semua cuti akan menampilkan seluruh data cuti, sedangkan *user* yang memiliki fitur lihat cuti sendiri hanya akan menampilkan data cuti miliknya. *User* yang memiliki fitur tambah cuti akan secara otomatis menampilkan tombol tambah pada pojok kanan bawah untuk mengakses halaman tambah cuti.

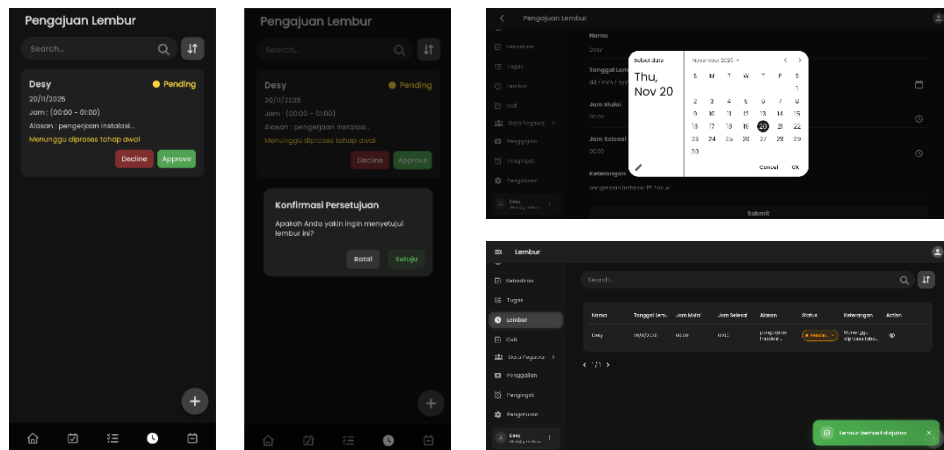
User yang memiliki fitur *approve* atau *decline* cuti akan menampilkan tombol *approve* dan *decline*. User hanya dapat mengajukan cuti baru jika pengajuan cuti sebelumnya telah disetujui atau ditolak. Penolakan (*decline*) cuti akan disertai dengan catatan penolakan dari *approver*.



Gambar 4.6 Implementasi Cuti

6. Lembur

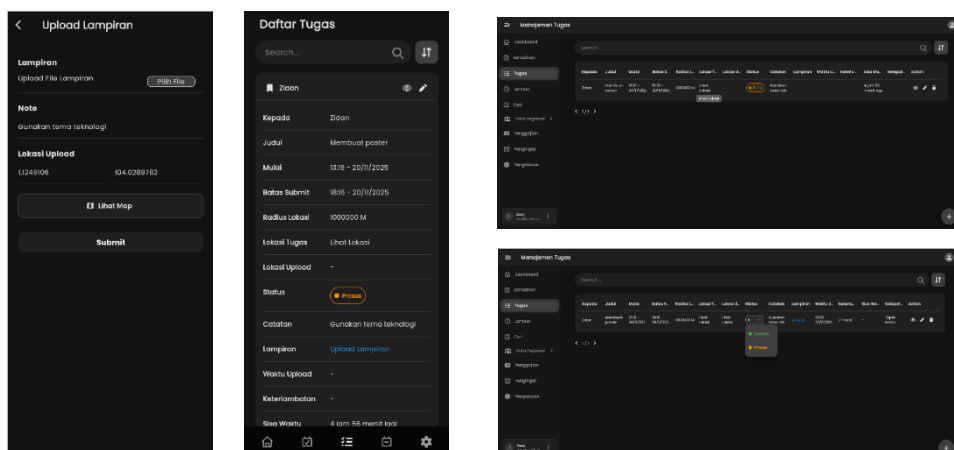
Pada halaman lembur, *user* yang memiliki fitur lembur dapat mengakses halaman tersebut. *User* yang memiliki fitur tambah lembur akan secara otomatis menampilkan tombol tambah pada pojok kanan bawah halaman yang jika diklik akan mengarah pada *modal form* tambah lembur. *User* tidak dapat mengajukan lembur baru jika pengajuan lembur sebelumnya belum disetujui (*approve*) atau ditolak. Penolakan lembur akan disertai dengan catatan penolakan. *User* yang memiliki fitur lihat semua lembur akan menampilkan seluruh data lembur karyawan, sedangkan *user* yang memiliki fitur lihat lembur sendiri hanya akan menampilkan data lembur miliknya. *User* yang memiliki fitur *approve* dan *decline* lembur akan secara otomatis menampilkan tombol *approve* dan *decline*.



Gambar 4.7 Implementasi Lembur

7. Tugas

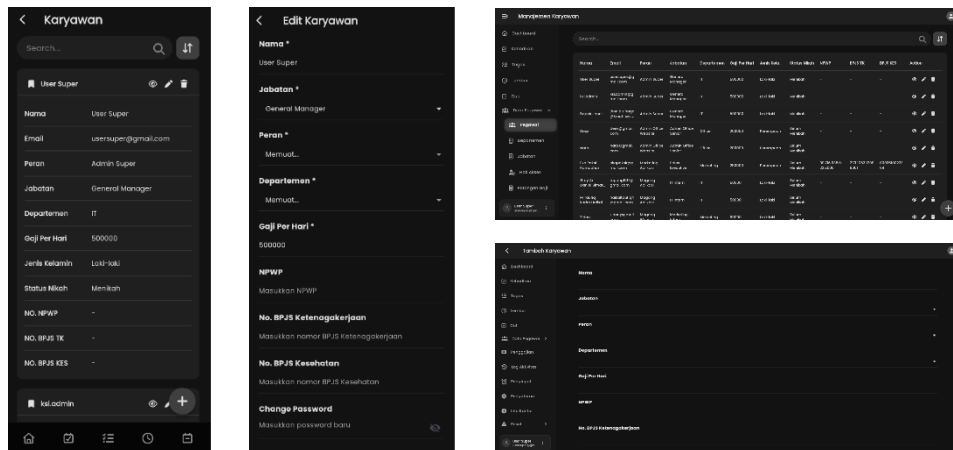
Pada halaman tugas, *user* yang memiliki fitur lihat semua tugas akan secara otomatis menampilkan seluruh daftar tugas karyawan, sedangkan *user* yang memiliki fitur lihat tugas sendiri hanya akan menampilkan daftar tugas yang ditujukan kepadanya. *User* yang memiliki fitur tambah, edit, dan hapus tugas akan secara otomatis menampilkan tombol-tombol tersebut pada halaman. *User* yang memiliki fitur *upload* lampiran dan ditugaskan dapat mengunggah lampiran pada tugas dengan PIC dirinya sendiri. *User* hanya dapat mengunggah lampiran di area yang ditugaskan. Jika *user* terlambat mengunggah lampiran, sistem akan secara otomatis menghasilkan (*generate*) status terlambat. *User* yang memiliki fitur ubah status tugas dapat mengubah status tugas *user*.



Gambar 4.8 Implementasi Tugas

8. Karyawan

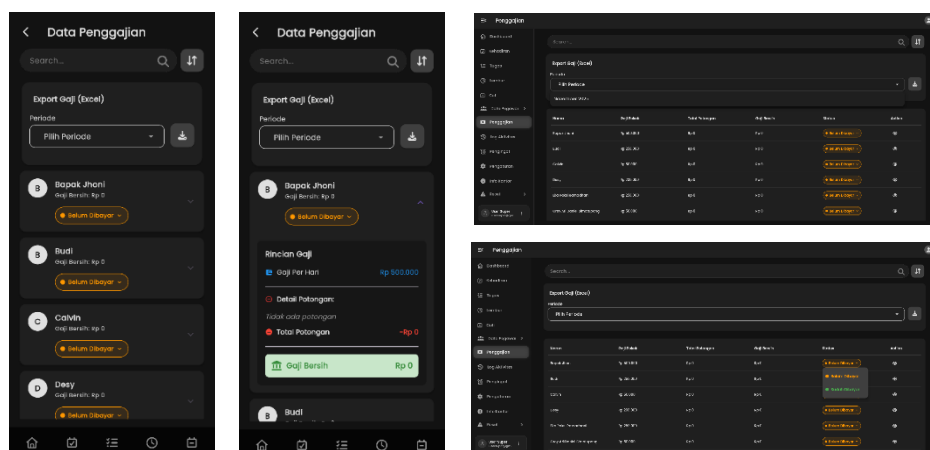
User yang memiliki fitur karyawan akan secara otomatis menampilkan halaman karyawan. Pada halaman karyawan, user dapat melakukan tambah, edit, dan hapus data karyawan.



Gambar 4.9 Implementasi Karyawan

9. Gaji

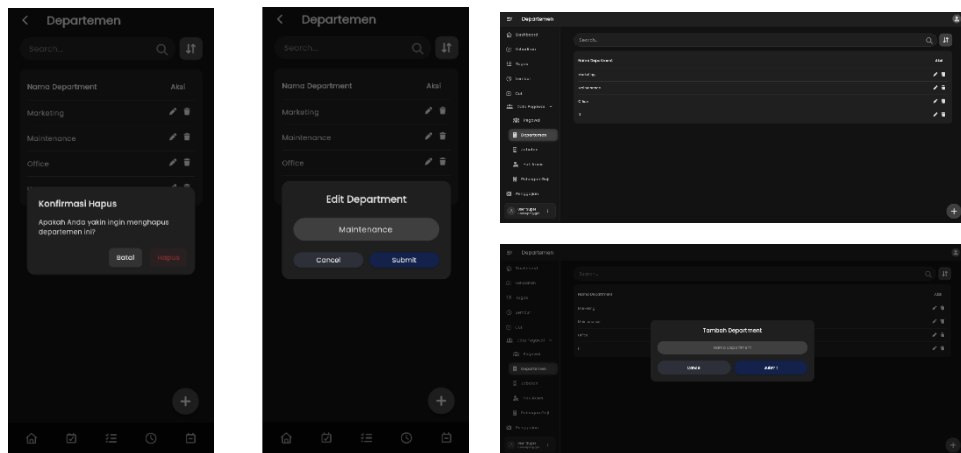
User yang memiliki fitur gaji akan secara otomatis menampilkan halaman gaji karyawan yang terdaftar. Pada halaman gaji, user dapat melihat daftar gaji per hari karyawan, potongan gaji, dan gaji bersih karyawan. User juga dapat melakukan *update* status gaji karyawan dan melakukan *export* gaji karyawan ke Excel sesuai periode yang dipilih. Data yang tampil di halaman karyawan setiap bulan akan otomatis ter-reset karena telah dibatasi untuk filter GET data gaji perbulan.



Gambar 4.10 Implementasi Karyawan

10. Departemen

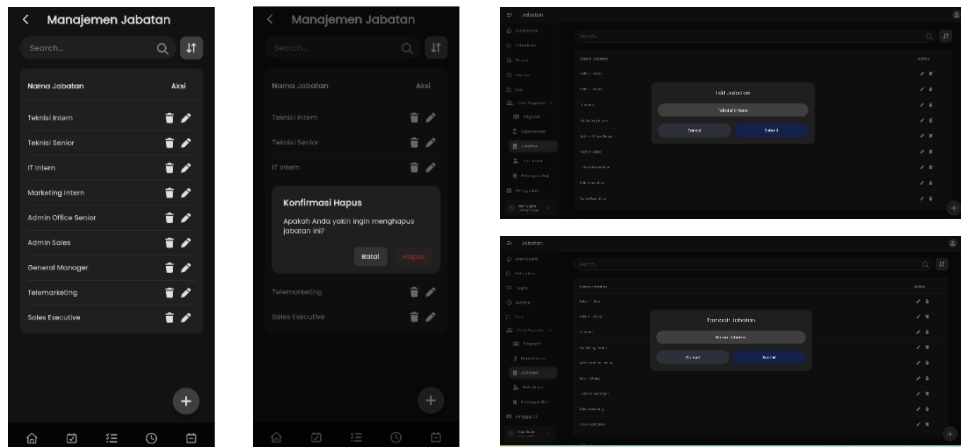
User yang memiliki fitur departemen akan secara otomatis menampilkan halaman departemen. Pada halaman departemen, *user* dapat menambah, mengedit, dan menghapus data departemen. *User* tidak dapat menambahkan atau mengedit nama departemen jika nama tersebut sudah terdaftar karena pada sisi *backend* terdapat validasi *unique*.



Gambar 4.11 Implementasi Departemen

11. Jabatan

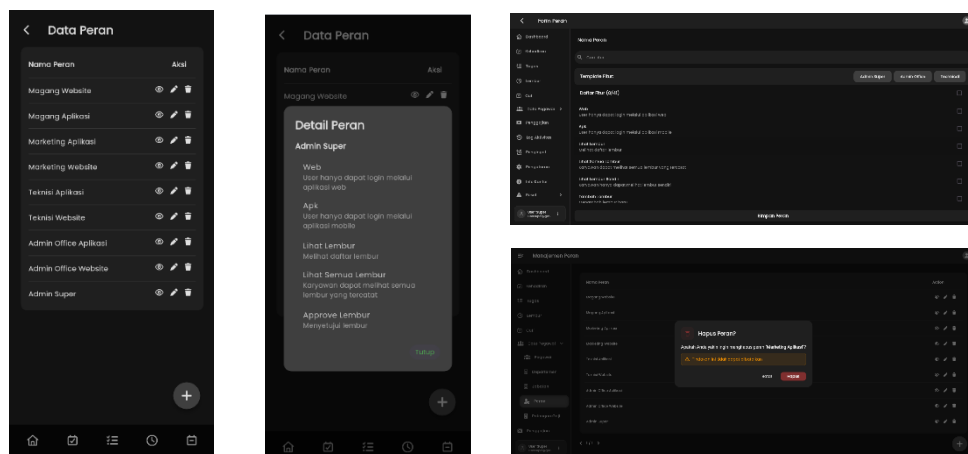
User yang memiliki fitur jabatan akan secara otomatis menampilkan halaman jabatan. Pada halaman jabatan, *user* dapat menambahkan, mengedit, dan menghapus data jabatan. *User* tidak dapat menambahkan atau mengedit nama jabatan jika nama tersebut sudah terdaftar karena pada sisi *backend* terdapat validasi *unique*.



Gambar 4.12 Implementasi Jabatan

12. Peran

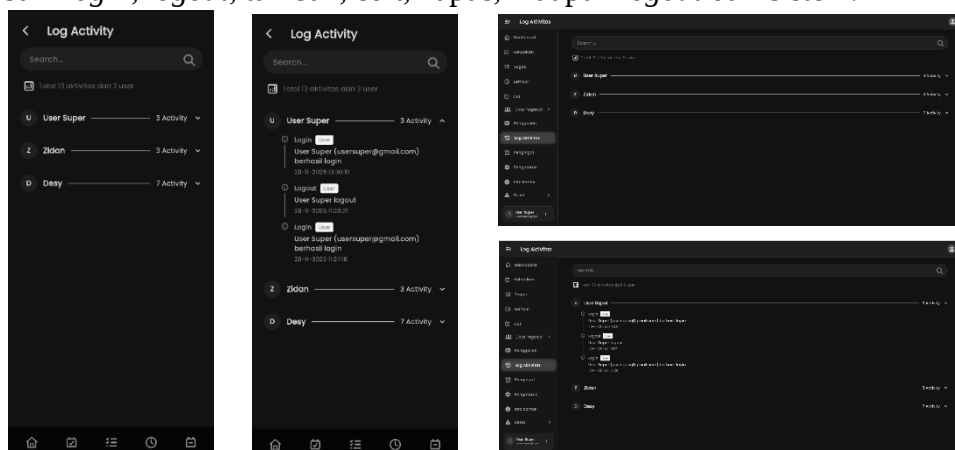
User yang memiliki fitur peran akan secara otomatis menampilkan halaman peran. Pada halaman peran, *user* dapat menambahkan, mengedit, dan menghapus data peran. Saat menambah dan mengedit, *user* pertama akan menambahkan atau mengedit nama peran dan memilih data fitur yang dapat diakses oleh peran tersebut. Terdapat 41 pembagian fitur yang terdaftar. *User* tidak dapat menambahkan atau mengedit nama peran jika nama peran tersebut sudah terdaftar karena pada sisi *backend* terdapat validasi *unique*.



Gambar 4.13 Implementasi Peran

13. Log Aktivitas

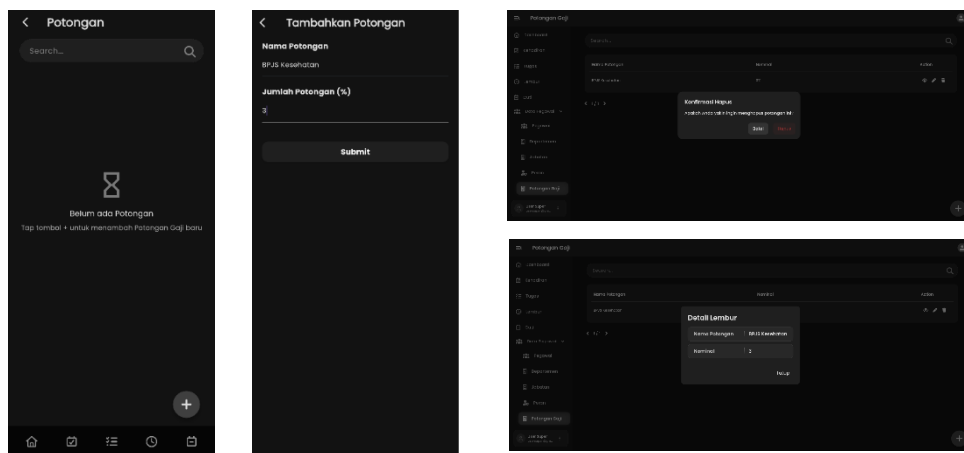
User yang memiliki fitur *log* aktivitas akan secara otomatis menampilkan halaman *log* aktivitas. Pada halaman *log* aktivitas, *user* dapat melihat seluruh daftar *log* aktivitas karyawan yang terdaftar di *database*, baik *login*, *logout*, tambah, edit, hapus, maupun *logout* dari sistem.



Gambar 4.14 Implementasi Log Aktivitas

14. Potongan

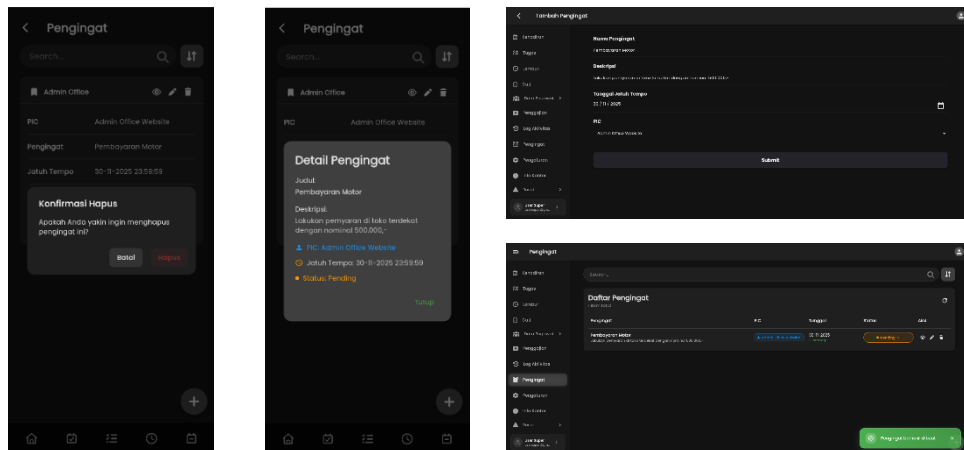
User yang memiliki fitur potongan akan secara otomatis menampilkan halaman potongan. Pada halaman potongan, user dapat menambahkan, mengedit, dan menghapus data potongan. User tidak dapat menambahkan atau mengedit nama potongan jika nama potongan tersebut sudah terdaftar karena pada sisi *backend* terdapat validasi *unique*. Pada halaman potongan, inputan persen tidak boleh lebih dari 100%.



Gambar 4.15 Implementasi Potongan

15. Reminder

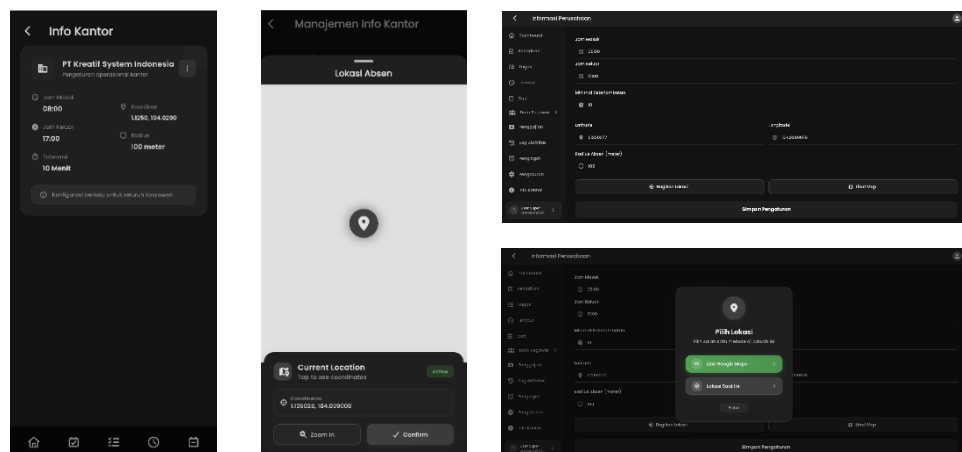
User yang memiliki fitur pengingat akan secara otomatis menampilkan halaman *reminder*. Pada halaman *reminder*, user dapat menambahkan, mengedit, dan menghapus data *reminder*. User tidak dapat menambahkan atau mengedit nama *Reminder* jika nama tersebut sudah terdaftar karena pada sisi *backend* terdapat validasi *unique*. PIC yang terpilih akan secara otomatis dikirimkan *email* pengingat H-7 hari dari tanggal jatuh tempo. Jika data pengingat diubah menjadi selesai, *email* tidak akan dikirim lagi, begitu juga setelah melewati tanggal jatuh tempo.



Gambar 4.16 Implementasi *Reminder*

16. Info Kantor

User yang memiliki fitur kantor akan secara otomatis menampilkan halaman info kantor. Pada halaman info kantor, *user* dapat menambah dan mengedit data kantor. Data kantor tersebut berkaitan dengan fitur absensi yang memuat informasi jam masuk dan minimal keterlambatan. Info kantor juga menyimpan data lokasi kantor dan radius yang berkaitan dengan absensi.

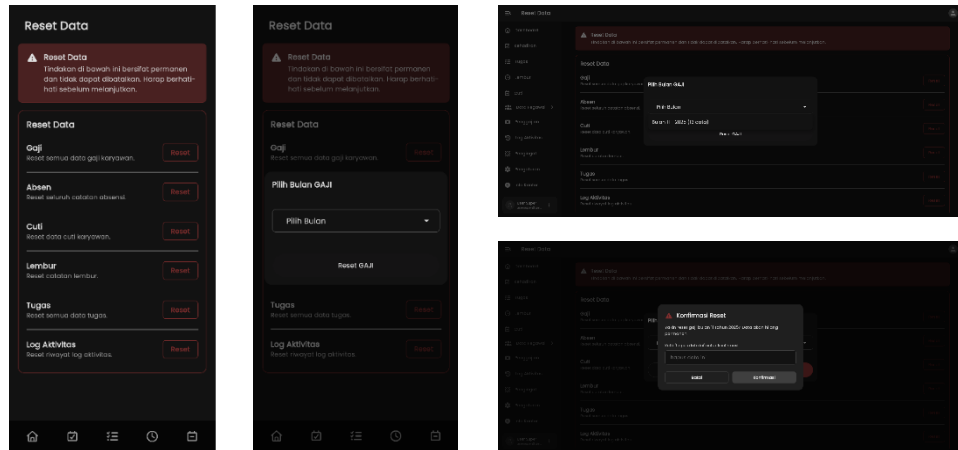


Gambar 4.17 Implementasi Info Kantor

17. Reset Data

User yang memiliki fitur *danger* akan secara otomatis menampilkan halaman *reset data*. Pada halaman *reset data*, *user* dapat melakukan penghapusan data secara massal di tabel per bulan. Pada halaman ini

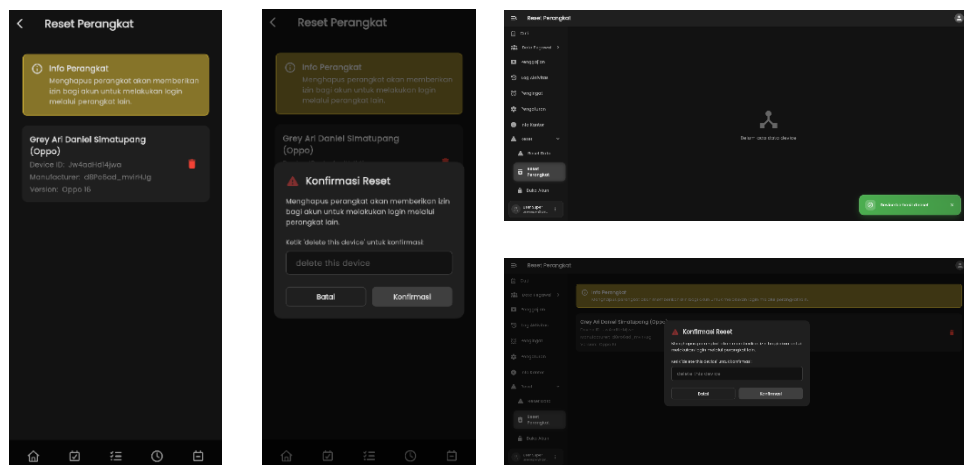
terdapat *reset* gaji, absen, cuti, lembur, tugas, dan *log* aktivitas. Saat mengklik *reset*, sistem akan menampilkan *dropdown* data per bulan, dan saat diklik akan muncul *modal input* konfirmasi.



Gambar 4.18 Implementasi *Reset Data*

18. *Reset Perangkat*

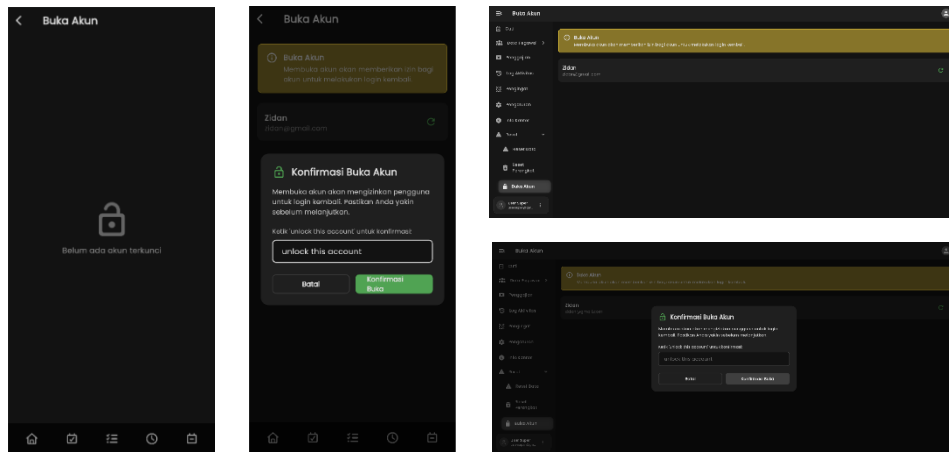
User yang memiliki fitur *reset device* akan secara otomatis menampilkan halaman *reset device*. Halaman ini merupakan halaman daftar *device user* yang *login*, hanya diperuntukkan untuk satu *device* yaitu yang memiliki fitur APK. Pada halaman *reset device*, *user* dapat menghapus data *device user* yang terdaftar agar *user* dapat berganti *device* jika di kemudian hari pengguna aplikasi mengganti HP.



Gambar 4.19 Implementasi *Reset Perangkat*

19. Unlock

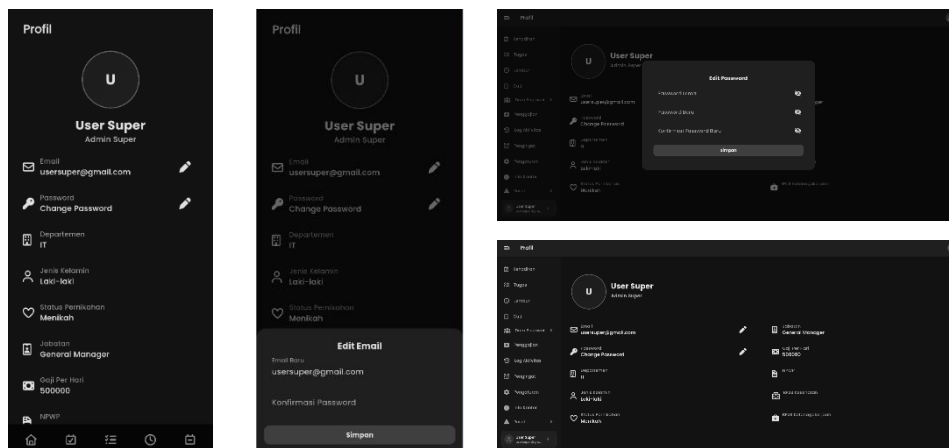
User yang memiliki fitur buka akun akan secara otomatis menampilkan halaman *unlock*. Pada halaman *unlock*, user dapat melihat setiap daftar akun *user* yang terkunci akibat gagal *login* sebanyak 3 kali. Saat user mengklik buka akun terkunci, sistem akan menampilkan *modal* inputan konfirmasi.



Gambar 4.20 Implementasi Reset Unlock

20. Profile

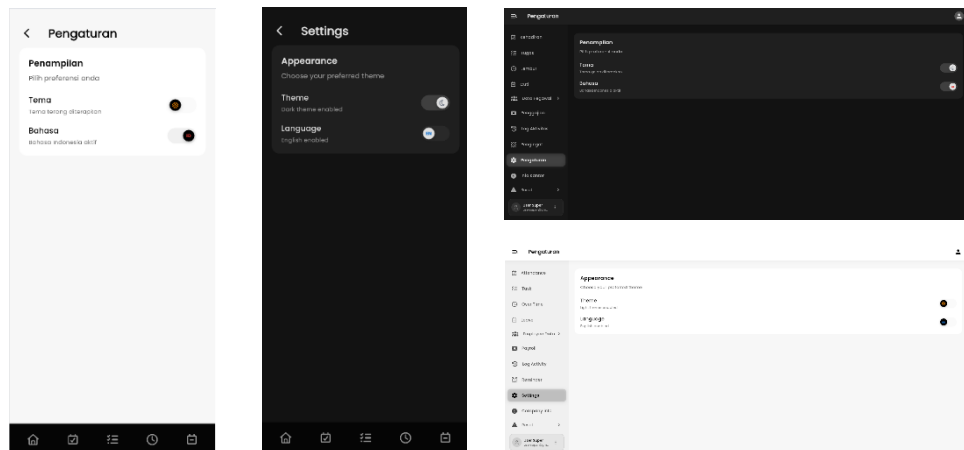
User secara *default* dapat mengunjungi halaman *profile*, di mana pada halaman ini terdapat data *user* yang sedang *login*. Pada halaman ini, *user* juga dapat mengubah *email* yang terdaftar dan *password*. Saat *user* mengklik edit *password* atau *email*, akan muncul *modal form* edit.



Gambar 4.21 Implementasi Reset Profile

21. Pengaturan

User secara *default* dapat mengunjungi halaman pengaturan, di mana *user* dapat melakukan perubahan tema aplikasi dan bahasa aplikasi. Tema yang tersedia yaitu terang dan gelap. Bahasa yang tersedia pada aplikasi adalah Indonesia dan Inggris.



Gambar 4.22 Implementasi Pengaturan

4.2 Pengujian User Acceptance Testing (UAT)

4.2.1 Pengujian Perangkat Lunak (Functional Testing)

Pada tahap ini dilakukan pengujian *fungsi* untuk memastikan bahwa seluruh fitur dalam perangkat lunak bekerja sesuai dengan kebutuhan dan *Skenario* bisnis yang telah ditetapkan sebelumnya. Pengujian dilakukan dengan menggunakan metode *blackbox*, yaitu metode yang berfokus pada pemeriksaan *output* berdasarkan *input* yang diberikan tanpa melihat struktur internal kode program. Dalam penelitian ini, terdapat 28 *fungsi* utama yang menjadi objek pengujian. Berikut hasil uji *fungsi* dengan metode *blackbox*:

1. Testing Scenario Use Case Login

Pengujian ini didasari oleh *Scenario Use Case Login* (Tabel 3.12). Use case scenario mendefinisikan aktor *login* ke dalam sistem, namun pengujian dilakukan secara detail untuk memvalidasi setiap aksi dan respon sistem terhadap setiap aktor. Hal ini untuk memastikan sistem validasi *login* dapat berjalan sesuai kebutuhan.

Tabel 4.2 *Black Box Testing Login*

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Email dan kata sandi terisi dengan benar. Akun memiliki akses fitur web dan APK. Login dilakukan via web dan via aplikasi.</i>	<i>Login berhasil divalidasi dan berpindah ke halaman dashboard.</i>	<i>Login berhasil divalidasi dan berpindah ke halaman dashboard.</i>	✓	
TC02	<i>Email dan kata sandi terisi dengan benar. Akun hanya memiliki akses fitur web. Login dilakukan via web.</i>	<i>Login berhasil divalidasi dan berpindah ke halaman dashboard.</i>	<i>Login berhasil divalidasi dan berpindah ke halaman dashboard.</i>	✓	
TC03	<i>Email dan kata sandi terisi dengan benar. Akun hanya memiliki akses fitur web. Login dilakukan via APK.</i>	<i>Login tidak berhasil divalidasi, tetap berada pada halaman login, dan memunculkan pesan "Login via APK tidak diperbolehkan untuk akun ini".</i>	<i>Login tidak berhasil divalidasi, tetap berada pada halaman login, dan memunculkan pesan "Login via APK tidak diperbolehkan untuk akun ini".</i>	✓	
TC04	<i>Email dan kata sandi terisi dengan benar.</i>	<i>Login berhasil divalidasi dan berpindah ke halaman dashboard.</i>	<i>Login berhasil divalidasi dan berpindah ke halaman dashboard.</i>	✓	

	Akun hanya memiliki akses fitur APK. <i>Login</i> dilakukan via APK.				
TC05	<i>Email</i> dan kata sandi terisi dengan benar. Akun hanya memiliki akses fitur APK. <i>Login</i> dilakukan via web.	<i>Login</i> tidak berhasil divalidasi, tetap berada pada halaman <i>login</i> , dan memunculkan pesan " <i>Login</i> via web tidak diperbolehkan untuk akun ini".	<i>Login</i> tidak berhasil divalidasi, tetap berada pada halaman <i>login</i> , dan memunculkan pesan " <i>Login</i> via web tidak diperbolehkan untuk akun ini".	✓	
TC06	<i>Email</i> terisi dengan benar dan kata sandi terisi dengan salah.	<i>Login</i> tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan " <i>Email</i> atau <i>Password</i> salah". Ketika sudah tiga kali percobaan gagal, akun akan terkunci dan memunculkan pesan "Akun terkunci setelah 3 kali percobaan gagal, hubungi <i>admin</i> ". Jika <i>user</i> berhasil <i>login</i> sebelum percobaan ketiga, penghitung percobaan <i>login</i> akan kembali ke 0.	<i>Login</i> tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan " <i>Email</i> atau <i>Password</i> salah". Setelah 3 kali percobaan gagal, akun terkunci dan tidak bisa <i>login</i> . Jika <i>login</i> berhasil sebelum percobaan ketiga, penghitung percobaan <i>login</i> kembali ke 0.	✓	

TC07	<i>Email</i> terisi dengan benar dan kata sandi tidak terisi.	<i>Login</i> tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan “ <i>Please enter your password</i> ”.	<i>Login</i> tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan “ <i>Please enter your password</i> ”.	✓	
TC08	<i>Email</i> terisi dengan salah dan kata sandi terisi dengan benar.	<i>Login</i> tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan “ <i>Email</i> atau <i>Password</i> salah”.	<i>Login</i> tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan “ <i>Email</i> dan <i>Password</i> salah”.	✓	
TC09	<i>Email</i> terisi dengan salah dan kata sandi terisi dengan salah.	<i>Login</i> tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan “ <i>Email</i> atau <i>Password</i> salah”.	<i>Login</i> tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan “ <i>Email</i> atau <i>Password</i> salah”.	✓	
TC10	<i>Email</i> terisi dengan salah dan kata sandi tidak terisi.	<i>Login</i> tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan “ <i>Please enter your password</i> ”.	<i>Login</i> tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan “ <i>Please enter your password</i> ”.	✓	
TC11	<i>Email</i> tidak terisi dan kata sandi terisi dengan benar.	<i>Login</i> tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan “ <i>Please enter your email</i> ”.	<i>Login</i> tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan “ <i>Please enter your email</i> ”.	✓	

TC12	Email tidak terisi dan kata sandi terisi dengan salah.	Login tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan “Please enter your email”.	Login tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan “Please enter your email”.	✓	
TC13	Email tidak terisi dan kata sandi tidak terisi.	Login tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan “Please enter your email” dan “Please enter your password”.	Login tidak berhasil divalidasi sehingga masih berada pada halaman <i>login</i> dan memunculkan pesan “Please enter your email” dan “Please enter your password”.	✓	

Hasil analisis:

1) Persentase test case yang lolos = $\frac{13}{13} = 100\%$

2) Persentase test case yang gagal = $\frac{0}{13} = 0\%$

3) Persentase test case yang telah diuji = 13 = 100%

4) Persentase test case yang belum diuji = 0 = 0%

Jumlah total test case dalam pengujian = 13

2. Testing Scenario Use Case Manajemen Departemen

Pengujian ini didasari oleh *Scenario Use Case* Manajemen Departemen (Tabel 3.15). Use case scenario mendefinisikan masuk ke dalam halaman departemen, namun pengujian dilakukan secara detail untuk memvalidasi data tambah, edit, dan hapus departemen. Hal ini untuk memastikan manajemen departemen berjalan lancar.

Tabel 4.3 *Black Box* Testing Manajemen Departemen

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	Admin Super mengakses	Sistem menampilkan semua data departemen.	Sistem menampilkan semua data departemen.	✓	

	halaman daftar departemen dengan data departemen tersedia di <i>database</i> .				
TC02	<i>Admin Super</i> mengklik tombol tambah departemen.	Sistem memunculkan <i>modal</i> inputan tambah departemen.	Sistem memunculkan inputan tambah departemen	✓	
TC03	<i>Admin Super</i> mengklik tombol hapus departemen	Sistem memunculkan <i>modal</i> konfirmasi hapus departemen.	Sistem memunculkan <i>modal</i> konfirmasi hapus departemen.	✓	
TC04	Sistem memunculkan <i>modal</i> konfirmasi hapus departemen.	Sistem menutup <i>modal</i> dan menampilkan notifikasi.	Sistem menutup <i>modal</i> dan menampilkan notifikasi.	✓	
TC05	<i>Admin Super</i> mengklik tombol edit pada salah satu data departemen.	Sistem memunculkan <i>modal</i> edit data departemen.	Sistem memunculkan <i>modal</i> edit data departemen.	✓	
TC06	<i>Admin Super</i> mengklik tombol <i>submit</i> pada <i>modal</i> edit departemen.	Sistem menutup <i>modal</i> dan menampilkan pesan notifikasi.	Sistem menutup <i>modal</i> dan menampilkan pesan notifikasi.	✓	

TC07	<i>Admin Super</i> mengklik tombol <i>cancel</i> pada <i>modal</i> edit departemen.	Sistem menutup <i>modal</i> konfirmasi tanpa ada perubahan data.	Sistem menutup <i>modal</i> konfirmasi tanpa ada perubahan data.	✓	
TC08	<i>Admin Super</i> mengisi nama departemen dengan benar dan preferensi bahasa Indonesia/Inggris.	Departemen berhasil disimpan ke <i>database</i> dan sistem menampilkan pesan "Departemen berhasil dibuat"/" <i>Department created successfully</i> ".	Departemen berhasil disimpan ke <i>database</i> dan sistem menampilkan pesan "Departemen berhasil dibuat"/" <i>Department created successfully</i> ".	✓	
TC09	<i>Admin Super</i> tidak mengisi nama departemen. preferensi bahasa Indonesia/Inggris.	Departemen tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama departemen wajib diisi." / " <i>Department name is required.</i> ".	Departemen tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama departemen wajib diisi." / " <i>Department name is required.</i> ".	✓	
TC10	<i>Admin Super</i> mengisi nama departemen yang sudah ada di <i>database</i> dan preferensi bahasa Indonesia/Inggris.	Departemen tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama departemen sudah digunakan"/" <i>Department name has already been taken.</i> ".	Departemen tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama departemen sudah digunakan."/ " <i>Department name has already been taken.</i> ".	✓	
TC11	<i>Admin Super</i> mengisi nama departemen lebih dari 255 karakter	Departemen tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama departemen	Departemen tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama departemen tidak boleh	✓	

	dan preferensi bahasa Indonesia/ Inggris.	tidak boleh lebih dari 255 karakter."/"/ <i>Department name must not exceed 255 characters.</i> ".	lebih dari 255 karakter."/"/ <i>Department name must not exceed 255 characters.</i> ".		
TC12	<i>Admin Super</i> memilih departemen yang ada, mengubah nama departemen dengan nama baru yang valid, dan preferensi bahasa Indonesia/ Inggris.	Data departemen berhasil diperbarui dan sistem menampilkan pesan "Departemen berhasil diperbarui"/"/ <i>Department updated successfully</i> ".	Data departemen berhasil diperbarui dan sistem menampilkan pesan "Departemen berhasil diperbarui"/"/ <i>Department updated successfully</i> ".	✓	
TC13	<i>Admin Super</i> mengubah nama departemen menjadi kosong dan preferensi bahasa Indonesia/ Inggris.	Departemen tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama departemen wajib diisi." / <i>"Department name is required."</i> .	Departemen tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama departemen wajib diisi." / <i>"Department name is required."</i> .	✓	
TC14	<i>Admin Super</i> mengubah nama departemen menjadi nama yang sudah digunakan departemen lain dan preferensi	Departemen tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama departemen sudah digunakan."/ <i>"Department name has already been taken."</i> .	Departemen tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama departemen sudah digunakan."/ <i>"Department name has already been taken."</i> .	✓	

	bahasa Indonesia/ Inggris.				
TC15	<i>Admin Super</i> mengubah nama departemen menjadi lebih dari 255 karakter dan preferensi bahasa Indonesia.	Departemen tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama departemen tidak boleh lebih dari 255 karakter."/"Department name must not exceed 255 characters".	Departemen tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama departemen tidak boleh lebih dari 255 karakter."/"Department name must not exceed 255 characters".	✓	
TC16	<i>Admin Super</i> memilih departemen yang ada untuk dihapus dan preferensi bahasa Indonesia/Inggris.	Departemen berhasil dihapus dari <i>database</i> dan sistem menampilkan pesan "Departemen berhasil dihapus"/"Department deleted successfully".	Departemen berhasil dihapus dari <i>database</i> dan sistem menampilkan pesan "Departemen berhasil dihapus"/"Department deleted successfully".	✓	

Hasil analisis:

$$1) \text{ Persentase test case yang lolos} = \frac{16}{16} = 100\%$$

$$2) \text{ Persentase test case yang gagal} = \frac{0}{16} = 0\%$$

$$3) \text{ Persentase test case yang telah diuji} = 16 = 100\%$$

$$4) \text{ Persentase test case yang belum diuji} = 0 = 0\%$$

Jumlah total test case dalam pengujian = 16

3. Testing Scenario Use Case Manajemen Jabatan

Pengujian ini didasari oleh *Scenario Use Case* Manajemen Jabatan (Tabel 3.14). Use case scenario mendefinisikan masuk ke dalam halaman jabatan, namun pengujian dilakukan secara detail untuk memvalidasi data tambah, edit, dan hapus jabatan. Hal ini untuk memastikan manajemen jabatan berjalan lancar.

Tabel 4.4 *Black Box* Testing Manajemen Jabatan

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin Super</i> mengakses halaman daftar jabatan dengan data jabatan tersedia di <i>database</i> .	Sistem menampilkan semua data jabatan.	Sistem menampilkan semua data jabatan.	✓	
TC02	<i>Admin Super</i> mengklik tombol tambah jabatan.	Sistem memunculkan <i>modal input</i> tambah jabatan.	Sistem memunculkan <i>modal input</i> tambah jabatan.	✓	
TC03	<i>Admin Super</i> mengklik tombol hapus jabatan.	Sistem memunculkan <i>modal</i> konfirmasi hapus jabatan.	Sistem memunculkan <i>modal</i> konfirmasi hapus jabatan.	✓	
TC04	<i>Admin Super</i> mengklik tombol <i>delete</i> pada <i>modal</i> konfirmasi hapus jabatan.	Sistem menutup <i>modal</i> dan menampilkan notifikasi.	Sistem menutup <i>modal</i> dan menampilkan notifikasi.	✓	
TC05	<i>Admin Super</i> mengklik tombol batal pada <i>modal</i> konfirmasi hapus jabatan.	Sistem menutup <i>modal</i> konfirmasi tanpa adanya perubahan data.	Sistem menutup <i>modal</i> konfirmasi tanpa adanya perubahan data.	✓	
TC06	<i>Admin Super</i> mengklik tombol edit pada salah satu data jabatan.	Sistem memunculkan <i>modal</i> edit data jabatan.	Sistem memunculkan <i>modal</i> edit data jabatan.	✓	

TC07	<i>Admin Super</i> mengklik tombol <i>submit</i> pada <i>modal</i> edit jabatan.	Sistem menutup <i>modal</i> dan menampilkan pesan notifikasi.	Sistem menutup <i>modal</i> dan menampilkan pesan notifikasi.	✓	
TC08	<i>Admin Super</i> mengklik tombol <i>cancel</i> pada <i>modal</i> edit jabatan.	Sistem menutup <i>modal</i> konfirmasi tanpa ada perubahan data.	Sistem menutup <i>modal</i> konfirmasi tanpa ada perubahan data.	✓	
TC09	<i>Admin Super</i> mengisi nama jabatan dengan benar dan preferensi bahasa Indonesia/Inggris.	Jabatan berhasil disimpan ke <i>database</i> dan sistem menampilkan pesan "Jabatan berhasil dibuat"/" <i>Job title created</i> <i>successfully</i> ".	Jabatan berhasil disimpan ke <i>database</i> dan sistem menampilkan pesan "Jabatan berhasil dibuat"/" <i>Job title</i> <i>created successfully</i> ".	✓	
TC10	<i>Admin Super</i> tidak mengisi nama jabatan, preferensi bahasa Indonesia/Inggris.	Jabatan tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama jabatan tidak boleh kosong."/ " <i>Job</i> <i>title name cannot be</i> <i>empty</i> ".	Jabatan tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama jabatan tidak boleh kosong."/ " <i>Job title name</i> <i>cannot be empty</i> ".	✓	
TC11	<i>Admin Super</i> mengisi nama jabatan yang sudah ada di <i>database</i> dan preferensi bahasa Indonesia/Inggris.	Jabatan tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama jabatan sudah digunakan."/ " <i>Job</i> <i>title has already been</i> <i>taken</i> ".	Jabatan tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama jabatan sudah digunakan."/ " <i>Job title has</i> <i>already been taken</i> ".	✓	

TC12	<i>Admin Super</i> mengisi nama jabatan lebih dari 255 karakter dan preferensi bahasa Indonesia/Inggris.	Jabatan tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama jabatan tidak boleh lebih dari 255 karakter."/" <i>Job title must not exceed 255 characters.</i> ".	Jabatan tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama jabatan tidak boleh lebih dari 255 karakter."/" <i>Job title must not exceed 255 characters.</i> ".	✓	
TC13	<i>Admin Super</i> memilih jabatan yang ada, mengubah nama jabatan dengan nama baru yang valid, dan preferensi bahasa Indonesia/Inggris.	Data jabatan berhasil diperbarui dan sistem menampilkan pesan "Jabatan berhasil diperbarui"/" <i>Job title updated successfully</i> ".	Data jabatan berhasil diperbarui dan sistem menampilkan pesan "Jabatan berhasil diperbarui"/" <i>Job title updated successfully</i> ".	✓	
TC14	<i>Admin Super</i> mengubah nama jabatan menjadi kosong dan preferensi bahasa Indonesia/Inggris.	Jabatan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama jabatan tidak boleh kosong."/" <i>Job title name cannot be empty</i> ".	Jabatan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama jabatan tidak boleh kosong."/" <i>Job title name cannot be empty</i> ".	✓	
TC15	<i>Admin Super</i> mengubah nama jabatan menjadi nama yang sudah digunakan jabatan lain dan	Jabatan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama jabatan sudah digunakan."/" <i>Job</i>	Jabatan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama jabatan sudah digunakan."/" <i>Job title has already been taken.</i> ".	✓	

	preferensi bahasa Indonesia/Inggris.	<i>title has already been taken.</i> ".			
TC16	<i>Admin Super</i> mengubah nama jabatan menjadi lebih dari 255 karakter dan preferensi bahasa Indonesia/Inggris.	Jabatan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama jabatan tidak boleh lebih dari 255 karakter."/" <i>Job title must not exceed 255 characters.</i> ".	Jabatan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama jabatan tidak boleh lebih dari 255 karakter."/" <i>Job title must not exceed 255 characters.</i> ".	✓	
TC17	<i>Admin Super</i> memilih jabatan yang ada untuk dihapus dan preferensi bahasa Indonesia/Inggris.	Jabatan berhasil dihapus dari <i>database</i> dan sistem menampilkan pesan "Jabatan berhasil dihapus"/" <i>Job title deleted successfully</i> ".	Jabatan berhasil dihapus dari <i>database</i> dan sistem menampilkan pesan "Jabatan berhasil dihapus"/" <i>Job title deleted successfully</i> ".	✓	

Hasil analisis:

1) *Persentase test case* yang lolos = $\frac{17}{17} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{17} = 0\%$

3) *Persentase test case* yang telah diuji = 17 = 100%

4) *Persentase test case* yang belum diuji = 0 = 0%

Jumlah total *test case* dalam pengujian = 17

4. Testing Scenario Use Case Manajemen Peran Hak Akses

Pengujian ini didasari oleh *Scenario Use Case* Manajemen Peran Hak Akses (Tabel 3.16). Use case scenario mendefinisikan masuk ke dalam halaman peran hak akses, namun pengujian dilakukan secara detail untuk memvalidasi data tambah, edit, dan hapus peran hak akses. Hal ini untuk memastikan manajemen peran hak akses berjalan lancar.

Tabel 4.5 *Black Box* Testing Peran Hak Akses

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin Super</i> mengakses halaman daftar peran dengan data peran tersedia di <i>database</i> .	Sistem menampilkan semua data peran beserta tombol aksi (lihat <i>detail</i> , edit, hapus).	Sistem menampilkan semua data peran beserta tombol aksi (lihat <i>detail</i> , edit, hapus).	✓	
TC02	<i>Admin Super</i> mengklik tombol tambah peran.	Sistem memunculkan halaman <i>input</i> tambah peran dengan <i>field</i> nama peran dan daftar fitur yang bisa dipilih.	Sistem memunculkan halaman <i>input</i> tambah peran dengan <i>field</i> nama peran dan daftar fitur yang bisa dipilih.	✓	
TC03	<i>Admin Super</i> mengklik tombol mata (lihat <i>detail</i>) pada salah satu data peran.	Sistem memunculkan <i>modal detail</i> peran yang menampilkan nama peran dan daftar fitur yang terkait.	Sistem memunculkan <i>modal detail</i> peran yang menampilkan nama peran dan daftar fitur yang terkait.	✓	
TC04	<i>Admin Super</i> mengklik tombol hapus peran.	Sistem memunculkan <i>modal</i> konfirmasi hapus peran.	Sistem memunculkan <i>modal</i> konfirmasi hapus peran.	✓	
TC05	<i>Admin Super</i> mengklik tombol <i>delete</i> pada <i>modal</i> konfirmasi hapus peran.	Sistem menutup <i>modal</i> dan menampilkan notifikasi.	Sistem menutup <i>modal</i> dan menampilkan notifikasi.	✓	
TC06	<i>Admin Super</i> mengklik tombol	Sistem menutup <i>modal</i> konfirmasi tanpa adanya perubahan data.	Sistem menutup <i>modal</i> konfirmasi tanpa adanya perubahan data.	✓	

	batal pada <i>modal</i> konfirmasi.				
TC07	<i>Admin Super</i> mengklik tombol edit pada salah satu data peran.	Sistem memunculkan halaman edit data peran dengan nama peran dan fitur yang sudah terpilih sebelumnya.	Sistem memunculkan halaman edit data peran dengan nama peran dan fitur yang sudah terpilih sebelumnya.	✓	
TC08	<i>Admin Super</i> mengklik tombol <i>submit</i> pada halaman edit peran.	Sistem menutup halaman dan menampilkan pesan notifikasi.	Sistem menutup halaman dan menampilkan pesan notifikasi.	✓	
TC09	<i>Admin Super</i> mengklik tombol kembali pada halaman edit peran.	Sistem menutup halaman tanpa ada perubahan data.	Sistem menutup halaman tanpa ada perubahan data.	✓	
TC10	<i>Admin Super</i> mengisi nama peran dengan benar, memilih beberapa fitur (klik checkbox fitur), dan preferensi bahasa Indonesia/Inggris.	Peran berhasil disimpan ke <i>database</i> dengan fitur yang dipilih dan sistem menampilkan pesan "Peran berhasil dibuat"/"Role created successfully".	Peran berhasil disimpan ke <i>database</i> dengan fitur yang dipilih dan sistem menampilkan pesan "Peran berhasil dibuat"/"Role created successfully".	✓	
TC11	<i>Admin Super</i> mengisi nama peran dengan benar tanpa	Peran berhasil disimpan ke <i>database</i> tanpa fitur terkait dan sistem menampilkan pesan "Peran berhasil	Peran berhasil disimpan ke <i>database</i> tanpa fitur terkait dan sistem menampilkan pesan	✓	

	memilih fitur apapun, preferensi bahasa Indonesia/Inggris.	dibuat"/"Role created successfully".	"Peran berhasil dibuat"/"Role created successfully".		
TC12	Admin Super tidak mengisi nama peran, preferensi bahasa Indonesia/Inggris.	Peran tidak berhasil disimpan tombol diklik tanpa menghasilkan pesan.	Peran tidak berhasil disimpan tombol diklik tanpa menghasilkan pesan.	✓	
TC13	Admin Super mengisi nama peran yang sudah ada di database dan preferensi bahasa Indonesia/Inggris.	Peran tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama peran sudah digunakan."/"Role name has already been taken.".	Peran tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama peran sudah digunakan."/"Role name has already been taken.".	✓	
TC14	Admin Super mengisi nama peran lebih dari 255 karakter dan preferensi bahasa Indonesia/Inggris.	Peran tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama peran tidak boleh lebih dari 255 karakter."/"Role name must not exceed 255 characters.".	Peran tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama peran tidak boleh lebih dari 255 karakter."/"Role name must not exceed 255 characters.".	✓	
TC15	Admin Super memilih peran yang ada, mengubah nama peran dengan nama baru yang	Data peran dan fitur berhasil diperbarui dan sistem menampilkan pesan "Peran berhasil diperbarui"/"Role updated successfully".	Data peran dan fitur berhasil diperbarui dan sistem menampilkan pesan "Peran berhasil diperbarui"/"Role updated successfully".	✓	

	valid, mengubah pilihan fitur (klik/ <i>unclick checkbox</i>), dan preferensi bahasa Indonesia/Inggris.				
TC16	<i>Admin Super</i> memilih peran yang ada, mengubah nama peran saja tanpa mengubah fitur, dan preferensi bahasa Indonesia/Inggris.	Data peran berhasil diperbarui tanpa mengubah relasi fitur dan sistem menampilkan pesan "Peran berhasil diperbarui"/" <i>Role updated successfully</i> ".	Data peran berhasil diperbarui tanpa mengubah relasi fitur dan sistem menampilkan pesan "Peran berhasil diperbarui"/" <i>Role updated successfully</i> ".	✓	
TC17	<i>Admin Super</i> memilih peran yang ada, hanya mengubah pilihan fitur tanpa mengubah nama peran, dan preferensi bahasa Indonesia/Inggris.	Relasi fitur berhasil diperbarui dan sistem menampilkan pesan "Peran berhasil diperbarui"/" <i>Role updated successfully</i> ".	Relasi fitur berhasil diperbarui dan sistem menampilkan pesan "Peran berhasil diperbarui"/" <i>Role updated successfully</i> ".	✓	
TC18	<i>Admin Super</i> memilih peran yang ada dan menghapus semua pilihan fitur (<i>unclick</i>	Semua relasi fitur berhasil dihapus dari peran dan sistem menampilkan pesan "Peran berhasil diperbarui"/" <i>Role updated successfully</i> ".	Semua relasi fitur berhasil dihapus dari peran dan sistem menampilkan pesan "Peran berhasil diperbarui"/" <i>Role updated successfully</i> ".	✓	

	semua checkbox), preferensi bahasa Indonesia/Inggris.				
TC19	<i>Admin Super</i> mengubah nama peran menjadi kosong dan preferensi bahasa Indonesia/Inggris.	Peran tidak berhasil disimpan tombol diklik tanpa menghasilkan pesan.	Peran tidak berhasil disimpan tombol diklik tanpa menghasilkan pesan.	✓	
TC20	<i>Admin Super</i> mengubah nama peran menjadi nama yang sudah digunakan peran lain dan preferensi bahasa Indonesia/Inggris.	Peran tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama peran sudah digunakan."/"Role name has already been taken.".	Peran tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama peran sudah digunakan."/"Role name has already been taken.".	✓	
TC21	<i>Admin Super</i> mengubah nama peran menjadi lebih dari 255 karakter dan preferensi bahasa Indonesia/Inggris.	Peran tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama peran tidak boleh lebih dari 255 karakter."/"Role name must not exceed 255 characters.".	Peran tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama peran tidak boleh lebih dari 255 karakter."/"Role name must not exceed 255 characters.".	✓	
TC22	<i>Admin Super</i> memilih peran yang ada untuk dihapus (tidak ada user yang	Peran berhasil dihapus dari database beserta relasi fitur dan sistem menampilkan pesan "Peran berhasil	Peran berhasil dihapus dari database beserta relasi fitur dan sistem menampilkan pesan "Peran berhasil dihapus"/"Role deleted successfully".	✓	

	menggunakan peran tersebut), preferensi bahasa Indonesia/Inggris.	dihapus"/"Role deleted successfully".			
TC23	Admin Super memilih peran yang ada untuk dihapus (ada user yang menggunakan peran tersebut), preferensi bahasa Indonesia/Inggris.	Sistem mengubah peran_id user menjadi null, peran berhasil dihapus dari database, dan sistem menampilkan pesan "Peran berhasil dihapus"/"Role deleted successfully".	Peran berhasil dihapus dari database beserta relasi fitur menjadil null dan sistem menampilkan pesan "Peran berhasil dihapus"/"Role deleted successfully".	✓	

Hasil analisis:

1) Persentase test case yang lolos = $\frac{23}{23} = 100\%$

2) Persentase test case yang gagal = $\frac{0}{23} = 0\%$

3) Persentase test case yang telah diuji = 23 = 100%

4) Persentase test case yang belum diuji = 0 = 0%

Jumlah total test case dalam pengujian = 23

5. Testing Scenario Use Case Manajemen Karyawan

Pengujian ini didasari oleh Scenario Use Case Manajemen Karyawan (Tabel 3.17). Use case scenario mendefinisikan masuk ke dalam halaman karyawan, namun pengujian dilakukan secara detail untuk memvalidasi data tambah, edit, dan hapus karyawan. Hal ini untuk memastikan manajemen karyawan berjalan lancar.

Tabel 4.6 Black Box Testing Manajemen Karyawan

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G

TC01	<i>Admin Super</i> mengakses halaman daftar karyawan dengan data karyawan tersedia di <i>database</i> .	Sistem menampilkan semua data karyawan beserta relasi peran, departemen, dan jabatan dengan tombol aksi (lihat <i>detail</i> , edit, hapus).	Sistem menampilkan semua data karyawan beserta relasi peran, departemen, dan jabatan dengan tombol aksi (lihat <i>detail</i> , edit, hapus).	✓	
TC02	<i>Admin Super</i> mengklik tombol tambah karyawan.	Sistem mengarahkan ke halaman tambah karyawan dengan <i>form input</i> data karyawan.	Sistem mengarahkan ke halaman tambah karyawan dengan <i>form input</i> data karyawan.	✓	
TC03	<i>Admin Super</i> mengklik tombol mata (lihat <i>detail</i>) pada salah satu data karyawan.	Sistem memunculkan <i>modal detail</i> karyawan yang menampilkan semua informasi karyawan beserta relasi peran, departemen, dan jabatan.	Sistem memunculkan <i>modal detail</i> karyawan yang menampilkan semua informasi karyawan beserta relasi peran, departemen, dan jabatan.	✓	
TC04	<i>Admin Super</i> mengklik tombol hapus karyawan.	Sistem memunculkan <i>modal</i> konfirmasi hapus karyawan.	Sistem memunculkan <i>modal</i> konfirmasi hapus karyawan.	✓	
TC05	<i>Admin Super</i> mengklik tombol <i>delete</i> pada <i>modal</i> konfirmasi hapus karyawan.	Sistem menutup <i>modal</i> dan menampilkan notifikasi.	Sistem menutup <i>modal</i> dan menampilkan notifikasi.	✓	
TC06	<i>Admin Super</i> mengklik tombol batal pada <i>modal</i> konfirmasi.	Sistem menutup <i>modal</i> konfirmasi tanpa adanya perubahan data.	Sistem menutup <i>modal</i> konfirmasi tanpa adanya perubahan data.	✓	

TC07	<i>Admin Super</i> mengklik tombol edit pada salah satu data karyawan.	Sistem mengarahkan ke halaman edit karyawan dengan data karyawan yang sudah terisi otomatis.	Sistem mengarahkan ke halaman edit karyawan dengan data karyawan yang sudah terisi otomatis.	✓	
TC08	<i>Admin Super</i> mengisi semua <i>field</i> dengan benar dan preferensi bahasa Indonesia/Inggris.	Karyawan berhasil disimpan dengan <i>email auto-generate</i> , sistem menampilkan pesan "Data karyawan berhasil diperbarui"/" <i>Employee data updated successfully</i> ".	Karyawan berhasil disimpan dengan <i>email auto-generate</i> , sistem menampilkan pesan "Data karyawan berhasil diperbarui"/" <i>Employee data updated successfully</i> ".	✓	
TC09	<i>Admin Super</i> mengosongkan salah satu <i>field</i> atau mengosongkan semua <i>field</i> , preferensi bahasa Indonesia/Inggris.	Karyawan tidak berhasil disimpan dan sistem menampilkan pesan validasi "Harap isi semua <i>field</i> "/" <i>Please fill in all fields</i> ".	Karyawan tidak berhasil disimpan dan sistem menampilkan pesan validasi "Harap isi semua <i>field</i> "/" <i>Please fill in all fields</i> ".	✓	
TC10	<i>Admin Super</i> menambah karyawan dengan nama yang sama (duplikat nama diperbolehkan) tetapi data unik lainnya berbeda, preferensi bahasa Indonesia/Inggris.	Karyawan berhasil disimpan dengan <i>email</i> berbeda (<i>auto-increment</i> angka), sistem menampilkan pesan "Karyawan berhasil dibuat"/" <i>Employee created successfully</i> ".	Karyawan berhasil disimpan dengan <i>email</i> berbeda (<i>auto-increment</i> angka), sistem menampilkan pesan "Karyawan berhasil dibuat"/" <i>Employee created successfully</i> ".	✓	

TC11	<i>Admin Super</i> mengisi NPWP yang sudah digunakan karyawan lain (duplikat), preferensi bahasa Indonesia/Inggris.	Karyawan tidak berhasil disimpan dan sistem menampilkan pesan validasi "NPWP sudah digunakan oleh karyawan lain."/"NPWP has already been taken".	Karyawan tidak berhasil disimpan dan sistem menampilkan pesan validasi "NPWP sudah digunakan oleh karyawan lain."/"NPWP has already been taken".	✓	
TC12	<i>Admin Super</i> mengisi NPWP dengan format tidak valid (mengandung huruf atau karakter selain angka, titik, strip, garis miring), preferensi bahasa Indonesia/Inggris.	Karyawan tidak berhasil disimpan dan sistem menampilkan pesan validasi "NPWP hanya boleh berisi angka, titik, strip, atau garis miring."/"NPWP may only contain numbers, dots, dashes, or slashes".	Karyawan tidak berhasil disimpan dan sistem menampilkan pesan validasi "NPWP hanya boleh berisi angka, titik, strip, atau garis miring."/"NPWP may only contain numbers, dots, dashes, or slashes".	✓	
TC13	<i>Admin Super</i> mengisi NPWP dengan format valid (angka, titik, strip, garis miring), preferensi bahasa Indonesia/Inggris.	Karyawan berhasil disimpan dengan NPWP yang valid, sistem menampilkan pesan "Karyawan berhasil dibuat"/"Employee created successfully".	Karyawan berhasil disimpan dengan NPWP yang valid, sistem menampilkan pesan "Karyawan berhasil dibuat"/"Employee created successfully".	✓	
TC14	<i>Admin Super</i> mengisi BPJS Kesehatan yang	Karyawan tidak berhasil disimpan dan sistem menampilkan pesan	Karyawan tidak berhasil disimpan dan sistem menampilkan pesan validasi	✓	

	sudah digunakan karyawan lain (duplikat), preferensi bahasa Indonesia/Inggris.	validasi "BPJS Kesehatan sudah digunakan oleh karyawan lain."/"Health BPJS has already been taken.".	"BPJS Kesehatan sudah digunakan oleh karyawan lain."/"Health BPJS has already been taken.".		
TC15	Admin Super mengisi BPJS Kesehatan dengan format tidak valid (mengandung huruf atau karakter selain angka), preferensi bahasa Indonesia/Inggris.	Karyawan tidak berhasil disimpan dan sistem menampilkan pesan validasi "BPJS Kesehatan hanya boleh berisi angka."/"Health BPJS must contain only numbers.".	Karyawan tidak berhasil disimpan dan sistem menampilkan pesan validasi "BPJS Kesehatan hanya boleh berisi angka."/"Health BPJS must contain only numbers.".	✓	
TC16	Admin Super mengisi BPJS Kesehatan dengan format valid (hanya angka), preferensi bahasa Indonesia/Inggris.	Karyawan berhasil disimpan dengan BPJS Kesehatan yang valid, sistem menampilkan pesan "Karyawan berhasil dibuat"/"Employee created successfully".	Karyawan berhasil disimpan dengan BPJS Kesehatan yang valid, sistem menampilkan pesan "Karyawan berhasil dibuat"/"Employee created successfully".	✓	
TC17	Admin Super mengisi BPJS Ketenagakerjaan yang sudah digunakan karyawan lain (duplikat),	Karyawan tidak berhasil disimpan dan sistem menampilkan pesan validasi "BPJS Ketenagakerjaan sudah digunakan oleh karyawan	Karyawan tidak berhasil disimpan dan sistem menampilkan pesan validasi "BPJS Ketenagakerjaan sudah digunakan oleh karyawan lain."/"Employment BPJS has already been taken.".	✓	

	preferensi bahasa Indonesia/Inggris.	lain."/ <i>"Employment BPJS has already been taken."</i> .			
TC18	<i>Admin Super</i> mengisi BPJS Ketenagakerjaan dengan format tidak valid (mengandung huruf atau karakter selain angka), preferensi bahasa Indonesia/Inggris.	Karyawan tidak berhasil disimpan dan sistem menampilkan pesan validasi "BPJS Ketenagakerjaan hanya boleh berisi angka."/ <i>"Employment BPJS must contain only numbers."</i> .	Karyawan tidak berhasil disimpan dan sistem menampilkan pesan validasi "BPJS Ketenagakerjaan hanya boleh berisi angka."/ <i>"Employment BPJS must contain only numbers."</i> .	✓	
TC19	<i>Admin Super</i> mengisi BPJS Ketenagakerjaan dengan format valid (hanya angka), preferensi bahasa Indonesia/Inggris.	Karyawan berhasil disimpan dengan BPJS Ketenagakerjaan yang valid, sistem menampilkan pesan "Karyawan berhasil dibuat"/ <i>"Employee created successfully"</i> .	Karyawan berhasil disimpan dengan BPJS Ketenagakerjaan yang valid, sistem menampilkan pesan "Karyawan berhasil dibuat"/ <i>"Employee created successfully"</i> .	✓	
TC20	<i>Admin Super</i> memilih karyawan yang ada dan mengubah satu atau beberapa <i>field</i> dengan data valid, preferensi	Data karyawan berhasil diperbarui dan sistem menampilkan pesan "User berhasil diperbarui"/ <i>"User updated successfully"</i> .	Data karyawan berhasil diperbarui dan sistem menampilkan pesan "User berhasil diperbarui"/ <i>"User updated successfully"</i> .	✓	

	bahasa Indonesia/Inggris.				
TC21	<i>Admin Super</i> memilih karyawan yang ada dan hanya mengubah <i>password</i> dengan <i>password</i> minimal 6 karakter, preferensi bahasa Indonesia/Inggris.	<i>Password</i> karyawan berhasil diperbarui (<i>encrypted</i>) dan sistem menampilkan pesan " <i>User</i> berhasil diperbarui"/" <i>User</i> <i>updated successfully</i> ".	<i>Password</i> karyawan berhasil diperbarui (<i>encrypted</i>) dan sistem menampilkan pesan " <i>User</i> berhasil diperbarui"/" <i>User updated</i> <i>successfully</i> ".	✓	
TC22	<i>Admin Super</i> memilih karyawan yang ada dan mengubah data tanpa mengubah <i>password</i> (<i>password</i> kosong), preferensi bahasa Indonesia/Inggris.	Data karyawan berhasil diperbarui tanpa mengubah <i>password</i> dan sistem menampilkan pesan " <i>User</i> berhasil diperbarui"/" <i>User</i> <i>updated successfully</i> ".	Data karyawan berhasil diperbarui tanpa mengubah <i>password</i> dan sistem menampilkan pesan " <i>User</i> berhasil diperbarui"/" <i>User</i> <i>updated successfully</i> ".	✓	
TC23	<i>Admin Super</i> memilih karyawan yang ada dan mengubah NPWP menjadi nilai	Data karyawan berhasil diperbarui dengan NPWP baru dan sistem menampilkan pesan " <i>User</i> berhasil diperbarui"/" <i>User</i> <i>updated successfully</i> ".	Data karyawan berhasil diperbarui dengan NPWP baru dan sistem menampilkan pesan " <i>User</i> berhasil diperbarui"/" <i>User updated</i> <i>successfully</i> ".	✓	

	yang belum digunakan karyawan lain, preferensi bahasa Indonesia/Inggris.				
TC24	<i>Admin Super</i> memilih karyawan yang ada dan mengubah BPJS Kesehatan menjadi nilai yang belum digunakan karyawan lain, preferensi bahasa Indonesia/Inggris.	Data karyawan berhasil diperbarui dengan BPJS Kesehatan baru dan sistem menampilkan pesan " <i>User berhasil diperbarui</i> "/" <i>User updated successfully</i> ".	Data karyawan berhasil diperbarui dengan BPJS Kesehatan baru dan sistem menampilkan pesan " <i>User berhasil diperbarui</i> "/" <i>User updated successfully</i> ".	✓	
TC25	<i>Admin Super</i> memilih karyawan yang ada dan mengubah BPJS Ketenagakerjaan menjadi nilai yang belum digunakan karyawan lain, preferensi bahasa Indonesia/Inggris.	Data karyawan berhasil diperbarui dengan BPJS Ketenagakerjaan baru dan sistem menampilkan pesan " <i>User berhasil diperbarui</i> "/" <i>User updated successfully</i> ".	Data karyawan berhasil diperbarui dengan BPJS Ketenagakerjaan baru dan sistem menampilkan pesan " <i>User berhasil diperbarui</i> "/" <i>User updated successfully</i> ".	✓	

TC26	<i>Admin Super</i> mengubah nama menjadi lebih dari 255 karakter, preferensi bahasa Indonesia/Inggris.	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama tidak boleh lebih dari 255 karakter."/"Name must not exceed 255 characters.".	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama tidak boleh lebih dari 255 karakter."/"Name must not exceed 255 characters.".	✓	
TC27	<i>Admin Super</i> mengubah gaji per hari dengan nilai negatif, preferensi bahasa Indonesia/Inggris.	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Gaji per hari tidak boleh bernilai negatif."/"Daily salary cannot be negative.".	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Gaji per hari tidak boleh bernilai negatif."/"Daily salary cannot be negative.".	✓	
TC28	<i>Admin Super</i> mengubah gaji per hari dengan karakter non-numerik, preferensi bahasa Indonesia/Inggris.	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Gaji per hari harus berupa angka."/"Daily salary must be a number.".	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Gaji per hari harus berupa angka."/"Daily salary must be a number.".	✓	
TC29	<i>Admin Super</i> mengubah <i>password</i> menjadi kurang dari 6 karakter, preferensi bahasa Indonesia/Inggris.	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Password minimal terdiri dari 6 karakter."/"Password must be at least 6 characters.".	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Password minimal terdiri dari 6 karakter."/"Password must be at least 6 characters.".	✓	
TC30	<i>Admin Super</i> mengubah NPWP	Karyawan tidak berhasil diperbarui dan sistem	Karyawan tidak berhasil diperbarui dan sistem	✓	

	menjadi NPWP yang sudah digunakan karyawan lain, preferensi bahasa Indonesia/Inggris.	menampilkan pesan validasi "NPWP sudah digunakan oleh karyawan lain."/"NPWP has already been taken.".	menampilkan pesan validasi "NPWP sudah digunakan oleh karyawan lain."/"NPWP has already been taken.".		
TC31	<i>Admin Super</i> mengubah NPWP dengan format tidak valid (mengandung karakter selain angka, titik, strip, garis miring), preferensi bahasa Indonesia/Inggris.	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "NPWP hanya boleh berisi angka, titik, strip, atau garis miring."/"NPWP may only contain numbers, dots, dashes, or slashes.".	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "NPWP hanya boleh berisi angka, titik, strip, atau garis miring."/"NPWP may only contain numbers, dots, dashes, or slashes.".	✓	
TC32	<i>Admin Super</i> mengubah BPJS Kesehatan menjadi BPJS Kesehatan yang sudah digunakan karyawan lain, preferensi bahasa Indonesia/Inggris.	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "BPJS Kesehatan sudah digunakan oleh karyawan lain."/"Health BPJS has already been taken.".	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "BPJS Kesehatan sudah digunakan oleh karyawan lain."/"Health BPJS has already been taken.".	✓	
TC33	<i>Admin Super</i> mengubah BPJS Kesehatan dengan format tidak valid (mengandung	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "BPJS Kesehatan hanya boleh berisi	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "BPJS Kesehatan hanya boleh	✓	

	karakter selain angka), preferensi bahasa Indonesia/Inggris.	angka."/Health BPJS must contain only numbers."	berisi angka."/Health BPJS must contain only numbers."		
TC34	Admin Super mengubah BPJS Ketenagakerjaan menjadi BPJS Ketenagakerjaan yang sudah digunakan karyawan lain, preferensi bahasa Indonesia/Inggris.	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "BPJS Ketenagakerjaan sudah digunakan oleh karyawan lain."/Employment BPJS has already been taken."	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "BPJS Ketenagakerjaan sudah digunakan oleh karyawan lain."/Employment BPJS has already been taken."	✓	
TC35	Admin Super mengubah BPJS Ketenagakerjaan dengan format tidak valid (mengandung karakter selain angka), preferensi bahasa Indonesia/Inggris.	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "BPJS Ketenagakerjaan hanya boleh berisi angka."/Employment BPJS must contain only numbers."	Karyawan tidak berhasil diperbarui dan sistem menampilkan pesan validasi "BPJS Ketenagakerjaan hanya boleh berisi angka."/Employment BPJS must contain only numbers."	✓	
TC36	Admin Super memilih karyawan yang ada untuk dihapus,	Karyawan berhasil dihapus dari database dan sistem menampilkan pesan "User berhasil dihapus"/User deleted successfully".	Karyawan berhasil dihapus dari database dan sistem menampilkan pesan "User berhasil dihapus"/User deleted successfully".	✓	

	preferensi bahasa Indonesia/Inggris.				
--	---	--	--	--	--

Hasil analisi:

1) *Persentase test case* yang lolos = $\frac{36}{36} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{36} = 0\%$

3) *Persentase test case* yang telah diuji = $36 = 100\%$

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

Jumlah total *test case* dalam pengujian = 36

6. Testing Scenario Use Case Log Aktivitas

Pengujian ini didasari oleh *Scenario Use Case Log Aktivitas* (Tabel 3.20).

Meskipun use case scenario mendefinisikan skenario normal yaitu Admin Super dapat melihat log aktivitas user, pengujian dilakukan secara detail untuk memvalidasi berbagai jenis log aktivitas yang dapat ditampilkan sistem, meliputi log login, logout, create (tambah data), update (edit data), dan delete (hapus data). Hal ini untuk memastikan sistem dapat mencatat dan menampilkan seluruh aktivitas user dengan lengkap.

Tabel 4.7 *Black Box Testing Log Aktivitas*

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	Admin Super mengakses halaman log aktivitas dengan data log aktivitas tersedia di database.	Sistem menampilkan data log aktivitas beserta nama user, waktu, dan jenis aktivitas.	Sistem menampilkan data log aktivitas beserta nama user, waktu, dan jenis aktivitas.	✓	
TC02	Admin Super melihat log aktivitas yang	Sistem menampilkan log login dengan informasi user dan waktu login.	Sistem menampilkan log login dengan informasi user dan waktu login.	✓	

	mencatat aksi <i>login user detail</i> <i>log aktivitas login</i> <i>user tertentu.</i>				
TC03	<i>Admin Super</i> melihat <i>log</i> aktivitas yang mencatat aksi <i>logout user.</i>	Sistem menampilkan <i>log</i> <i>logout</i> dengan informasi <i>user</i> dan waktu <i>logout</i> .	Sistem menampilkan <i>log</i> <i>logout</i> dengan informasi <i>user</i> dan waktu <i>logout</i> .	✓	
TC04	<i>Admin Super</i> melihat <i>log</i> aktivitas yang mencatat aksi tambah data.	Sistem menampilkan <i>log</i> <i>create</i> dengan informasi <i>user</i> , waktu, dan data yang ditambahkan.	Sistem menampilkan <i>log</i> <i>create</i> dengan informasi <i>user</i> , waktu, dan data yang ditambahkan.	✓	
TC05	<i>Admin Super</i> melihat <i>log</i> aktivitas yang mencatat aksi edit data.	Sistem menampilkan <i>log</i> <i>update</i> dengan informasi <i>user</i> , waktu, dan data yang diubah.	Sistem menampilkan <i>log</i> <i>update</i> dengan informasi <i>user</i> , waktu, dan data yang diubah.	✓	
TC06	<i>Admin Super</i> melihat <i>log</i> aktivitas yang mencatat aksi hapus data.	Sistem menampilkan <i>log</i> <i>delete</i> dengan informasi <i>user</i> , waktu, dan data yang dihapus.	Sistem menampilkan <i>log delete</i> dengan informasi <i>user</i> , waktu, dan data yang dihapus.	✓	

Hasil analisi:

1) *Persentase test case* yang lolos = $\frac{6}{6} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{6} = 0\%$

3) *Persentase test case* yang telah diuji = 6 = 100%

4) *Persentase test case* yang belum diuji = 0 = 0%

Jumlah total *test case* dalam pengujian = 6

7. Testing *Scenario Use Case* Manajemen Kantor

Pengujian ini didasari oleh *Scenario Use Case* Manajemen Kantor (Tabel 3.18). Use case scenario mendefinisikan masuk ke dalam halaman kantor, namun pengujian dilakukan secara detail untuk memvalidasi data tambah, dan edit kantor. Hal ini untuk memastikan manajemen kantor berjalan lancar.

Tabel 4.8 *Black Box* Testing Manajemen Kantor

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin Super</i> mengakses halaman data kantor dengan data kantor sudah tersedia di <i>database</i> .	Sistem menampilkan data kantor (jam masuk, jam keluar, minimal keterlambatan, koordinat lat/lng, radius meter).	Sistem menampilkan data kantor (jam masuk, jam keluar, minimal keterlambatan, koordinat lat/lng, radius meter).	✓	
TC02	<i>Admin Super</i> mengakses halaman data kantor tanpa ada data kantor di <i>database</i> (pertama kali).	Sistem menampilkan halaman kosong/data <i>null</i> dengan pesan.	Sistem menampilkan halaman kosong/data <i>null</i> dengan pesan.	✓	
TC03	<i>Admin Super</i> mengisi semua <i>field</i> dengan benar (jam masuk, jam keluar, minimal	Data kantor berhasil disimpan ke <i>database</i> dan sistem menampilkan pesan "Data kantor berhasil ditambahkan".	Data kantor berhasil disimpan ke <i>database</i> dan sistem menampilkan pesan "Data kantor berhasil ditambahkan".	✓	

	keterlambatan, lat, lng, radius meter) saat belum ada data kantor.				
TC04	<i>Admin Super</i> tidak mengisi jam masuk (<i>field</i> kosong) preferensi bahasa Indonesia/Inggris.	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Jam masuk wajib diisi."/" <i>Check-in time is required.</i> ".	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Jam masuk wajib diisi."/" <i>Check-in time is required.</i> ".	✓	
TC05	<i>Admin Super</i> tidak mengisi jam keluar (<i>field</i> kosong).	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Jam keluar wajib diisi."/" <i>Check-out time is required.</i> ".	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Jam keluar wajib diisi."/" <i>Check-out time is required.</i> ".	✓	
TC06	<i>Admin Super</i> tidak mengisi minimal keterlambatan (<i>field</i> kosong).	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Wajib diisi".	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Wajib diisi".	✓	
TC07	<i>Admin Super</i> tidak mengisi <i>latitude</i> (<i>field</i> kosong).	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Wajib diisi".	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Wajib diisi".	✓	
TC08	<i>Admin Super</i> mengisi <i>latitude</i> dengan karakter non-numerik.	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Format tidak valid".	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Format tidak valid".	✓	

TC09	<i>Admin Super</i> tidak mengisi <i>longitude</i> (<i>field</i> kosong).	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi “Wajib diisi”.	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi “Wajib diisi”.	✓	
TC10	<i>Admin Super</i> mengisi <i>longitude</i> dengan karakter non-numerik.	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi “Format tidak valid”.	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi “Format tidak valid”.	✓	
TC11	<i>Admin Super</i> tidak mengisi radius meter (<i>field</i> kosong).	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi “Wajib diisi”.	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi “Wajib diisi”.	✓	
TC12	<i>Admin Super</i> mengisi radius meter dengan karakter non-integer.	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi “Harus berupa angka positif”.	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi “Harus berupa angka positif”.	✓	
TC13	<i>Admin Super</i> mengubah data kantor yang sudah ada dengan data valid.	Data kantor berhasil diperbarui dan sistem menampilkan pesan "Data kantor berhasil diperbarui".	Data kantor berhasil diperbarui dan sistem menampilkan pesan "Data kantor berhasil diperbarui".	✓	
TC14	<i>Admin Super</i> mengubah minimal keterlambatan dengan nilai valid.	Data kantor berhasil diperbarui dengan minimal keterlambatan baru dan sistem menampilkan pesan "Data kantor berhasil diperbarui".	Data kantor berhasil diperbarui dengan minimal keterlambatan baru dan sistem menampilkan pesan "Data kantor berhasil diperbarui".	✓	

TC15	<i>Admin Super</i> mengubah koordinat <i>latitude</i> dan <i>longitude</i> dengan nilai valid.	Data kantor berhasil diperbarui dengan koordinat baru dan sistem menampilkan pesan "Data kantor berhasil diperbarui".	Data kantor berhasil diperbarui dengan koordinat baru dan sistem menampilkan pesan "Data kantor berhasil diperbarui".	✓	
TC16	<i>Admin Super</i> mengubah radius meter dengan nilai valid.	Data kantor berhasil diperbarui dengan radius meter baru dan sistem menampilkan pesan "Data kantor berhasil diperbarui".	Data kantor berhasil diperbarui dengan radius meter baru dan sistem menampilkan pesan "Data kantor berhasil diperbarui".	✓	
TC17	<i>Admin Super</i> mengubah jam masuk menjadi kosong saat <i>update</i> .	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Jam masuk wajib diisi."/" <i>Check-in time is required.</i> ".	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Jam masuk wajib diisi."/" <i>Check-in time is required.</i> ".	✓	
TC18	<i>Admin Super</i> mengubah jam keluar menjadi kosong saat <i>update</i> .	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Jam keluar wajib diisi."/" <i>Check-out time is required.</i> ".	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Jam keluar wajib diisi."/" <i>Check-out time is required.</i> ".	✓	
TC19	<i>Admin Super</i> mengubah <i>latitude</i> dengan karakter non-numerik saat <i>update</i> .	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Format tidak valid"	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi "Format tidak valid".	✓	

TC20	<i>Admin Super</i> mengubah <i>longitude</i> dengan karakter non- numerik saat <i>update.</i>	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi “Format tidak valid”	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi “Format tidak valid”.	✓	
TC21	<i>Admin Super</i> mengubah radius meter dengan karakter non- <i>integer</i> saat <i>update.</i>	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi “Harus berupa angka positif”.	Data kantor tidak berhasil disimpan dan sistem menampilkan pesan validasi “Harus berupa angka positif”.	✓	

Hasil analisis:

1) *Persentase test case* yang lolos = $\frac{21}{21} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{21} = 0\%$

3) *Persentase test case* yang telah diuji = $21 = 100\%$

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

Jumlah total *test case* dalam pengujian = 21

8. Testing Scenario Use Case Mengelola Reset

Pengujian ini didasari oleh *Scenario Use Case Mengelola Reset* (Tabel 3.21). Use case scenario mendefinisikan masuk ke dalam halaman reset, namun pengujian dilakukan secara detail untuk memvalidasi data hapus massal. Hal ini untuk memastikan manajemen kantor berjalan lancar.

Tabel 4.9 *Black Box Testing Mengelola Reset*

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin Super</i> mengakses	Sistem menampilkan 6 pilihan reset massal data	Sistem menampilkan 6 pilihan reset massal data sesuai yang diharapkan.	✓	

	halaman daftar reset data	yaitu <i>log</i> aktivitas, absensi, cuti, lembur, gaji, dan tugas			
TC02	<i>Admin Super</i> mengklik tombol reset pada salah satu fitur.	Sistem memunculkan <i>dropdown</i> berisi daftar data perbulan yang tersedia untuk fitur terkait.	<i>Dropdown</i> muncul menampilkan data perbulan sesuai fitur yang dipilih.	✓	
TC03	<i>Admin Super</i> memilih bulan dari <i>dropdown</i> dan menekan tombol reset.	Sistem menampilkan <i>modal</i> konfirmasi dengan inputan verifikasi.	Sistem menampilkan <i>modal</i> konfirmasi dengan inputan verifikasi.	✓	
TC04	<i>Admin Super</i> pada <i>modal</i> konfirmasi memasukkan inputan sesuai instruksi dan menekan tombol konfirmasi.	Tombol konfirmasi berubah warna menjadi merah dan dapat diklik. Setelah diklik, <i>modal</i> tertutup, data terhapus, dan muncul notifikasi sukses.	Tombol dapat di klik, modal tertutup, data terhapus secara massal, notifikasi sukses muncul.	✓	
TC05	<i>Admin Super</i> pada <i>modal</i> konfirmasi memasukkan inputan tidak sesuai instruksi dan menekan tombol konfirmasi.	Tombol konfirmasi berwarna abu-abu dan tidak dapat diklik, tidak ada perubahan data.	Tombol konfirmasi tidak dapat diklik berwarna abu-abu dan tidak ada perubahan data.	✓	
TC06	<i>Admin Super</i> pada <i>modal</i> konfirmasi menekan tombol batal.	Modal konfirmasi tertutup dan kembali ke <i>dropdown</i> data perbulan tanpa perubahan data.	Modal tertutup dan kembali ke <i>dropdown</i> tanpa perubahan data.	✓	

TC07	<i>Admin Super</i> menghapus data absensi pada bulan tertentu.	Sistem menghapus semua data absensi pada bulan tersebut beserta file <i>video user</i> dari <i>storage</i> , lalu menampilkan notifikasi "Sebanyak X absensi berhasil dihapus (beserta videonya)".	Data absensi dan file <i>video</i> terhapus dari <i>database</i> dan <i>storage</i> , notifikasi sukses muncul.	✓	
TC08	<i>Admin Super</i> menghapus data tugas pada bulan tertentu.	Sistem menghapus semua data tugas pada bulan tersebut beserta file lampiran dari <i>storage</i> , lalu menampilkan notifikasi "Sebanyak X data tugas + semua file lampiran berhasil dihapus".	Data tugas dan file lampiran terhapus dari <i>database</i> dan <i>storage</i> , notifikasi sukses muncul.	✓	
TC09	<i>Admin Super</i> menghapus data cuti pada bulan tertentu.	Sistem menghapus semua data cuti pada bulan tersebut dan menampilkan notifikasi "Sebanyak X data cuti berhasil dihapus".	Data cuti terhapus dari <i>database</i> , notifikasi sukses muncul.	✓	
TC10	<i>Admin Super</i> menghapus data lembur pada bulan tertentu.	Sistem menghapus semua data lembur pada bulan tersebut dan menampilkan notifikasi "Sebanyak X data lembur berhasil dihapus".	Data lembur terhapus dari <i>database</i> , notifikasi sukses muncul.	✓	
TC11	<i>Admin Super</i> menghapus data gaji pada bulan tertentu.	Sistem menghapus semua data gaji pada bulan tersebut dan menampilkan	Data gaji terhapus dari <i>database</i> , notifikasi sukses muncul.	✓	

		notifikasi "Sebanyak X data gaji berhasil dihapus".			
TC12	<i>Admin Super</i> menghapus data log aktivitas pada bulan tertentu.	Sistem menghapus semua log aktivitas pada bulan tersebut dan menampilkan notifikasi "Sebanyak X log aktivitas berhasil dihapus".	Sistem menghapus semua log aktivitas pada bulan tersebut dan menampilkan notifikasi "Sebanyak X log aktivitas berhasil dihapus".	✓	

Hasil analisis:

1) *Persentase test case* yang lolos = $\frac{12}{12} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{12} = 0\%$

3) *Persentase test case* yang telah diuji = $12 = 100\%$

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

Jumlah total *test case* dalam pengujian = 12

9. Testing Scenario Use Case Manajemen Potongan Gaji

Pengujian ini didasari oleh *Scenario Use Case* Manajemen Potongan Gaji (Tabel 3.19). Use case scenario mendefinisikan masuk ke dalam halaman potongan gaji, namun pengujian dilakukan secara detail untuk memvalidasi tambah, edit, dan hapus potongan. Hal ini untuk memastikan manajemen potongan gaji berjalan lancar.

Tabel 4.10 *Black Box* Testing Manajemen Potongan Gaji

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin Super</i> mengakses halaman daftar potongan gaji dengan data potongan gaji	Sistem menampilkan semua data potongan gaji.	Sistem menampilkan semua data potongan gaji.	✓	

	tersedia di <i>database</i> .				
TC02	<i>Admin Super</i> mengklik tombol tambah potongan gaji.	Sistem memunculkan halaman <i>input</i> tambah potongan gaji dengan <i>field</i> nama potongan dan <i>Persentase</i> .	Sistem memunculkan halaman <i>input</i> tambah potongan gaji dengan <i>field</i> nama potongan dan <i>Persentase</i> .	✓	
TC03	<i>Admin Super</i> mengklik tombol hapus potongan gaji.	Sistem memunculkan <i>modal</i> konfirmasi hapus potongan gaji.	Sistem memunculkan <i>modal</i> konfirmasi hapus potongan gaji.	✓	
TC04	<i>Admin Super</i> mengklik tombol <i>delete</i> pada <i>modal</i> konfirmasi hapus potongan gaji.	Sistem menutup <i>modal</i> dan menampilkan notifikasi.	Sistem menutup <i>modal</i> dan menampilkan notifikasi.	✓	
TC05	<i>Admin Super</i> mengklik tombol batal pada <i>modal</i> konfirmasi.	Sistem menutup <i>modal</i> konfirmasi tanpa adanya perubahan data.	Sistem menutup <i>modal</i> konfirmasi tanpa adanya perubahan data.	✓	
TC06	<i>Admin Super</i> mengklik tombol edit pada salah satu data potongan gaji.	Sistem memunculkan halaman edit data potongan gaji dengan data yang sudah terisi otomatis.	Sistem memunculkan halaman edit data potongan gaji dengan data yang sudah terisi otomatis.	✓	

TC07	<i>Admin Super</i> mengklik tombol <i>submit</i> pada halaman edit potongan gaji.	Sistem menutup halaman dan menampilkan pesan notifikasi.	Sistem menutup halaman dan menampilkan pesan notifikasi.	✓	
TC08	<i>Admin Super</i> mengklik tombol kembali pada halaman edit potongan gaji.	Sistem menutup halaman konfirmasi tanpa ada perubahan data.	Sistem menutup halaman konfirmasi tanpa ada perubahan data.	✓	
TC09	<i>Admin Super</i> mengisi nama potongan dengan benar dan <i>Persentase</i> valid (0-100), preferensi bahasa Indonesia/Inggris.	Potongan gaji berhasil disimpan ke <i>database</i> dan sistem menampilkan pesan "Potongan berhasil ditambahkan"/" <i>Deduction added successfully</i> ".	Potongan gaji berhasil disimpan ke <i>database</i> dan sistem menampilkan pesan "Potongan berhasil ditambahkan"/" <i>Deduction added successfully</i> ".	✓	
TC10	<i>Admin Super</i> mengisi nama potongan dengan benar dan <i>Persentase</i> desimal valid (contoh: 12.5), preferensi bahasa Indonesia/Inggris.	Potongan gaji berhasil disimpan dengan <i>Persentase</i> desimal dan sistem menampilkan pesan "Potongan berhasil ditambahkan"/" <i>Deduction added successfully</i> ".	Potongan gaji berhasil disimpan dengan <i>Persentase</i> desimal dan sistem menampilkan pesan "Potongan berhasil ditambahkan"/" <i>Deduction added successfully</i> ".	✓	

TC11	<i>Admin Super</i> tidak mengisi nama potongan (<i>field</i> kosong), preferensi bahasa Indonesia/Inggris.	Potongan gaji tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama Potongan harus diisi"/" <i>Deduction name must be filled</i> ".	Potongan gaji tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama Potongan harus diisi"/" <i>Deduction name must be filled</i> ".	✓	
TC12	<i>Admin Super</i> mengisi nama potongan yang sudah ada di <i>database</i> (duplikat), preferensi bahasa Indonesia/Inggris.	Potongan gaji tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama potongan sudah digunakan."/ <i>"Deduction name already exists."</i> .	Potongan gaji tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama potongan sudah digunakan."/ <i>"Deduction name already exists."</i> .	✓	
TC13	<i>Admin Super</i> mengisi nama potongan lebih dari 255 karakter, preferensi bahasa Indonesia/Inggris.	Potongan gaji tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama potongan tidak boleh lebih dari 255 karakter."/ <i>"Deduction name must not exceed 255 characters."</i> .	Potongan gaji tidak berhasil disimpan dan sistem menampilkan pesan validasi "Nama potongan tidak boleh lebih dari 255 karakter."/ <i>"Deduction name must not exceed 255 characters."</i> .	✓	
TC14	<i>Admin Super</i> tidak mengisi <i>Persentase</i> (<i>field</i> kosong), preferensi bahasa Indonesia/Inggris.	Potongan gaji tidak berhasil disimpan dan sistem menampilkan pesan validasi "Jumlah potongan harus berupa angka saja"/" <i>Deduction amount must be a number</i> ".	Potongan gaji tidak berhasil disimpan dan sistem menampilkan pesan validasi "Jumlah potongan harus berupa angka saja"/" <i>Deduction amount must be a number</i> ".	✓	

TC15	<i>Admin Super</i> mengisi <i>Persentase</i> dengan nilai negatif (kurang dari 0), preferensi bahasa Indonesia/Inggris.	Potongan gaji tidak berhasil disimpan dan sistem menampilkan pesan validasi " <i>Persentase</i> minimal 0."/" <i>Percentage cannot be less than 0.</i> ".	Potongan gaji tidak berhasil disimpan dan sistem menampilkan pesan validasi " <i>Persentase</i> minimal 0."/" <i>Percentage cannot be less than 0.</i> ".	✓	
TC16	<i>Admin Super</i> mengisi <i>Persentase</i> dengan nilai lebih dari 100, preferensi bahasa Indonesia/Inggris.	Potongan gaji tidak berhasil disimpan dan sistem menampilkan pesan validasi " <i>Persentase</i> maksimal 100."/" <i>Percentage cannot be greater than 100.</i> ".	Potongan gaji tidak berhasil disimpan dan sistem menampilkan pesan validasi " <i>Persentase</i> maksimal 100."/" <i>Percentage cannot be greater than 100.</i> ".	✓	
TC17	<i>Admin Super</i> mengisi <i>Persentase</i> dengan karakter non-numerik (huruf atau karakter khusus), preferensi bahasa Indonesia/Inggris.	Potongan gaji tidak berhasil disimpan dan sistem menampilkan pesan validasi "Jumlah potongan harus berupa angka saja"/" <i>Deduction amount must be a number.</i> ".	Potongan gaji tidak berhasil disimpan dan sistem menampilkan pesan validasi "Jumlah potongan harus berupa angka saja"/" <i>Deduction amount must be a number.</i> ".	✓	
TC18	<i>Admin Super</i> memilih potongan gaji yang ada, mengubah nama potongan dengan	Data potongan gaji berhasil diperbarui dan sistem menampilkan pesan "Potongan berhasil	Data potongan gaji berhasil diperbarui dan sistem menampilkan pesan "Potongan berhasil diperbarui"/" <i>Deduction updated successfully.</i> ".	✓	

	nama baru yang valid, preferensi bahasa Indonesia/Inggris.	diperbarui"/" <i>Deduction updated successfully</i> ".			
TC19	<i>Admin Super</i> memilih potongan gaji yang ada dan mengubah <i>Persentase</i> dengan nilai valid (0-100), preferensi bahasa Indonesia/Inggris.	Data potongan gaji berhasil diperbarui dengan <i>Persentase</i> baru dan sistem menampilkan pesan "Potongan berhasil diperbarui"/" <i>Deduction updated successfully</i> ".	Data potongan gaji berhasil diperbarui dengan <i>Persentase</i> baru dan sistem menampilkan pesan "Potongan berhasil diperbarui"/" <i>Deduction updated successfully</i> ".	✓	
TC20	<i>Admin Super</i> memilih potongan gaji yang ada dan mengubah nama serta <i>Persentase</i> sekaligus, preferensi bahasa Indonesia/Inggris.	Data potongan gaji berhasil diperbarui dan sistem menampilkan pesan "Potongan berhasil diperbarui"/" <i>Deduction updated successfully</i> ".	Data potongan gaji berhasil diperbarui dan sistem menampilkan pesan "Potongan berhasil diperbarui"/" <i>Deduction updated successfully</i> ".	✓	
TC21	<i>Admin Super</i> memilih potongan gaji yang ada dan mengubah <i>Persentase</i> menjadi 0 (nilai minimum),	Data potongan gaji berhasil diperbarui dengan <i>Persentase</i> 0 dan sistem menampilkan pesan "Potongan berhasil diperbarui"/" <i>Deduction updated successfully</i> ".	Data potongan gaji berhasil diperbarui dengan <i>Persentase</i> 0 dan sistem menampilkan pesan "Potongan berhasil diperbarui"/" <i>Deduction updated successfully</i> ".	✓	

	preferensi bahasa Indonesia/Inggris.				
TC22	<i>Admin Super</i> memilih potongan gaji yang ada dan mengubah <i>Persentase</i> menjadi 100 (nilai maksimum), preferensi bahasa Indonesia/Inggris.	Data potongan gaji berhasil diperbarui dengan <i>Persentase</i> 100 dan sistem menampilkan pesan "Potongan berhasil diperbarui"/" <i>Deduction updated successfully</i> ".	Data potongan gaji berhasil diperbarui dengan <i>Persentase</i> 100 dan sistem menampilkan pesan "Potongan berhasil diperbarui"/" <i>Deduction updated successfully</i> ".	✓	
TC23	<i>Admin Super</i> memilih potongan gaji yang ada dan mengubah <i>Persentase</i> menjadi nilai desimal valid, preferensi bahasa Indonesia/Inggris.	Data potongan gaji berhasil diperbarui dengan <i>Persentase</i> desimal dan sistem menampilkan pesan "Potongan berhasil diperbarui"/" <i>Deduction updated successfully</i> ".	Data potongan gaji berhasil diperbarui dengan <i>Persentase</i> desimal dan sistem menampilkan pesan "Potongan berhasil diperbarui"/" <i>Deduction updated successfully</i> ".	✓	
TC24	<i>Admin Super</i> mengubah nama potongan menjadi kosong, preferensi bahasa Indonesia/Inggris.	Potongan gaji tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Jumlah potongan harus berupa angka saja"/" <i>Deduction amount must be a number</i> ".	Potongan gaji tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Jumlah potongan harus berupa angka saja"/" <i>Deduction amount must be a number</i> ".	✓	

TC25	<i>Admin Super</i> mengubah nama potongan menjadi nama yang sudah digunakan potongan lain preferensi bahasa Indonesia/Inggris.	Potongan gaji tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama potongan sudah digunakan."/" <i>Deduction name already exists.</i> ".	Potongan gaji tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama potongan sudah digunakan."/" <i>Deduction name already exists.</i> ".	✓	
TC26	<i>Admin Super</i> mengubah nama potongan menjadi lebih dari 255 karakter, preferensi bahasa Indonesia/Inggris.	Potongan gaji tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama potongan tidak boleh lebih dari 255 karakter."/" <i>Deduction name must not exceed 255 characters.</i> ".	Potongan gaji tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Nama potongan tidak boleh lebih dari 255 karakter."/" <i>Deduction name must not exceed 255 characters.</i> ".	✓	
TC27	<i>Admin Super</i> mengubah <i>Persentase</i> menjadi kosong, preferensi bahasa Indonesia/Inggris.	Potongan gaji tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Jumlah potongan harus berupa angka saja"/" <i>Deduction amount must be a number.</i> ".	Potongan gaji tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Jumlah potongan harus berupa angka saja"/" <i>Deduction amount must be a number.</i> ".	✓	
TC28	<i>Admin Super</i> mengubah <i>Persentase</i> menjadi nilai negatif (kurang dari 0), preferensi	Potongan gaji tidak berhasil diperbarui dan sistem menampilkan pesan validasi " <i>Persentase</i> minimal 0."/" <i>Percentage cannot be less than 0.</i> ".	Potongan gaji tidak berhasil diperbarui dan sistem menampilkan pesan validasi " <i>Persentase</i> minimal 0."/" <i>Percentage cannot be less than 0.</i> ".	✓	

	bahasa Indonesia/Inggris.				
TC29	<i>Admin Super</i> mengubah <i>Persentase</i> menjadi nilai lebih dari 100, preferensi bahasa Indonesia/Inggris.	Potongan gaji tidak berhasil diperbarui dan sistem menampilkan pesan validasi " <i>Persentase</i> maksimal 100."/" <i>Percentage cannot be greater than 100.</i> ".	Potongan gaji tidak berhasil diperbarui dan sistem menampilkan pesan validasi " <i>Persentase</i> maksimal 100."/" <i>Percentage cannot be greater than 100.</i> ".	✓	
TC30	<i>Admin Super</i> mengubah <i>Persentase</i> dengan karakter non-numerik (huruf atau karakter khusus), preferensi bahasa Indonesia/Inggris.	Potongan gaji tidak berhasil diperbarui dan sistem menampilkan pesan validasi " <i>Persentase</i> hanya boleh angka."/" <i>Percentage must contain numbers only.</i> ".	Potongan gaji tidak berhasil diperbarui dan sistem menampilkan pesan validasi " <i>Persentase</i> hanya boleh angka."/" <i>Percentage must contain numbers only.</i> ".	✓	
TC31	<i>Admin Super</i> memilih potongan gaji yang ada untuk dihapus, preferensi bahasa Indonesia/Inggris.	Potongan gaji berhasil dihapus dari <i>database</i> dan sistem menampilkan pesan " <i>Potongan</i> berhasil dihapus"/" <i>Deduction deleted successfully</i> ".	Potongan gaji berhasil dihapus dari <i>database</i> dan sistem menampilkan pesan " <i>Potongan</i> berhasil dihapus"/" <i>Deduction deleted successfully</i> ".	✓	

Hasil analisis:

1) *Persentase test case* yang lolos = $\frac{31}{31} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{31} = 0\%$

3) *Persentase test case* yang telah diuji = $31 = 100\%$

4) *Persentase test case* yang belum diuji = 0 = 0%

Jumlah total *test case* dalam pengujian = 31

10. Testing *Scenario Use Case* Manajemen Gaji

Pengujian ini didasari oleh *Scenario Use Case* Manajemen Gaji (Tabel 3.22). *Use case scenario* mendefinisikan masuk ke dalam halaman gaji, namun pengujian dilakukan secara detail untuk memvalidasi fitur manajemen gaji. Hal ini untuk memastikan manajemen gaji berjalan lancar.

Tabel 4.11 *Black Box* Testing Manajemen Gaji

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin Super</i> dan <i>Admin Office</i> mengakses halaman penggaji.	Sistem menampilkan halaman daftar data gaji dengan informasi nama karyawan, gaji pokok, gaji kotor, total potongan, gaji bersih, dan status.	Sistem menampilkan halaman daftar gaji sesuai spesifikasi.	✓	
TC02	<i>Admin Super</i> dan <i>Admin Office</i> melakukan perhitungan gaji pada bulan yang belum ada data absensi.	Sistem tetap menghitung gaji dengan total kehadiran = 0 dan gaji kotor = 0 untuk semua karyawan.	Sistem menghitung gaji dengan nilai 0 untuk karyawan tanpa kehadiran.	✓	
TC03	<i>Admin Super</i> dan <i>Admin Office</i> mengklik tombol "Ubah Status" pada salah satu data gaji.	Sistem menampilkan <i>modal/dropdown</i> untuk memilih status pembayaran.	Modal ubah status muncul dengan pilihan status yang tersedia.	✓	

TC04	<i>Admin Super dan Admin Office</i> memilih status baru dan menekan tombol "Simpan".	Sistem memperbarui status gaji dan menampilkan notifikasi "Status gaji berhasil diubah".	Status gaji diperbarui, notifikasi "Status gaji berhasil diubah" muncul.	✓	
TC05	<i>Admin Super dan Admin Office</i> melihat daftar periode gaji yang tersedia di <i>database</i> .	Sistem menampilkan <i>dropdown</i> /daftar periode (bulan dan tahun) yang memiliki data gaji, diurutkan dari terbaru ke terlama	Daftar periode gaji ditampilkan sesuai urutan.	✓	
TC06	Saat pada view web <i>Admin Super</i> dan <i>Admin Office</i> mengklik tombol lihat <i>detail</i> .	Sistem memunculkan <i>detail</i> data gaji.	Sistem memunculkan <i>detail</i> data gaji.	✓	

Hasil analisis:

- 1) *Persentase test case* yang lolos = $\frac{6}{6} = 100\%$
 - 2) *Persentase test case* yang gagal = $\frac{0}{6} = 0\%$
 - 3) *Persentase test case* yang telah diuji = 6 = 100%
 - 4) *Persentase test case* yang belum diuji = 0 = 0%
- Jumlah total *test case* dalam pengujian = 6

11. Testing Scenario Use Case Export Excel Gaji

Pengujian ini didasari oleh *Scenario Use Case* Export Excel Gaji (Tabel 3.23). Use case scenario mendefinisikan masuk ke dalam halaman gaji, namun pengujian dilakukan secara detail untuk memvalidasi fitur export gaji. Hal ini untuk memastikan export gaji berjalan lancar.

Tabel 4.12 *Black Box* Testing Export Excel Gaji

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	Admin mengklik fitur ekspor gaji pada halaman penggajian.	Sistem menampilkan <i>dropdown</i> untuk memilih periode ekspor (bulan dan tahun).	<i>Dropdown</i> ekspor dengan pilihan bulan dan tahun ditampilkan.	✓	
TC02	Admin memilih periode dan mengklik tombol "Ekspor ke Excel".	Sistem mengunduh file Excel dengan nama "Laporan_HR_[NamaBulan]_[Tahun].xlsx" yang berisi 5 <i>sheet</i> : Ringkasan, Gaji, Absensi, Cuti, dan Lembur.	File Excel terunduh dengan format nama dan struktur <i>sheet</i> yang benar.	✓	
TC03	Admin membuka <i>sheet</i> "Ringkasan" pada file hasil ekspor.	<i>Sheet</i> berisi judul laporan, tanggal ekspor, periode, tabel ringkasan data (total karyawan, total absensi, total cuti, total lembur, total gaji bersih), dan top 5 kehadiran terbaik.	Data ringkasan lengkap dan akurat sesuai periode.	✓	
TC04	Admin membuka <i>sheet</i> "Absensi" pada file hasil ekspor.	<i>Sheet</i> berisi tabel data absensi dengan kolom (No, Nama, Tanggal, Hari, Check In, Check Out, Durasi Kerja, Status).	Data absensi lengkap dengan perhitungan durasi kerja yang benar.	✓	
TC05	Admin membuka <i>sheet</i> "Cuti" pada file hasil ekspor.	<i>Sheet</i> berisi tabel data cuti dengan kolom (No, Nama, Tipe Cuti, Tanggal Mulai,	Data cuti lengkap dengan perhitungan durasi yang benar.	✓	

		Tanggal Selesai, Durasi (Hari), Alasan, Status).			
TC06	<i>Admin</i> membuka <i>sheet</i> "Lembur" pada file hasil ekspor.	<i>Sheet</i> berisi tabel data lembur dengan kolom (No, Nama, Tanggal, Jam Mulai, Jam Selesai, Durasi (Jam), Deskripsi, Status).	Data lembur lengkap dengan perhitungan durasi jam yang benar.	✓	
TC07	<i>Admin</i> mengekspor tanpa memilih periode.	Sistem gagal mengekspor dan memunculkan validasi gagal.	Sistem gagal mengekspor dan memunculkan validasi gagal.	✓	

Hasil analisis:

1) *Persentase test case* yang lolos = $\frac{7}{7} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{7} = 0\%$

3) *Persentase test case* yang telah diuji = 7 = 100%

4) *Persentase test case* yang belum diuji = 0 = 0%

Jumlah total *test case* dalam pengujian = 7

12. Testing *Scenario Use Case* Manajemen Peningat

Pengujian ini didasari oleh *Scenario Use Case* Manajemen Peningat (Tabel 3.24). *Use case scenario* mendefinisikan masuk ke dalam halaman gaji, namun pengujian dilakukan secara detail untuk memvalidasi fitur tambah, edit, dan hapus peningat. Hal ini untuk memastikan manajemen peningat berjalan lancar.

Tabel 4.13 *Black Box* Testing Manajemen Peningat

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> mengakses	Sistem menampilkan semua data peningat (judul, deskripsi, tanggal	Sistem menampilkan semua data peningat (judul, deskripsi, tanggal jatuh tempo, status,	✓	

	halaman daftar pengingat dengan data pengingat tersedia di <i>database</i> .	jatuh tempo, status, PIC/peran, sisa hari/jam) diurutkan berdasarkan tanggal jatuh tempo.	PIC/peran, sisa hari/jam) diurutkan berdasarkan tanggal jatuh tempo.		
TC02	<i>Admin Super/Admin Office</i> mengklik tombol tambah pengingat.	Sistem memunculkan <i>modal input</i> tambah pengingat dengan <i>field</i> PIC/peran, judul, deskripsi, tanggal jatuh tempo, dan status.	Sistem memunculkan <i>modal input</i> tambah pengingat dengan <i>field</i> PIC/peran, judul, deskripsi, tanggal jatuh tempo, dan status.	✓	
TC03	<i>Admin Super/Admin Office</i> mengklik tombol hapus pengingat.	Sistem memunculkan <i>modal</i> konfirmasi hapus pengingat.	Sistem memunculkan <i>modal</i> konfirmasi hapus pengingat.	✓	
TC04	<i>Admin Super/Admin Office</i> mengklik tombol <i>delete</i> pada <i>modal</i> konfirmasi hapus pengingat.	Sistem menutup <i>modal</i> dan menampilkan notifikasi.	Sistem menutup <i>modal</i> dan menampilkan notifikasi.	✓	
TC05	<i>Admin Super/Admin Office</i> mengklik tombol batal pada <i>modal</i> konfirmasi.	Sistem menutup <i>modal</i> konfirmasi tanpa adanya perubahan data.	Sistem menutup <i>modal</i> konfirmasi tanpa adanya perubahan data.	✓	

TC06	Admin Super/Admin Office mengklik tombol edit pada salah satu data pengingat.	Sistem memunculkan halaman edit data pengingat dengan data yang sudah terisi otomatis.	Sistem memunculkan halaman edit data pengingat dengan data yang sudah terisi otomatis.	✓	
TC07	Admin Super/Admin Office mengklik tombol submit pada modal edit pengingat.	Sistem menutup modal dan menampilkan pesan notifikasi.	Sistem menutup modal dan menampilkan pesan notifikasi.	✓	
TC08	Admin Super/Admin Office mengklik tombol cancel pada modal edit pengingat.	Sistem menutup modal konfirmasi tanpa ada perubahan data.	Sistem menutup modal konfirmasi tanpa ada perubahan data.	✓	
TC09	Admin Super/Admin Office mengisi semua field dengan benar (PIC, nama pengingat, deskripsi, tanggal jatuh tempo) preferensi bahasa Indonesia/Inggris.	Pengingat berhasil disimpan dengan tanggal jatuh tempo diset ke jam 23:59:59 dan sistem menampilkan pesan "Pengingat berhasil dibuat"/"Reminder created successfully".	Pengingat berhasil disimpan dengan tanggal jatuh tempo diset ke jam 23:59:59 dan sistem menampilkan pesan "Pengingat berhasil dibuat"/"Reminder created successfully".	✓	
TC10	Admin Super/Admin	Pengingat tidak berhasil disimpan dan sistem	Pengingat tidak berhasil disimpan dan sistem	✓	

	<i>Office</i> tidak mengisi semua <i>field</i> atau salah satu <i>field</i> , preferensi bahasa Indonesia/Inggris.	menampilkan pesan validasi "Harap isi semua <i>field</i> "/"Please fill in all data".	menampilkan pesan validasi "Harap isi semua <i>field</i> "/"Please fill in all data".		
TC11	<i>Admin Super/Admin Office</i> mengisi judul yang sudah ada di <i>database</i> (duplikat), preferensi bahasa Indonesia/Inggris.	Pengingat tidak berhasil disimpan dan sistem menampilkan pesan validasi "Judul pengingat sudah digunakan."/"Reminder title already exists.".	Pengingat tidak berhasil disimpan dan sistem menampilkan pesan validasi "Judul pengingat sudah digunakan."/"Reminder title already exists.".	✓	
TC12	<i>Admin Super/Admin Office</i> mengisi judul lebih dari 255 karakter, preferensi bahasa Indonesia/Inggris.	Pengingat tidak berhasil disimpan dan sistem menampilkan pesan validasi "Judul tidak boleh lebih dari 255 karakter."/"Title must not exceed 255 characters.".	Pengingat tidak berhasil disimpan dan sistem menampilkan pesan validasi "Judul tidak boleh lebih dari 255 karakter."/"Title must not exceed 255 characters.".	✓	
TC13	<i>Admin Super/Admin Office</i> memilih pengingat yang ada dan mengubah satu atau beberapa <i>field</i> dengan data valid, preferensi bahasa Indonesia/Inggris.	Data pengingat berhasil diperbarui dan sistem menampilkan pesan "Pengingat berhasil diperbarui"/"Reminder updated successfully".	Data pengingat berhasil diperbarui dan sistem menampilkan pesan "Pengingat berhasil diperbarui"/"Reminder updated successfully".	✓	

TC14	<i>Admin Super/Admin Office</i> mengubah PIC/peran dengan peran yang valid, preferensi bahasa Indonesia/Inggris.	Data pengingat berhasil diperbarui dengan PIC baru dan sistem menampilkan pesan "Pengingat berhasil diperbarui"/" <i>Reminder successfully</i> ".	Data pengingat berhasil diperbarui dengan PIC baru dan sistem menampilkan pesan "Pengingat berhasil diperbarui"/" <i>Reminder updated successfully</i> ".	✓	
TC15	<i>Admin Super/Admin Office</i> mengubah judul dengan judul baru yang valid, preferensi bahasa Indonesia/Inggris.	Data pengingat berhasil diperbarui dengan judul baru dan sistem menampilkan pesan "Pengingat berhasil diperbarui"/" <i>Reminder updated successfully</i> ".	Data pengingat berhasil diperbarui dengan judul baru dan sistem menampilkan pesan "Pengingat berhasil diperbarui"/" <i>Reminder updated successfully</i> ".	✓	
TC16	<i>Admin Super/Admin Office</i> mengubah deskripsi dengan deskripsi baru, preferensi bahasa Indonesia/Inggris.	Data pengingat berhasil diperbarui dengan deskripsi baru dan sistem menampilkan pesan "Pengingat berhasil diperbarui"/" <i>Reminder updated successfully</i> ".	Data pengingat berhasil diperbarui dengan deskripsi baru dan sistem menampilkan pesan "Pengingat berhasil diperbarui"/" <i>Reminder updated successfully</i> ".	✓	
TC17	<i>Admin Super/Admin Office</i> mengubah tanggal jatuh tempo dengan tanggal baru yang valid, preferensi bahasa Indonesia/Inggris.	Data pengingat berhasil diperbarui dengan tanggal jatuh tempo baru (jam diset ke 23:59:59) dan sistem menampilkan pesan "Pengingat berhasil diperbarui"/" <i>Reminder updated successfully</i> ".	Data pengingat berhasil diperbarui dengan tanggal jatuh tempo baru (jam diset ke 23:59:59) dan sistem menampilkan pesan "Pengingat berhasil diperbarui"/" <i>Reminder updated successfully</i> ".	✓	

TC18	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> mengubah status dari "Pending" menjadi "Selesai".	Data pengingat berhasil diperbarui dengan status "Selesai".	Data pengingat berhasil diperbarui dengan status "Selesai".	✓	
TC19	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> mengubah judul menjadi judul yang sudah digunakan pengingat lain, preferensi bahasa Indonesia/Inggris.	Pengingat tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Judul pengingat sudah digunakan."/" <i>Reminder title already exists.</i> ".	Pengingat tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Judul pengingat sudah digunakan."/" <i>Reminder title already exists.</i> ".	✓	
TC20	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> mengubah judul menjadi lebih dari 255 karakter, preferensi bahasa Indonesia/Inggris.	Pengingat tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Judul tidak boleh lebih dari 255 karakter."/" <i>Title must not exceed 255 characters.</i> ".	Pengingat tidak berhasil diperbarui dan sistem menampilkan pesan validasi "Judul tidak boleh lebih dari 255 karakter."/" <i>Title must not exceed 255 characters.</i> ".	✓	
TC21	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> memilih pengingat yang ada untuk dihapus, preferensi bahasa Indonesia/Inggris.	Pengingat berhasil dihapus dari <i>database</i> dan sistem menampilkan pesan "Pengingat berhasil dihapus"/" <i>Reminder deleted successfully.</i> ".	Pengingat berhasil dihapus dari <i>database</i> dan sistem menampilkan pesan "Pengingat berhasil dihapus"/" <i>Reminder deleted successfully.</i> ".	✓	

TC22	Sistem <i>scheduler</i> berjalan pada jam 08:00 (production) atau setiap menit (<i>local</i>) dan terdapat pengingat dengan status "Pending" yang jatuh tempo H-7 atau kurang.	Sistem menjalankan command pengingat: kirim, mencari pengingat yang memenuhi kriteria (H-7 atau kurang), dan mengirim <i>email</i> ke semua <i>user</i> yang terkait dengan peran/PIC pengingat tersebut melalui <i>queue</i> .	Sistem menjalankan command pengingat: kirim, mencari pengingat yang memenuhi kriteria (H-7 atau kurang), dan mengirim <i>email</i> ke semua <i>user</i> yang terkait dengan peran/PIC pengingat tersebut melalui <i>queue</i> .	✓	
TC23	Sistem <i>scheduler</i> berjalan dan terdapat pengingat dengan status "Pending" yang jatuh tempo hari ini (H-0).	Sistem mengirim <i>email</i> pengingat ke semua <i>user</i> yang terkait dengan peran/PIC dan mencatat waktu pengiriman di <i>field last_notified_at</i> .	Sistem mengirim <i>email</i> pengingat ke semua <i>user</i> yang terkait dengan peran/PIC dan mencatat waktu pengiriman di <i>field last_notified_at</i> .	✓	
TC24	Sistem <i>scheduler</i> berjalan dan terdapat pengingat dengan status "Pending" yang jatuh tempo antara H-1 sampai H-7.	Sistem mengirim <i>email</i> pengingat ke semua <i>user</i> yang terkait dengan peran/PIC dan mencatat waktu pengiriman di <i>field last_notified_at</i> .	Sistem mengirim <i>email</i> pengingat ke semua <i>user</i> yang terkait dengan peran/PIC dan mencatat waktu pengiriman di <i>field last_notified_at</i> .	✓	
TC25	Sistem <i>scheduler</i> berjalan dan terdapat pengingat dengan status "Pending" yang sudah terlewat	Sistem tidak mengirim <i>email</i> karena pengingat sudah terlewat (<i>diffInDays</i> negatif).	Sistem tidak mengirim <i>email</i> karena pengingat sudah terlewat (<i>diffInDays</i> negatif).	✓	

	(lewat dari tanggal jatuh tempo).				
TC26	Sistem <i>scheduler</i> berjalan dan terdapat pengingat dengan status "Pending" yang jatuh tempo lebih dari 7 hari lagi (H-8 atau lebih).	Sistem tidak mengirim <i>email</i> karena belum masuk rentang H-7.	Sistem tidak mengirim <i>email</i> karena belum masuk rentang H-7.	✓	
TC27	Sistem <i>scheduler</i> berjalan dan terdapat pengingat dengan status "Selesai" yang jatuh tempo H-7 atau kurang.	Sistem tidak mengirim <i>email</i> karena status bukan "Pending".	Sistem tidak mengirim <i>email</i> karena status bukan "Pending".	✓	
TC28	Sistem <i>scheduler</i> berjalan dan terdapat pengingat dengan status "Pending" yang memenuhi kriteria tetapi peran/PIC tidak memiliki <i>user</i> terkait.	Sistem melewati pengingat tersebut dan mencatat <i>log</i> bahwa peran atau <i>user</i> kosong, tidak ada <i>email</i> yang dikirim.	Sistem melewati pengingat tersebut dan mencatat <i>log</i> bahwa peran atau <i>user</i> kosong, tidak ada <i>email</i> yang dikirim.	✓	

Hasil analisis:

1) *Persentase test case* yang lolos = $\frac{28}{28} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{28} = 0\%$

3) *Persentase test case* yang telah diuji = $28 = 100\%$

4) *Persentase test case* yang belum diuji = 0 = 0%

Jumlah total *test case* dalam pengujian = 28

13. Testing *Scenario Use Case* Manajemen Tugas

Pengujian ini didasari oleh *Scenario Use Case* Manajemen Tugas (Tabel 3.27). *Use case scenario* mendefinisikan masuk ke dalam halaman tugas, namun pengujian dilakukan secara detail untuk memvalidasi fitur tambah, edit, dan hapus tugas. Hal ini untuk memastikan manajemen tugas berjalan lancar.

Tabel 4.14 *Black Box* Testing Manajemen Tugas

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin Super/Admin Office</i> mengakses halaman daftar tugas.	Sistem menampilkan semua data tugas dari semua <i>user</i> dengan informasi nama tugas, pegawai, tanggal penugasan, batas penugasan, status, dan lampiran.	Sistem menampilkan semua data tugas dari semua <i>user</i> dengan informasi nama tugas, pegawai, tanggal penugasan, batas penugasan, status, dan lampiran.	✓	
TC02	<i>Admin Super/Admin Office</i> mengklik tombol Tambah Tugas Baru.	Sistem menampilkan <i>form input</i> dengan <i>field</i> pegawai, nama tugas, tanggal penugasan, batas penugasan, instruksi tugas, koordinat <i>latitude</i> , koordinat <i>longitude</i> , dan radius meter.	Sistem menampilkan <i>form input</i> dengan <i>field</i> pegawai, nama tugas, tanggal penugasan, batas penugasan, instruksi tugas, koordinat <i>latitude</i> , koordinat <i>longitude</i> , dan radius meter.	✓	
TC03	<i>Admin Super/Admin Office</i> mengklik	Sistem memunculkan <i>modal</i> konfirmasi hapus tugas.	Sistem memunculkan <i>modal</i> konfirmasi hapus tugas.	✓	

	tombol hapus tugas.				
TC04	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> mengklik tombol <i>delete</i> pada <i>modal</i> konfirmasi hapus tugas.	Sistem menutup <i>modal</i> dan menampilkan notifikasi.	Sistem menutup <i>modal</i> dan menampilkan notifikasi.	✓	
TC05	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> mengklik tombol batal pada <i>modal</i> konfirmasi hapus tugas.	Sistem menutup <i>modal</i> konfirmasi tanpa adanya perubahan data.	Sistem menutup <i>modal</i> konfirmasi tanpa adanya perubahan data.	✓	
TC06	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> mengklik tombol Edit pada salah satu tugas.	Sistem menampilkan <i>form</i> edit dengan data tugas yang sudah terisi otomatis.	Sistem menampilkan <i>form</i> edit dengan data tugas yang sudah terisi otomatis.	✓	
TC07	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> mengklik <i>dropdown</i> "Ubah Status" pada tugas tertentu.	Sistem menampilkan pilihan status "Proses" dan "Selesai".	Sistem menampilkan pilihan status "Proses" dan "Selesai".	✓	
TC08	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> mengisi semua <i>field</i>	Sistem menyimpan data tugas dengan status <i>default</i> "Proses", mengirim notifikasi ke pegawai yang	Sistem menyimpan data tugas dengan status <i>default</i> "Proses", mengirim notifikasi ke pegawai yang ditugaskan, dan	✓	

	dengan data valid dan menekan tombol "Simpan", preferensi bahasa Indonesia/Inggris.	ditugaskan, dan menampilkan pesan "Tugas berhasil dibuat"/" <i>Task created successfully</i> ".	menampilkan pesan "Tugas berhasil dibuat"/" <i>Task created successfully</i> ".		
TC09	<i>Admin Super/Admin Office</i> mengosongkan <i>field required</i> (pegawai, nama tugas, tanggal penugasan, dll), preferensi bahasa Indonesia/Inggris.	Sistem menampilkan pesan validasi untuk setiap <i>field</i> yang wajib diisi.	Sistem menampilkan pesan validasi untuk setiap <i>field</i> yang wajib diisi.	✓	
TC10	<i>Admin Super/Admin Office</i> mengisi <i>field</i> dengan batas penugasan lebih awal dari tanggal penugasan, preferensi bahasa Indonesia/Inggris.	Sistem menampilkan pesan validasi "Batas penugasan tidak boleh sebelum tanggal penugasan"/" <i>Deadline cannot be before the assignment date</i> ".	Sistem menampilkan pesan validasi "Batas penugasan tidak boleh sebelum tanggal penugasan"/" <i>Deadline cannot be before the assignment date</i> ".	✓	
TC11	<i>Admin Super/Admin Office</i> mengisi <i>field</i> radius meter kurang dari 10 meter, preferensi	Sistem menampilkan pesan validasi "Radius meter minimal 10 meter"/" <i>Radius meter must be at least 10 meters</i> ".	Sistem menampilkan pesan validasi "Radius meter minimal 10 meter"/" <i>Radius meter must be at least 10 meters</i> ".	✓	

	bahasa Indonesia/Inggris.				
TC12	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> mengisi <i>field</i> nama tugas lebih dari 255 karakter, preferensi bahasa Indonesia/Inggris.	Sistem menampilkan pesan validasi "Nama tugas tidak boleh lebih dari 255 karakter"/" <i>Task name must not exceed 255 characters</i> ".	Sistem menampilkan pesan validasi "Nama tugas tidak boleh lebih dari 255 karakter"/" <i>Task name must not exceed 255 characters</i> ".	✓	
TC13	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> mengubah data tugas (tanpa mengganti PIC) dan menekan "Simpan", preferensi bahasa Indonesia/Inggris.	Sistem memperbarui data tugas, mengirim notifikasi <i>update</i> ke pegawai terkait, dan menampilkan pesan "Tugas berhasil diperbarui"/" <i>Task updated successfully</i> ".	Sistem memperbarui data tugas, mengirim notifikasi <i>update</i> ke pegawai terkait, dan menampilkan pesan "Tugas berhasil diperbarui"/" <i>Task updated successfully</i> ".	✓	
TC14	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> mengganti PIC (pegawai) pada tugas dan menekan "Simpan", preferensi bahasa Indonesia/Inggris.	Sistem memperbarui PIC, mengirim notifikasi "Tugas Dialihkan" ke pegawai lama, mengirim notifikasi "Tugas Baru Diberikan" ke pegawai baru, dan menampilkan pesan "Tugas berhasil diperbarui"/" <i>Task updated successfully</i> ".	Sistem memperbarui PIC, mengirim notifikasi "Tugas Dialihkan" ke pegawai lama, mengirim notifikasi "Tugas Baru Diberikan" ke pegawai baru, dan menampilkan pesan "Tugas berhasil diperbarui"/" <i>Task updated successfully</i> ".	✓	
TC15	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> mengubah	Sistem menampilkan pesan validasi "Batas penugasan tidak boleh sebelum tanggal	Sistem menampilkan pesan validasi "Batas penugasan tidak boleh sebelum tanggal	✓	

	batas penugasan lebih awal dari tanggal penugasan saat edit, preferensi bahasa Indonesia/Inggris.	penugasan"/" <i>Deadline cannot be before the assignment date</i> ".	penugasan"/" <i>Deadline cannot be before the assignment date</i> ".		
TC16	<i>Admin Super/Admin Office</i> mengubah radius meter kurang dari 10 meter saat edit, preferensi bahasa Indonesia/Inggris.	Sistem menampilkan pesan validasi "Radius meter minimal 10 meter"/"Radius meter <i>must be at least</i> 10 meters".	Sistem menampilkan pesan validasi "Radius meter minimal 10 meter"/"Radius meter <i>must be at least</i> 10 meters".	✓	
TC17	<i>Admin Super/Admin Office</i> mengubah status tugas dari "Proses" menjadi "Selesai", preferensi bahasa Indonesia/Inggris.	Sistem memperbarui status, mengirim notifikasi "Tugas Selesai" ke pegawai, dan menampilkan pesan "Status tugas berhasil diperbarui"/"Task status <i>updated successfully</i> ".	Sistem memperbarui status, mengirim notifikasi "Tugas Selesai" ke pegawai, dan menampilkan pesan "Status tugas berhasil diperbarui"/"Task status <i>updated successfully</i> ".	✓	
TC18	<i>Admin Super/Admin Office</i> mengubah status tugas dari "Menunggu Admin" menjadi "Proses",	Sistem memperbarui status, mengirim notifikasi "Tugas Diproses" ke pegawai, dan menampilkan pesan "Status tugas berhasil diperbarui"/"Task status <i>updated successfully</i> ".	Sistem memperbarui status, mengirim notifikasi "Tugas Diproses" ke pegawai, dan menampilkan pesan "Status tugas berhasil diperbarui"/"Task status <i>updated successfully</i> ".	✓	

	preferensi bahasa Indonesia/Inggris.				
TC19	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> mengubah status tugas dari "Menunggu <i>Admin</i> " menjadi "Selesai", preferensi bahasa Indonesia/Inggris.	Sistem memperbarui status, mengirim notifikasi "Tugas Selesai" ke pegawai, dan menampilkan pesan "Status tugas berhasil diperbarui"/" <i>Task</i> status <i>updated successfully</i> ".	Sistem memperbarui status, mengirim notifikasi "Tugas Selesai" ke pegawai, dan menampilkan pesan "Status tugas berhasil diperbarui"/" <i>Task</i> status <i>updated successfully</i> ".	✓	
TC20	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> mengonfirmasi penghapusan tugas yang memiliki file lampiran dengan status bukan "Selesai", preferensi bahasa Indonesia/Inggris.	Sistem menghapus data tugas dari <i>database</i> , menghapus file lampiran dari <i>storage</i> , mengirim notifikasi "Tugas Dihapus" ke pegawai, dan menampilkan pesan "Tugas berhasil dihapus"/" <i>Task deleted successfully</i> ".	Sistem menghapus data tugas dari <i>database</i> , menghapus file lampiran dari <i>storage</i> , mengirim notifikasi "Tugas Dihapus" ke pegawai, dan menampilkan pesan "Tugas berhasil dihapus"/" <i>Task deleted successfully</i> ".	✓	
TC21	<i>Admin</i> <i>Super/Admin</i> <i>Office</i> menghapus tugas dengan status "Selesai", preferensi bahasa Indonesia/Inggris.	Sistem menghapus data tugas dan file lampiran, tetapi TIDAK mengirim notifikasi ke pegawai (karena tugas sudah selesai), dan menampilkan pesan "Tugas berhasil	Sistem menghapus data tugas dan file lampiran, tetapi TIDAK mengirim notifikasi ke pegawai (karena tugas sudah selesai), dan menampilkan pesan "Tugas berhasil dihapus"/" <i>Task deleted successfully</i> ".	✓	

		dihapus"/"Task deleted successfully".			
TC22	Admin Super/Admin Office menghapus tugas tanpa file lampiran, preferensi bahasa Indonesia/Inggris.	Sistem menghapus data tugas dari database, mengirim notifikasi "Tugas Dihapus" ke pegawai (jika status bukan "Selesai"), dan menampilkan pesan "Tugas berhasil dihapus"/"Task deleted successfully".	Sistem menghapus data tugas dari database, mengirim notifikasi "Tugas Dihapus" ke pegawai (jika status bukan "Selesai"), dan menampilkan pesan "Tugas berhasil dihapus"/"Task deleted successfully".	✓	

Hasil analisis:

1) Persentase test case yang lolos = $\frac{22}{22} = 100\%$

2) Persentase test case yang gagal = $\frac{0}{22} = 0\%$

3) Persentase test case yang telah diuji = 22 = 100%

4) Persentase test case yang belum diuji = 0 = 0%

Jumlah total test case dalam pengujian = 22

14. Testing Scenario Use Case Upload Lampiran Tugas

Pengujian ini didasari oleh *Scenario Use Case Upload Lampiran Tugas* (Tabel 3.28). Use case scenario mendefinisikan masuk ke dalam halaman tugas, namun pengujian dilakukan secara detail untuk memvalidasi fitur *upload* lampiran tugas yang lebih detail. Hal ini untuk memastikan *upload* lampiran tugas berjalan lancar.

Tabel 4.15 Black Box Testing Upload Lampiran Tugas

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	Teknisi mengakses detail tugas yang	Sistem menampilkan detail tugas dengan informasi nama tugas, instruksi, tanggal penugasan, batas	Detail tugas ditampilkan lengkap dengan tombol upload.	✓	

	ditugaskan kepadanya.	penugasan, lokasi (koordinat), radius, status, dan tombol "Upload Lampiran".			
TC02	Teknisi mengklik tombol "Upload Lampiran".	Sistem menampilkan <i>form</i> upload dengan <i>field</i> file lampiran, koordinat <i>latitude</i> otomatis (dari GPS), dan koordinat <i>longitude</i> otomatis (dari GPS).	<i>Form</i> upload muncul dengan koordinat GPS terekam.	✓	
TC03	Teknisi memilih file lampiran lalu menekan "Upload".	Menyimpan file, mengubah status tugas menjadi "Menunggu Admin", mencatat waktu upload, menghitung keterlambatan, mengirim notifikasi ke <i>user</i> , dan menampilkan pesan "Lampiran berhasil diupload" / " <i>Attachment uploaded successfully</i> ".	<i>File</i> tersimpan, status berubah, waktu dan keterlambatan tercatat, notifikasi terkirim.	✓	
TC04	Teknisi memilih file lampiran lalu menekan "Upload".	Sistem menyimpan file ke <i>folder</i> "tugas/videos", status berubah menjadi "Menunggu Admin", data terekam dengan benar.	<i>File video</i> tersimpan di <i>folder</i> yang sesuai.	✓	
TC05	Teknisi memilih file lampiran dengan format lain	Sistem menyimpan file ke <i>folder</i> "tugas/files", status berubah menjadi "Menunggu Admin".	<i>File</i> tersimpan di <i>folder default</i> untuk file lainnya.	✓	

TC06	Teknisi mencoba upload tanpa memilih file.	Sistem menampilkan pesan validasi "Harap upload lampiran" / "Please upload attachments".	Pesan validasi <i>error</i> muncul, tidak ada upload.	✓	
TC07	Teknisi mencoba upload file dengan ukuran lebih dari 200MB.	Sistem menampilkan pesan validasi "Ukuran lampiran tidak boleh lebih dari 200MB" / " <i>Attachment size must not exceed 200MB</i> ".	Pesan validasi <i>error</i> muncul, file tidak diupload.	✓	
TC08	Teknisi upload lampiran sebelum batas penugasan.	Sistem menyimpan waktu upload, mengubah status menjadi "Menunggu <i>Admin</i> ".	Data tercatat tidak terlambat, status berubah.	✓	
TC09	Teknisi upload lampiran setelah batas penugasan.	Sistem mencatat bahwa tugas terlambat, menghitung menit keterlambatan (selisih waktu upload dengan batas penugasan dalam menit), menyimpan waktu upload, mengubah status menjadi "Menunggu <i>Admin</i> ".	Data tercatat terlambat dengan perhitungan menit yang akurat.	✓	
TC10	Teknisi upload lampiran yang kedua kalinya (mengganti lampiran lama).	Sistem menghapus file lampiran lama dari <i>storage</i> , menyimpan file baru, memperbarui waktu upload dan keterlambatan, status tetap "Menunggu <i>Admin</i> ".	<i>File</i> lama terhapus, file baru tersimpan, data diperbarui.	✓	
TC11	<i>Admin</i> memeriksa file lampiran yang	<i>File</i> tersimpan di <i>folder</i> yang sesuai berdasarkan	<i>File</i> tersimpan di lokasi yang benar dan dapat diakses.	✓	

	diupload teknis di <i>storage</i> .	ekstensi (tugas/images, tugas/videos, atau tugas/files) dan dapat diakses via URL.		
--	--	---	--	--

Hasil analisi:

1) *Persentase test case* yang lolos = $\frac{11}{11} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{11} = 0\%$

3) *Persentase test case* yang telah diuji = 11 = 100%

4) *Persentase test case* yang belum diuji = 0 = 0%

Jumlah total *test case* dalam pengujian = 11

15. Testing *Scenario Use Case* Melihat Daftar Absensi

Pengujian ini didasari oleh *Scenario Use Case* Melihat Daftar Absensi (Tabel 3.25). *Use case scenario* mendefinisikan masuk ke dalam halaman absensi, namun pengujian dilakukan secara detail untuk memvalidasi melihat daftar absensi yang lebih detail. Hal ini untuk memastikan melihat daftar absensi berjalan lancar.

Tabel 4.16 *Black Box* Testing Melihat Daftar Absensi

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin Super</i> dengan fitur "lihat_semua_absensi" mengakses halaman daftar absensi.	Sistem menampilkan semua data absensi dari seluruh karyawan dengan informasi nama <i>user</i> , tanggal <i>check-in</i> , waktu , waktu <i>check-out</i> , status, koordinat, dan <i>video</i> .	Sistem menampilkan seluruh daftar absensi semua karyawan.	✓	
TC02	<i>Admin Office/Teknisi</i> dengan fitur "lihat_absensi_sendiri"	Sistem hanya menampilkan data	Sistem menampilkan hanya absensi <i>user</i> tersebut.	✓	

	mengakses halaman daftar absensi.	absensi milik <i>user</i> yang sedang <i>login</i> .			
TC03	<i>User</i> mengakses daftar absensi yang sudah terurut.	Sistem menampilkan data absensi diurutkan berdasarkan waktu <i>check-in</i> dari terbaru ke terlama.	Data absensi ditampilkan sesuai urutan waktu.	✓	
TC04	<i>Admin Super</i> melihat <i>detail</i> absensi tertentu.	Sistem menampilkan <i>detail</i> lengkap absensi nama <i>user</i> , tanggal dan waktu <i>check-in/out</i> , koordinat GPS <i>check-in/out</i> , status kehadiran, dan <i>video</i> absensi.	<i>Detail</i> absensi lengkap ditampilkan.	✓	
TC05	<i>User</i> melihat absensi yang belum melakukan <i>check-out</i> .	Sistem menampilkan data absensi dengan waktu <i>check-out</i> masih kosong.	Data <i>check-out</i> ditampilkan sebagai "-" atau " <i>null</i> ".	✓	
TC06	<i>User</i> mengklik <i>video</i> absensi pada daftar.	Sistem menampilkan <i>video</i> absensi yang tersimpan di <i>storage</i> dengan URL yang valid.	<i>Video</i> dapat diputar dengan lancar.	✓	

Hasil analisi:

1) *Persentase test case* yang lolos = $\frac{6}{6} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{6} = 0\%$

3) *Persentase test case* yang telah diuji = 6 = 100%

4) *Persentase test case* yang belum diuji = 0 = 0%

Jumlah total *test case* dalam pengujian = 6

16. Testing *Scenario Use Case* Melakukan Absensi

Pengujian ini didasari oleh *Scenario Use Case* Melakukan Absensi (Tabel 3.26). Use case scenario mendefinisikan masuk ke dalam halaman absensi, namun pengujian dilakukan secara detail untuk memvalidasi melakukan absensi yang lebih detail. Hal ini untuk memastikan melakukan absensi berjalan lancar.

Tabel 4.17 *Black Box* Testing Melakukan Absensi

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	User mengakses fitur <i>check-in</i> absensi.	Sistem menampilkan <i>form check-in</i> dengan <i>field</i> koordinat GPS (otomatis), tanggal (otomatis), waktu (otomatis), dan pilihan upload <i>video</i> .	<i>Form check-in</i> muncul dengan data GPS dan waktu terekam.	✓	
TC02	User berada di dalam radius kantor dan melakukan <i>check-in</i> dengan upload <i>video</i> file (mp4/avi/mpeg/mov).	Sistem menghitung jarak menggunakan <i>haversine formula</i> , memvalidasi lokasi dalam radius, menyimpan <i>video</i> ke <i>folder</i> "absensi/video", menentukan status "Tepat Waktu" atau "Terlambat", menyimpan data absensi, dan menampilkan pesan " <i>Check-in</i> berhasil".	Data absensi tersimpan dengan <i>video</i> dan status yang benar.	✓	
TC03	User berada di luar radius kantor dan mencoba <i>check-in</i> .	Sistem menampilkan pesan <i>error</i> "Anda berada di luar radius kantor!".	Notifikasi <i>error</i> muncul, <i>check-in</i> ditolak.	✓	

TC04	<i>User</i> mencoba <i>check-in</i> pada tanggal yang sudah pernah <i>check-in</i> .	Sistem menampilkan pesan <i>error</i> "Anda sudah melakukan <i>check-in</i> pada tanggal ini." dengan status.	Notifikasi <i>error</i> muncul, <i>check-in</i> ditolak.	✓	
TC05	<i>User</i> melakukan <i>check-in</i> sebelum atau sama dengan batas toleransi keterlambatan.	Sistem menyimpan absensi dengan status "Tepat Waktu".	Status absensi tercatat sebagai "Tepat Waktu".	✓	
TC06	<i>User</i> melakukan <i>check-in</i> setelah batas toleransi keterlambatan.	Sistem menyimpan absensi dengan status "Terlambat" berdasarkan perhitungan $\text{jam_masuk} + \text{minimal_keterlambatan}$ dari data kantor.	Status absensi tercatat sebagai "Terlambat".	✓	
TC07	<i>User</i> melakukan <i>check-in</i> dengan upload <i>video</i> dalam format base64.	Sistem <i>decode</i> base64, menyimpan <i>video</i> ke <i>folder</i> "absensi" dengan nama unik, menyimpan URL <i>video</i> di <i>database</i> .	<i>Video</i> tersimpan dengan format base64 berhasil dikonversi.	✓	
TC08	<i>User</i> melakukan <i>check-in</i> tanpa upload <i>video</i> .	Sistem menampilkan pesan validasi <i>error</i> "Harap rekam <i>video</i> dahulu sebelum <i>submit</i> ".	Sistem menampilkan pesan validasi <i>error</i> "Harap rekam <i>video</i> dahulu sebelum <i>submit</i> ".	✓	
TC09	<i>User</i> mengakses fitur <i>check-out</i> setelah melakukan <i>check-in</i> .	Sistem menampilkan <i>form check-out</i> dengan <i>field</i> koordinat GPS (otomatis), tanggal (otomatis), dan waktu (otomatis).	<i>Form check-out</i> muncul.	✓	

TC10	<i>User</i> melakukan <i>check-out</i> pada tanggal yang sama dengan <i>check-in</i> .	Sistem memperbarui data absensi dengan koordinat <i>check-out</i> , tanggal <i>check-out</i> , dan waktu <i>check-out</i> , lalu menampilkan pesan " <i>Check-out</i> berhasil".	Data <i>check-out</i> tersimpan, pesan sukses muncul.	✓	
TC11	<i>User</i> mencoba <i>check-out</i> tanpa melakukan <i>check-in</i> terlebih dahulu.	Sistem menampilkan pesan <i>error</i> "Anda belum melakukan <i>check-in</i> pada tanggal ini."	<i>Check-out</i> ditolak dan menampilkan pesan "Anda belum melakukan <i>check-in</i> pada tanggal ini."	✓	
TC12	<i>User</i> mencoba <i>check-out</i> dua kali pada tanggal yang sama.	Sistem menampilkan pesan <i>error</i> "Anda sudah melakukan <i>check-out</i> hari ini."	<i>Check-out</i> kedua ditolak dan menampilkan pesan "Anda sudah melakukan <i>check-out</i> hari ini."	✓	
TC13	<i>User</i> memeriksa <i>video</i> absensi.	Sistem memunculkan <i>modal</i> dan memutar <i>video user</i> .	Sistem memunculkan <i>modal</i> dan memutar <i>video user</i> .	✓	

Hasil analisis:

$$1) \text{ Persentase test case yang lolos} = \frac{13}{13} = 100\%$$

$$2) \text{ Persentase test case yang gagal} = \frac{0}{13} = 0\%$$

$$3) \text{ Persentase test case yang telah diuji} = 13 = 100\%$$

$$4) \text{ Persentase test case yang belum diuji} = 0 = 0\%$$

$$\text{Jumlah total test case dalam pengujian} = 13$$

17. Testing Scenario Use Case Pengajuan Cuti

Pengujian ini didasari oleh *Scenario Use Case* Pengajuan Cuti (Tabel 3.32).

Use case scenario mendefinisikan masuk ke dalam halaman cuti, namun pengujian dilakukan secara detail untuk memvalidasi melakukan pengajuan cuti yang lebih detail. Hal ini untuk memastikan melakukan pengajuan cuti berjalan lancar.

Tabel 4.18 *Black Box* Testing Pengajuan Cuti

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin Office</i> /Teknisi mengakses fitur pengajuan cuti.	Sistem menampilkan <i>form</i> pengajuan cuti dengan <i>field</i> tipe cuti, tanggal mulai, tanggal selesai, dan alasan.	<i>Form</i> pengajuan cuti ditampilkan lengkap.	✓	
TC02	<i>User</i> mengisi semua <i>field</i> dengan data valid dan menekan tombol " <i>Submit</i> ".	Sistem menyimpan pengajuan cuti dengan status "Pending", mengirim notifikasi ke pemohon, mengirim notifikasi ke semua <i>user</i> dengan fitur " <i>approve_cuti_step1</i> ", dan menampilkan pesan "Pengajuan cuti berhasil dikirim" / " <i>Leave request submitted successfully</i> ".	Data cuti tersimpan, notifikasi terkirim, pesan sukses muncul.	✓	
TC03	<i>User</i> mengisi tanggal selesai lebih awal dari tanggal mulai.	Sistem menampilkan pesan validasi "Tanggal selesai harus setelah atau sama dengan tanggal mulai" / " <i>End date must be after or equal to start date</i> ".	Pesan validasi <i>error</i> muncul, data tidak tersimpan.	✓	
TC04	<i>User</i> mengosongkan <i>field</i> .	Sistem menampilkan pesan validasi untuk setiap <i>field</i> wajib diisi.	Sistem menampilkan pesan validasi untuk setiap <i>field</i> wajib diisi.	✓	

TC05	User mengisi alasan dengan lebih dari 255 karakter.	Sistem menampilkan pesan validasi "Alasan tidak boleh lebih dari 255 karakter" / <i>"Reason must not exceed 255 characters"</i> .	Pesan validasi <i>error</i> muncul, data tidak tersimpan.	✓	
TC06	User mencoba mengajukan cuti baru saat masih memiliki pengajuan cuti dengan status "Pending" atau "Proses".	Sistem menampilkan pesan <i>error</i> "Anda masih memiliki pengajuan cuti yang belum diproses. Selesaikan dulu sebelum mengajukan cuti baru" / <i>"You still have an unprocessed leave request. Finish it before creating a new one"</i> .	Sistem memunculkan notifikasi dan pengajuan cuti tidak terkirim.	✓	
TC07	User mengajukan cuti baru setelah pengajuan sebelumnya sudah berstatus "Disetujui" atau "Ditolak".	Sistem menyimpan pengajuan cuti baru dengan normal.	Pengajuan cuti baru berhasil tersimpan.	✓	
TC08	User mengakses daftar cuti dengan fitur "lihat_cuti_sendiri".	Sistem menampilkan hanya data cuti milik <i>user</i> yang sedang <i>login</i> , diurutkan dari terbaru.	Daftar cuti <i>user</i> tersebut ditampilkan.	✓	

Hasil analisis:

1) Persentase test case yang lolos = $\frac{8}{8} = 100\%$

2) Persentase test case yang gagal = $\frac{0}{8} = 0\%$

3) Persentase test case yang telah diuji = $8 = 100\%$

4) *Persentase test case* yang belum diuji = 0 = 0%

Jumlah total *test case* dalam pengujian = 8

18. Testing *Scenario Use Case* Menyetujui Cuti Awal

Pengujian ini didasari oleh *Scenario Use Case* Menyetujui Cuti Awal (Tabel 3.33). *Use case scenario* mendefinisikan masuk ke dalam halaman cuti, namun pengujian dilakukan secara detail untuk memvalidasi menyetujui cuti awal yang lebih detail. Hal ini untuk memastikan melakukan menyetujui cuti awal berjalan lancar.

Tabel 4.19 *Black Box* Testing Menyetujui Cuti Awal

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin Office</i> dengan fitur "approve_cuti_step1" mengakses daftar cuti.	Sistem menampilkan SEMUA data cuti tanpa filter approval_step (termasuk step 0, 1, 2, 3).	Daftar semua cuti ditampilkan.	✓	
TC02	<i>Admin Office</i> mengklik tombol "Disetujui" pada cuti dengan approval_step = 0 (Pending).	Sistem mengubah approval_step menjadi 1, status menjadi "Proses", mengirim notifikasi "Cuti Disetujui Tahap Awal" ke pemohon, mengirim notifikasi ke semua user dengan fitur "approve_cuti_step2", dan menampilkan pesan "Cuti disetujui tahap awal".	approval_step = 1, status = "Proses", notifikasi terkirim, pesan sukses muncul.	✓	

TC03	<i>Admin Office</i> mencoba menyetujui cuti yang sudah <code>approval_step = 1</code> atau 2.	Sistem menampilkan pesan <i>error</i> "Cuti sudah diproses tahap awal".	Cuti tidak diproses ulang dan sistem memunculkan notifikasi.	✓	
TC04	<i>Admin Office</i> mencoba menolak cuti yang sudah disetujui (<code>approval_step = 1</code>) dan mengisi <i>modal</i> penolakan serta mengklik tombol lanjut.	Sistem menampilkan <i>modal input</i> catatan penolakan dan pesan sukses.	Sistem menampilkan <i>modal input</i> catatan penolakan dan pesan sukses.	✓	
TC05	<i>Admin Office</i> dengan fitur " <i>decline_cuti</i> " mengklik tombol "Tolak" pada cuti dengan <code>approval_step < 2</code> .	Sistem menampilkan <i>modal input</i> untuk catatan penolakan.	Modal <i>input</i> catatan penolakan muncul.	✓	
TC06	<i>Admin Office</i> mengisi catatan penolakan dan menekan tombol "Lanjut".	Sistem mengubah <code>approval_step</code> menjadi 3, status menjadi "Ditolak", menyimpan <code>catatan_penolakan</code> , mengirim notifikasi "Cuti Ditolak" ke pemohon, dan menampilkan pesan "Cuti ditolak dengan catatan revisi".	<code>approval_step = 3</code> , status = "Ditolak", catatan tersimpan, notifikasi terkirim.	✓	

TC07	<i>Admin Office</i> mencoba menolak cuti tanpa mengisi catatan penolakan.	Tombol lanjut tidak bisa diklik dan data tidak berubah.	Tombol lanjut tidak bisa diklik dan data tidak berubah.	✓	
TC08	<i>Admin Office</i> mencoba menolak cuti dengan approval_step = 2 (sudah final).	Sistem menampilkan notifikasi “statusajuan cuti sudah final, tidak dapat diubah kembali” dan data tidak berubah.	Sistem menampilkan notifikasi dan data tidak berubah.	✓	
TC09	Pemohon cuti menerima notifikasi setelah cutinya disetujui tahap awal.	Notifikasi muncul dengan informasi bahwa cuti disetujui tahap awal.	Notifikasi terkirim ke pemohon.	✓	
TC10	Pemohon cuti menerima notifikasi setelah cutinya ditolak.	Notifikasi muncul dengan informasi bahwa cuti ditolak beserta catatan penolakan.	Notifikasi terkirim ke pemohon dengan catatan.	✓	

Hasil analisis:

1) *Persentase test case* yang lolos = $\frac{10}{10} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{10} = 0\%$

3) *Persentase test case* yang telah diuji = 10 = 100%

4) *Persentase test case* yang belum diuji = 0 = 0%

Jumlah total *test case* dalam pengujian = 10

19. Testing *Scenario Use Case* Menyetujui Cuti Akhir

Pengujian ini didasari oleh *Scenario Use Case* Menyetujui Cuti Akhir (Tabel 3.33). *Use case scenario* mendefinisikan masuk ke dalam halaman cuti, namun pengujian dilakukan secara detail untuk memvalidasi menyetujui cuti akhir yang lebih detail. Hal ini untuk memastikan melakukan menyetujui cuti akhir berjalan lancar.

Tabel 4.20 *Black Box* Testing Menyetujui Cuti Akhir

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin Super</i> dengan fitur " <i>approve_cuti_step2</i> " mengakses daftar cuti.	Sistem menampilkan hanya cuti yang sudah lolos step1, final, atau ditolak.	Daftar cuti yang sudah step1 ke atas ditampilkan.	✓	
TC02	<i>Admin Super</i> mengklik tombol "Disetujui" pada cuti dengan <i>approval_step</i> = 1.	Sistem mengubah <i>approval_step</i> menjadi 2, status menjadi "Disetujui", mengirim notifikasi "Cuti Disetujui Final" ke pemohon, dan menampilkan pesan "Cuti disetujui final".	<i>approval_step</i> = 2, status = "Disetujui", notifikasi terkirim, pesan sukses muncul.	✓	
TC03	<i>Admin Super</i> dengan fitur " <i>decline_cuti</i> " mengklik tombol "Ditolak" pada cuti dengan <i>approval_step</i> = 1	Sistem menampilkan <i>modal input</i> untuk catatan penolakan.	Modal <i>input</i> catatan penolakan muncul.	✓	
TC04	<i>Admin Super</i> mengisi catatan penolakan dan menekan tombol "Tolak".	Sistem mengubah <i>approval_step</i> menjadi 3, status menjadi "Ditolak", menyimpan <i>catatan_penolakan</i> , mengirim notifikasi "Cuti Ditolak" ke pemohon, dan menampilkan pesan "Cuti	<i>approval_step</i> = 3, status = "Ditolak", catatan tersimpan, notifikasi terkirim.	✓	

		ditolak dengan catatan revisi".			
TC05	Admin Super mencoba menolak cuti dengan approval_step = 2 (sudah final).	Sistem menampilkan pesan <i>error</i> "Cuti sudah final, tidak bisa ditolak" dan data tidak tersimpan.	Sistem menampilkan pesan <i>error</i> "Cuti sudah final, tidak bisa ditolak" dan data tidak tersimpan.	✓	
TC06	Admin Super mencoba menolak cuti tanpa mengisi catatan penolakan.	Tombol lanjut tidak bisa ditekan dan data tidak tersimpan.	Tombol lanjut tidak bisa ditekan dan data tidak tersimpan.	✓	
TC07	Pemohon cuti menerima notifikasi setelah cutinya disetujui final.	Notifikasi muncul dengan informasi bahwa cuti disetujui final dan dapat digunakan.	Notifikasi terkirim ke pemohon.	✓	
TC08	Pemohon cuti menerima notifikasi setelah cutinya ditolak.	Notifikasi muncul dengan informasi bahwa cuti ditolak beserta catatan penolakan.	Notifikasi terkirim ke pemohon dengan catatan.	✓	

Hasil analisis:

1) *Persentase test case* yang lolos = $\frac{8}{8} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{8} = 0\%$

3) *Persentase test case* yang telah diuji = $8 = 100\%$

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

Jumlah total *test case* dalam pengujian = 8

20. Testing Scenario Use Case Pengajuan Lembur

Pengujian ini didasari oleh *Scenario Use Case* Pengajuan Lembur (Tabel 3.29). Use case scenario mendefinisikan masuk ke dalam halaman lembur, namun pengujian dilakukan secara detail untuk memvalidasi pengajuan

lembur yang lebih detail. Hal ini untuk memastikan melakukan pengajuan lembur berjalan lancar.

Tabel 4.21 *Black Box* Testing Pengajuan Lembur

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	Admin Office/Teknisi mengakses fitur pengajuan lembur.	Sistem menampilkan <i>form</i> pengajuan lembur dengan <i>field</i> tanggal, jam mulai, jam selesai, dan deskripsi.	<i>Form</i> pengajuan lembur ditampilkan lengkap.	✓	
TC02	User mengisi semua <i>field</i> dengan data valid dan menekan tombol "Ajukan".	Sistem menyimpan pengajuan lembur dengan status "Pending", mengirim notifikasi ke pemohon, mengirim notifikasi ke semua user dengan fitur " <i>approve_lembur_step1</i> ", dan menampilkan pesan "Pengajuan lembur berhasil dikirim" / " <i>Overtime request submitted successfully</i> ".	Data lembur tersimpan, notifikasi terkirim, pesan sukses muncul.	✓	
TC03	User mengisi jam selesai lebih awal dari jam mulai.	Sistem menampilkan pesan validasi "Jam selesai harus lebih besar dari jam mulai" / " <i>End time must be after start time</i> ".	Pesan validasi <i>error</i> muncul, data tidak tersimpan.	✓	
TC04	User mengosongkan <i>field</i> wajib.	Sistem menampilkan pesan validasi untuk setiap <i>field</i> wajib diisi.	Pesan validasi <i>error</i> muncul untuk <i>field</i> wajib	✓	

TC05	User mengisi deskripsi dengan lebih dari 255 karakter.	Sistem menampilkan pesan validasi "Deskripsi tidak boleh lebih dari 255 karakter" / "Description must not exceed 255 characters".	Pesan validasi <i>error</i> muncul, data tidak tersimpan.	✓	
TC06	User mencoba mengajukan lembur baru saat masih memiliki pengajuan lembur dengan status "Pending" atau "Proses".	Sistem menampilkan pesan <i>error</i> "Anda masih memiliki pengajuan lembur yang belum diproses. Selesaikan dulu sebelum mengajukan lembur baru" / "You still have an unprocessed overtime request. Finish it before creating a new one".	Sistem memunculkan notifikasi dan pengajuan lembur tidak terkirim.	✓	
TC07	User mengajukan lembur baru setelah pengajuan sebelumnya sudah berstatus "Disetujui" atau "Ditolak".	Sistem menyimpan pengajuan lembur baru dengan normal.	Pengajuan lembur baru berhasil tersimpan.	✓	
TC08	User mengakses daftar lembur dengan fitur "lihat_lembur_sendiri".	Sistem menampilkan hanya data lembur milik <i>user</i> yang sedang <i>login</i> , diurutkan dari terbaru.	Daftar lembur <i>user</i> tersebut ditampilkan.	✓	

Hasil analisis:

1) Persentase test case yang lolos = $\frac{8}{8} = 100\%$

2) Persentase test case yang gagal = $\frac{0}{8} = 0\%$

3) *Persentase test case yang telah diuji* = 8 = 100%

4) *Persentase test case yang belum diuji* = 0 = 0%

Jumlah total *test case* dalam pengujian = 8

21. Testing Scenario Use Case Menyetujui Lembur Awal

Pengujian ini didasari oleh *Scenario Use Case Menyetujui Lembur Awal* (Tabel 3.30). Use case scenario mendefinisikan masuk ke dalam halaman lembur, namun pengujian dilakukan secara detail untuk memvalidasi menyetujui lembur awal yang lebih detail. Hal ini untuk memastikan melakukan pengajuan lembur berjalan lancar.

Tabel 4.22 *Black Box Testing Menyetujui Lembur Awal*

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin Office</i> dengan fitur "approve_lembur_step1" mengakses daftar lembur.	Sistem menampilkan SEMUA data lembur tanpa filter approval_step (termasuk step 0, 1, 2, 3).	Daftar semua lembur ditampilkan.	✓	
TC02	<i>Admin Office</i> mengklik tombol "Disetujui" pada lembur dengan approval_step = 0 (Pending).	Sistem mengubah approval_step menjadi 1, status menjadi "Proses", mengirim notifikasi "Lembur Disetujui Tahap Awal" ke pemohon, mengirim notifikasi ke semua user dengan fitur "approve_lembur_step2", dan menampilkan pesan "Lembur disetujui tahap awal".	approval_step = 1, status = "Proses", notifikasi terkirim, pesan sukses muncul.	✓	

TC03	<i>Admin Office</i> mencoba menyetujui lembur yang sudah approval_step = 1 atau 2.	Sistem menampilkan pesan <i>error</i> "Lembur sudah diproses tahap awal".	Lembur tidak diproses ulang dan sistem memunculkan notifikasi.	✓	
TC04	<i>Admin Office</i> mencoba menolak lembur yang sudah disetujui (approval_step = 1) dan mengisi <i>modal</i> penolakan serta mengklik tombol lanjut.	Sistem menampilkan <i>modal input</i> catatan penolakan dan pesan sukses.	Sistem menampilkan <i>modal input</i> catatan penolakan dan pesan sukses.	✓	
TC05	<i>Admin Office</i> dengan fitur "decline_lembur" mengklik tombol "Tolak" pada lembur dengan approval_step < 2.	Sistem menampilkan <i>modal input</i> untuk catatan penolakan.	Modal <i>input</i> catatan penolakan muncul.	✓	
TC06	<i>Admin Office</i> mengisi catatan penolakan dan menekan tombol "Lanjut".	Sistem mengubah approval_step menjadi 3, status menjadi "Ditolak", menyimpan catatan_penolakan, mengirim notifikasi "Lembur Ditolak" ke pemohon, dan menampilkan pesan "Lembur ditolak dengan catatan revisi".	approval_step = 3, status = "Ditolak", catatan tersimpan, notifikasi terkirim.	✓	

TC07	<i>Admin Office</i> mencoba menolak lembur tanpa mengisi catatan penolakan.	Tombol lanjut tidak bisa diklik dan data tidak berubah.	Tombol lanjut tidak bisa diklik dan data tidak berubah.	✓	
TC08	<i>Admin Office</i> mencoba menolak lembur dengan <code>approval_step = 2</code> (sudah final).	Sistem menampilkan notifikasi "statusajuan lembur sudah final tidak dapat di ubah kembali" dan data tidak berubah.	Sistem menampilkan notifikasi dan data tidak berubah.	✓	
TC09	Pemohon lembur menerima notifikasi setelah lemburnya disetujui tahap awal.	Notifikasi muncul dengan informasi bahwa lembur disetujui tahap awal.	Notifikasi terkirim ke pemohon.	✓	
TC10	Pemohon lembur menerima notifikasi setelah lemburnya ditolak.	Notifikasi muncul dengan informasi bahwa lembur ditolak beserta catatan penolakan.	Notifikasi terkirim ke pemohon dengan catatan.	✓	

Hasil analisis:

1) *Persentase test case* yang lolos = $\frac{10}{10} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{10} = 0\%$

3) *Persentase test case* yang telah diuji = 10 = 100%

4) *Persentase test case* yang belum diuji = 0 = 0%

Jumlah total *test case* dalam pengujian = 10

22. Testing *Scenario Use Case* Menyetujui Lembur Akhir

Pengujian ini didasari oleh *Scenario Use Case* Menyetujui Lembur Akhir (Tabel 3.31). *Use case scenario* mendefinisikan masuk ke dalam halaman lembur, namun pengujian dilakukan secara detail untuk memvalidasi menyetujui lembur akhir yang lebih detail. Hal ini untuk memastikan melakukan pengajuan lembur berjalan lancar.

Tabel 4.23 *Black Box* Testing Menyetujui Lembur Akhir

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	Admin Super dengan fitur "approve_lembur_step2" mengakses daftar lembur.	Sistem menampilkan hanya lembur dengan lembur yang sudah lolos step1, final, atau ditolak.	Daftar lembur yang sudah step1 ke atas ditampilkan.	✓	
TC02	Admin Super mengklik tombol "Disetujui" pada lembur dengan approval_step = 1.	Sistem mengubah approval_step menjadi 2, status menjadi "Disetujui", mengirim notifikasi "Lembur Disetujui Final" ke pemohon, dan menampilkan pesan "Lembur disetujui final".	approval_step = 2, status = "Disetujui", notifikasi terkirim, pesan sukses muncul.	✓	
TC03	Admin Super dengan fitur "decline_lembur" mengklik tombol "Ditolak" pada lembur dengan approval_step = 1.	Sistem menampilkan modal input untuk catatan penolakan.	Modal input catatan penolakan muncul.	✓	
TC04	Admin Super mengisi catatan penolakan dan menekan tombol "Tolak".	Sistem mengubah approval_step menjadi 3, status menjadi "Ditolak", menyimpan catatan_penolakan, mengirim notifikasi "Lembur Ditolak" ke	approval_step = 3, status = "Ditolak", catatan tersimpan, notifikasi terkirim.	✓	

		pemohon, dan menampilkan pesan "Lembur ditolak dengan catatan revisi".		
TC05	<i>Admin Super</i> mencoba menolak lembur dengan approval_step = 2 (sudah final).	Sistem menampilkan pesan error "Lembur sudah final, tidak bisa ditolak" dan data tidak tersimpan.	Sistem menampilkan pesan error "Lembur sudah final, tidak bisa ditolak" dan data tidak tersimpan.	✓
TC06	<i>Admin Super</i> mencoba menolak lembur tanpa mengisi catatan penolakan.	Tombol lanjut tidak bisa di tekan dan data tidak tersimpan.	Tombol lanjut tidak bisa di tekan dan data tidak tersimpan.	✓
TC07	Pemohon lembur menerima notifikasi setelah lemburnya disetujui final.	Notifikasi muncul dengan informasi bahwa lembur disetujui final dan dapat digunakan.	Notifikasi terkirim ke pemohon.	✓
TC08	Pemohon lembur menerima notifikasi setelah lemburnya disetujui tahap akhir.	Notifikasi muncul dengan informasi bahwa lembur disetujui tahap akhir.	Notifikasi terkirim ke pemohon.	✓
TC09	Pemohon lembur menerima notifikasi setelah lemburnya ditolak.	Notifikasi muncul dengan informasi bahwa lembur ditolak beserta catatan penolakan.	Notifikasi terkirim ke pemohon dengan catatan.	✓

Hasil analisi:

- 1) *Persentase test case* yang lolos = $\frac{8}{8} = 100\%$
- 2) *Persentase test case* yang gagal = $\frac{0}{8} = 0\%$
- 3) *Persentase test case* yang telah diuji = 8 = 100%
- 4) *Persentase test case* yang belum diuji = 0 = 0%

Jumlah total *test case* dalam pengujian = 8

23. Testing Scenario Use Case Mengelola Profile

Pengujian ini didasari oleh *Scenario Use Case Mengelola Profile* (Tabel 3.35). Use case scenario mendefinisikan masuk ke dalam halaman profile, namun pengujian dilakukan secara detail untuk memvalidasi mengubah data *profile* yang lebih detail. Hal ini untuk memastikan mengelola *profile* berjalan lancar.

Tabel 4.24 *Black Box Testing Mengelola Profile*

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin Super/Admin Office/Teknisi</i> mengakses halaman profil.	Sistem menampilkan data profil <i>user</i> yang sedang <i>login</i> (nama, <i>email</i> , peran, dll).	Sistem menampilkan data profil <i>user</i> yang sedang <i>login</i> (nama, <i>email</i> , peran, dll).	✓	
TC02	<i>Admin Super/Admin Office/Teknisi</i> membuka <i>modal</i> ubah <i>email</i> .	Sistem menampilkan <i>modal form</i> dengan <i>field password</i> lama dan <i>email</i> baru.	Sistem menampilkan <i>modal form</i> dengan <i>field password</i> lama dan <i>email</i> baru.	✓	
TC03	<i>Admin Super/Admin Office/Teknisi</i> mengisi <i>password</i> lama dengan benar dan <i>email</i> baru yang valid (belum digunakan).	<i>Email</i> berhasil diperbarui dan sistem menampilkan pesan " <i>Email</i> berhasil diperbarui", <i>modal</i> tertutup.	<i>Email</i> berhasil diperbarui dan sistem menampilkan pesan " <i>Email</i> berhasil diperbarui", <i>modal</i> tertutup.	✓	

TC04	<i>Admin Super/Admin Office/Teknisi</i> mengisi <i>password</i> lama dengan salah dan <i>email</i> baru yang valid.	<i>Email</i> tidak berhasil diperbarui dan sistem menampilkan pesan " <i>Password</i> salah"	<i>Email</i> tidak berhasil diperbarui dan sistem menampilkan pesan " <i>Password</i> salah".	✓	
TC05	<i>Admin Super/Admin Office/Teknisi</i> tidak mengisi <i>password</i> lama (<i>field</i> kosong).	<i>Email</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	<i>Email</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	✓	
TC06	<i>Admin Super/Admin Office/Teknisi</i> tidak mengisi <i>email</i> baru (<i>field</i> kosong).	<i>Email</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	<i>Email</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	✓	
TC07	<i>Admin Super/Admin Office/Teknisi</i> mengisi <i>email</i> baru dengan format yang tidak valid.	<i>Email</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	<i>Email</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	✓	
TC08	<i>Admin Super/Admin Office/Teknisi</i> mengisi <i>email</i>	<i>Email</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	<i>Email</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	✓	

	baru yang sudah digunakan <i>user</i> lain.				
TC09	<i>Admin Super/Admin Office/Teknisi</i> mengisi <i>email</i> baru lebih dari 255 karakter.	<i>Email</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	<i>Email</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	✓	
TC10	<i>Admin Super/Admin Office/Teknisi</i> membuka <i>modal</i> ubah <i>password</i> .	Sistem menampilkan <i>modal form</i> dengan <i>field password</i> lama, <i>password</i> baru, dan konfirmasi <i>password</i> baru.	Sistem menampilkan <i>modal form</i> dengan <i>field password</i> lama, <i>password</i> baru, dan konfirmasi <i>password</i> baru.	✓	
TC11	<i>Admin Super/Admin Office/Teknisi</i> mengisi <i>password</i> lama dengan benar, <i>password</i> baru minimal 6 karakter, dan konfirmasi <i>password</i> sama dengan <i>password</i> baru.	<i>Password</i> berhasil diperbarui dan sistem menampilkan pesan " <i>Password</i> berhasil diperbarui", <i>modal</i> tertutup.	<i>Password</i> berhasil diperbarui dan sistem menampilkan pesan " <i>Password</i> berhasil diperbarui", <i>modal</i> tertutup.	✓	
TC12	<i>Admin Super/Admin Office/Teknisi</i> mengisi <i>password</i>	<i>Password</i> tidak berhasil diperbarui dan sistem menampilkan pesan " <i>Password</i> lama salah".	<i>Password</i> tidak berhasil diperbarui dan sistem menampilkan pesan " <i>Password</i> lama salah".	✓	

	lama dengan salah.				
TC13	<i>Admin</i> <i>Super/Admin</i> <i>Office/Teknisi</i> tidak mengisi <i>password</i> lama (<i>field</i> kosong).	<i>Password</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	<i>Password</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	✓	
TC14	<i>Admin</i> <i>Super/Admin</i> <i>Office/Teknisi</i> tidak mengisi <i>password</i> baru (<i>field</i> kosong).	<i>Password</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	<i>Password</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	✓	
TC15	<i>Admin</i> <i>Super/Admin</i> <i>Office/Teknisi</i> mengisi <i>password</i> baru kurang dari 6 karakter.	<i>Password</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	<i>Password</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	✓	
TC16	<i>Admin</i> <i>Super/Admin</i> <i>Office/Teknisi</i> mengisi konfirmasi <i>password</i> tidak sama dengan <i>password</i> baru.	<i>Password</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	<i>Password</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	✓	

TC17	<i>Admin Super/Admin Office/Teknisi</i> tidak mengisi konfirmasi <i>password</i> (<i>field</i> kosong).	<i>Password</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	<i>Password</i> tidak berhasil diperbarui dan sistem menampilkan pesan validasi.	✓	
------	--	--	--	---	--

Hasil analisis:

1) *Persentase test case* yang lolos = $\frac{17}{17} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{17} = 0\%$

3) *Persentase test case* yang telah diuji = 17 = 100%

4) *Persentase test case* yang belum diuji = 0 = 0%

Jumlah total *test case* dalam pengujian = 17

24. Testing Scenario Use Case Mengelola Pengaturan

Pengujian ini didasari oleh *Scenario Use Case* Mengelola Pengaturan (Tabel 3.36). Use case scenario mendefinisikan masuk ke dalam halaman pengaturan, namun pengujian dilakukan secara detail untuk memvalidasi mengelola pengaturan yang lebih detail. Hal ini untuk memastikan pengelola pengaturan berjalan lancar.

Tabel 4.25 *Black Box* Testing Mengelola Pengaturan

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin Super/Admin Office/Teknisi</i> mengakses halaman pengaturan untuk pertama kali	Sistem menampilkan pengaturan <i>default</i> (tema gelap, bahasa Indonesia) dan menyimpan ke <i>database</i> .	Sistem menampilkan pengaturan <i>default</i> (tema gelap, bahasa Indonesia) dan menyimpan ke <i>database</i> dengan pesan.	✓	

	(belum ada data pengaturan).				
TC02	<i>Admin</i> <i>Super/Admin</i> <i>Office/Teknisi</i> mengakses halaman pengaturan dengan data pengaturan sudah tersimpan.	Sistem menampilkan pengaturan <i>user</i> yang tersimpan (tema dan bahasa sesuai pilihan sebelumnya).	Sistem menampilkan pengaturan <i>user</i> yang tersimpan (tema dan bahasa sesuai pilihan sebelumnya).	✓	
TC03	<i>Admin</i> <i>Super/Admin</i> <i>Office/Teknisi</i> mengklik tombol tema "Terang".	Tema aplikasi berubah menjadi terang, pengaturan tersimpan ke <i>databas</i> .	Tema aplikasi berubah menjadi terang, pengaturan tersimpan ke <i>database</i> .	✓	
TC04	<i>Admin</i> <i>Super/Admin</i> <i>Office/Teknisi</i> mengklik tombol tema "Gelap".	Tema aplikasi berubah menjadi gelap, pengaturan tersimpan ke <i>database</i> .	Tema aplikasi berubah menjadi gelap, pengaturan tersimpan ke <i>database</i> .	✓	
TC05	<i>Admin</i> <i>Super/Admin</i> <i>Office/Teknisi</i> mengklik tombol bahasa "Indonesia".	Bahasa aplikasi berubah menjadi Indonesia, pengaturan tersimpan ke <i>database</i> .	Bahasa aplikasi berubah menjadi Indonesia, pengaturan tersimpan ke <i>database</i> .	✓	
TC06	<i>Admin</i> <i>Super/Admin</i> <i>Office/Teknisi</i>	Bahasa aplikasi berubah menjadi Inggris,	Bahasa aplikasi berubah menjadi Inggris, pengaturan tersimpan ke <i>database</i> .	✓	

	mengklik tombol bahasa "Inggris".	pengaturan tersimpan ke <i>database</i> .			
TC07	<i>Admin</i> <i>Super/Admin</i> <i>Office/Teknisi</i> mengubah tema dan bahasa sekaligus dalam satu waktu.	Tema dan bahasa aplikasi berubah sesuai pilihan, pengaturan tersimpan ke <i>database</i> .	Tema dan bahasa aplikasi berubah sesuai pilihan, pengaturan tersimpan ke <i>database</i> .	✓	
TC08	<i>Admin</i> <i>Super/Admin</i> <i>Office/Teknisi</i> <i>logout</i> dan <i>login</i> kembali setelah mengubah pengaturan.	Sistem menampilkan tema dan bahasa sesuai dengan pengaturan yang tersimpan sebelumnya.	Sistem menampilkan tema dan bahasa sesuai dengan pengaturan yang tersimpan sebelumnya.	✓	

Hasil analisis:

- 1) *Persentase test case* yang lolos = $\frac{8}{8} = 100\%$
 - 2) *Persentase test case* yang gagal = $\frac{0}{8} = 0\%$
 - 3) *Persentase test case* yang telah diuji = $8 = 100\%$
 - 4) *Persentase test case* yang belum diuji = $0 = 0\%$
- Jumlah total *test case* dalam pengujian = 8

25. Testing Scenario Use Case Menghapus Device Karyawan

Pengujian ini didasari oleh *Scenario Use Case Menghapus Device Karyawan* (Tabel 3.39). Use case scenario mendefinisikan masuk ke dalam halaman *device*, namun pengujian dilakukan secara detail untuk memvalidasi menghapus *device* karyawan yang lebih detail. Hal ini untuk memastikan menghapus *device* karyawan berjalan lancar.

Tabel 4.26 *Black Box* Testing Menghapus *Device* Karyawan

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>Admin Super</i> mengakses halaman manajemen <i>device</i> .	Sistem menampilkan daftar semua <i>device</i> yang terdaftar dengan informasi <i>user</i> pemilik, <i>device</i> id, <i>device</i> name, dan versi <i>device</i> .	Daftar semua <i>device</i> ditampilkan lengkap.	✓	
TC02	<i>Admin Super</i> mengklik tombol "Reset <i>device</i> " pada salah satu <i>user</i> .	Sistem menampilkan <i>modal</i> konfirmasi inputan reset <i>devicenya</i> .	Modal konfirmasi inputan muncul.	✓	
TC03	<i>Admin Super</i> mengisi inputan dengan benar dan mengklik konfirmasi.	Modal inputan tertutup, <i>device</i> terhapus dari <i>database</i> , dan notifikasi sukses muncul.	Modal inputan tertutup, <i>device</i> terhapus dari <i>database</i> , dan notifikasi sukses muncul.	✓	
TC04	<i>Admin Super</i> mengisi inputan dengan benar dan mengklik konfirmasi.	Tombol konfirmasi tidak dapat di tekan dan data tidak ada berubah.	Tombol konfirmasi tidak dapat di tekan dan data tidak ada berubah.	✓	
TC05	<i>Admin Super</i> membatalkan reset <i>device</i> dengan mengklik tombol "Batal" di <i>modal</i> konfirmasi.	Modal tertutup tanpa ada perubahan data <i>device</i> .	Modal tertutup, <i>device</i> tidak terhapus.	✓	

TC06	User yang deviceny di-reset mencoba login dari device lama.	Sistem kembali mencatat device user dan berhasil login.	Sistem kembali mencatat device user dan berhasil login.	✓	
TC07	User yang deviceny di-reset mencoba login dari device baru.	Sistem menerima login dan mendaftarkan device baru untuk user tersebut.	Login berhasil dengan device baru terdaftar.	✓	

Hasil analisi:

1) Persentase test case yang lolos = $\frac{7}{7} = 100\%$

2) Persentase test case yang gagal = $\frac{0}{7} = 0\%$

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

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

Jumlah total test case dalam pengujian = 7

26. Testing Scenario Use Case Membuka Akun Karyawan

Pengujian ini didasari oleh Scenario Use Case Membuka Akun Karyawan (Tabel 3.38). Use case scenario mendefinisikan masuk ke dalam halaman buka akun, namun pengujian dilakukan secara detail untuk memvalidasi membuka akun karyawan yang lebih detail. Hal ini untuk memastikan membuka akun karyawan berjalan lancar.

Tabel 4.27 Black Box Membuka Akun Karyawan

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	Admin Super mengakses halaman buka akun.	Sistem menampilkan daftar user nama dan email.	Daftar akun terkunci ditampilkan.	✓	
TC02	Admin Super mengklik tombol	Sistem menampilkan modal konfirmasi inputan.	Modal konfirmasi inputan muncul.	✓	

	buka akun pada salah satu data.				
TC03	<i>Admin Super</i> mengisi <i>modal</i> konfirmasi inputan dengan benar dan menekan tombol “Konfirmasi Buka”.	Sistem memperbarui data akun <i>user</i> , menampilkan pesan "Akun berhasil di buka", dan akun <i>user</i> hilang dari daftar akun terkunci.	Akun terbuka, data berhasil di perbarui, akun hilang dari daftar akun terkunci dan pesan sukses muncul.	✓	
TC04	<i>Admin Super</i> mengisi <i>modal</i> konfirmasi inputan dengan salah dan menekan tombol “Konfirmasi Buka”.	Tombol “Konfirmasi Buka” tidak bisa di tekan, <i>modal</i> masi terbuka dan data tidak ada yang berubah.	Data tidak ada perubahan, tombol tidak bisa diklik, dan <i>modal</i> masi terbuka.	✓	
TC05	<i>Admin Super</i> membatalkan pembukaan akun dengan mengklik tombol "Batal" di <i>modal</i> konfirmasi.	Modal konfirmasi inputan tertutup tanpa ada perubahan data <i>user</i> .	Modal tertutup, status akun tetap terkunci.	✓	
TC06	<i>User</i> yang akunnya sudah dibuka mencoba <i>login</i> dengan	Sistem menerima <i>login</i> dan <i>user</i> dapat mengakses aplikasi dengan normal.	<i>Login</i> berhasil, akses normal.	✓	

	<i>password</i> yang benar.				
--	-----------------------------	--	--	--	--

Hasil analisi:

- 1) *Persentase test case* yang lolos = $\frac{6}{6} = 100\%$
 - 2) *Persentase test case* yang gagal = $\frac{0}{6} = 0\%$
 - 3) *Persentase test case* yang telah diuji = 6 = 100%
 - 4) *Persentase test case* yang belum diuji = 0 = 0%
- Jumlah total *test case* dalam pengujian = 6

27. Testing Scenario Use Case Logout

Pengujian ini didasari oleh *Scenario Use Case Logout* (Tabel 3.13). Use case scenario mendefinisikan keluar dari sistem, namun pengujian dilakukan secara detail untuk memastikan *logout* berjalan lancar.

Tabel 4.28 *Black Box Testing Logout*

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	G
TC01	<i>User</i> yang sudah <i>login</i> mengakses endpoint <i>logout</i> dengan token valid.	Sistem menghapus <i>device</i> token <i>user</i> , menghapus current access token Sanctum, mencatat aktivitas <i>logout</i> ke <i>log</i> , dan menampilkan <i>login</i> .	Sistem memperbarui data <i>user</i> , mencatat aktivitas <i>logout</i> ke <i>log</i> , dan menampilkan <i>login</i> .	✓	
TC02	<i>User</i> berhasil <i>logout</i> dan mencoba mengakses endpoint dengan url.	Sistem menolak akses dan menampilkan halaman landing page.	Sitem menolak dan mengarahkan ke halaman landing page.	✓	
TC03	<i>User</i> yang sudah <i>logout</i> mencoba <i>login</i> kembali dari <i>device</i> yang sama.	Sistem menerima <i>login</i> dan membuat access token baru.	<i>Login</i> berhasil dengan token baru.	✓	

Hasil analisi:

- 1) *Persentase test case yang lolos* = $\frac{3}{3} = 100\%$
 - 2) *Persentase test case yang gagal* = $\frac{0}{3} = 0\%$
 - 3) *Persentase test case yang telah diuji* = 3 = 100%
 - 4) *Persentase test case yang belum diuji* = 0 = 0%
- Jumlah total *test case* dalam pengujian = 3

28. Testing *Scenario Use Case* Lupa Kata Sandi

Pengujian ini didasari oleh *Scenario Use Case* Lupa Kata Sandi (Tabel 3.37). Use case scenario mendefinisikan masuk ke dalam halaman lupa kata sandi, namun pengujian dilakukan secara detail untuk memvalidasi lupa kata sandi yang lebih detail. Hal ini untuk memastikan lupa kata sandi berjalan lancar.

Tabel 4.29 *Black Box* Testing Lupa Kata Sandi

Tes Case ID	Skenario	Luaran Yang Diharapkan	Luaran Yang Dihasilkan	Hasil	
				B	S
TC01	User menekan tombol forget <i>password</i> .	Sistem menampilkan halaman inputan <i>email</i> .	Sistem menampilkan halaman forget <i>password</i> .	✓	
TC02	User menekan tombol kembali ke <i>login</i> .	Sistem mengarahkan <i>user</i> kehalaman <i>login</i> .	Sistem mengarahkan <i>user</i> kehalaman <i>login</i> .	✓	
TC03	User mengisi inputan <i>email</i> reset <i>password</i> dengan <i>email</i> valid yang terdaftar.	Sistem mengirim <i>email</i> berisi link reset <i>password</i> , menyimpan token ke tabel <i>password</i> reset tokens, dan menampilkan pesan "Link reset <i>password</i> sudah dikirim ke <i>email</i> Anda."	<i>Email</i> terkirim, token tersimpan, pesan sukses muncul.	✓	

TC04	<i>User</i> memasukkan <i>email</i> yang tidak terdaftar di sistem.	Sistem menampilkan pesan validasi <i>error</i> " <i>Email</i> ini belum terdaftar di sistem."	Pesan validasi <i>error</i> muncul, <i>email</i> tidak dikirim.	✓	
TC05	<i>User</i> memasukkan <i>email</i> dengan format invalid.	Sistem menampilkan pesan validasi <i>error</i> "Format <i>email</i> tidak valid."	Pesan validasi <i>error</i> muncul, <i>email</i> tidak dikirim.	✓	
TC06	<i>User</i> tidak mengisi <i>field email</i> .	Sistem menampilkan pesan validasi <i>error</i> " <i>Email</i> wajib diisi."	Pesan validasi <i>error</i> muncul.	✓	
TC07	<i>User</i> request reset <i>password</i> berulang kali untuk <i>email</i> yang sama.	Sistem mengirim <i>email</i> baru setiap kali request (token lama akan diganti/expired).	<i>Email</i> terkirim setiap request, token diperbarui.	✓	
TC08	<i>User</i> memeriksa <i>email</i> yang diterima.	<i>Email</i> berisi link reset <i>password</i> dengan format https://domain.com/password/reset/{token}?email={email} .	<i>Email</i> diterima dengan link yang valid.	✓	
TC09	<i>User</i> mengklik link reset <i>password</i> dari <i>email</i> .	Sistem menampilkan <i>form</i> reset <i>password</i> dengan <i>field email</i> , <i>password</i> baru, dan konfirmasi <i>password</i> .	<i>Form</i> reset <i>password</i> ditampilkan dengan <i>email</i> sudah terisi.	✓	
TC10	<i>User</i> mengisi <i>password</i> baru dan konfirmasi <i>password</i>	Sistem mengubah <i>password user</i> dengan hash baru, menghapus token reset dari <i>database</i> , dan <i>redirect</i> ke	<i>Password</i> diperbarui, <i>redirect</i> ke halaman sukses.	✓	

	dengan benar (sama dan minimal 8 karakter).	halaman sukses " <i>Password</i> berhasil diubah".			
TC11	<i>User</i> mengisi <i>password</i> baru kurang dari 8 karakter.	Sistem menampilkan pesan validasi <i>error</i> " <i>Password</i> minimal 8 karakter."	Pesan validasi <i>error</i> muncul, <i>password</i> tidak diubah.	✓	
TC12	<i>User</i> mengisi <i>password</i> baru dan konfirmasi <i>password</i> tidak sama.	Sistem menampilkan pesan validasi <i>error</i> "Konfirmasi <i>password</i> tidak sama dengan <i>password</i> baru."	Pesan validasi <i>error</i> muncul, <i>password</i> tidak diubah.	✓	
TC13	<i>User</i> tidak mengisi <i>password</i> baru atau konfirmasi <i>password</i> .	Sistem menampilkan pesan validasi <i>error</i> " <i>Password</i> baru wajib diisi."	Pesan validasi <i>error</i> muncul.	✓	
TC14	<i>User</i> menggunakan token yang sudah expired atau invalid.	Sistem menampilkan pesan <i>error</i> "Token tidak valid atau sudah kadaluarsa." dan kembali ke <i>form</i> .	Error muncul, <i>password</i> tidak diubah.	✓	
TC15	<i>User</i> menggunakan link reset <i>password</i> yang sudah pernah digunakan dan	Sistem menampilkan pesan <i>error</i> "Token tidak valid atau sudah kadaluarsa."	Error muncul, token tidak bisa digunakan ulang.	✓	

	mengisi inputan dengan benar.				
TC16	User mengubah parameter <i>email</i> di URL menjadi <i>email</i> lain.	Sistem menampilkan <i>error</i> validasi atau token invalid karena <i>email</i> tidak cocok dengan token.	Error validasi muncul, reset gagal.	✓	
TC17	User berhasil reset <i>password</i> dan mencoba <i>login</i> dengan <i>password</i> baru.	Sistem menerima <i>login</i> dengan <i>password</i> baru.	<i>Login</i> berhasil dengan <i>password</i> baru.	✓	
TC18	User mencoba <i>login</i> dengan <i>password</i> lama setelah berhasil reset.	Sistem menolak <i>login</i> karena <i>password</i> sudah berubah.	<i>Login</i> gagal dengan <i>password</i> lama.	✓	

Hasil analisis:

1) *Persentase test case* yang lolos = $\frac{18}{18} = 100\%$

2) *Persentase test case* yang gagal = $\frac{0}{18} = 0\%$

3) *Persentase test case* yang telah diuji = $18 = 100\%$

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

Jumlah total *test case* dalam pengujian = 18

4.2.2 Pengujian Perangkat Lunak (Non-Functional Testing)

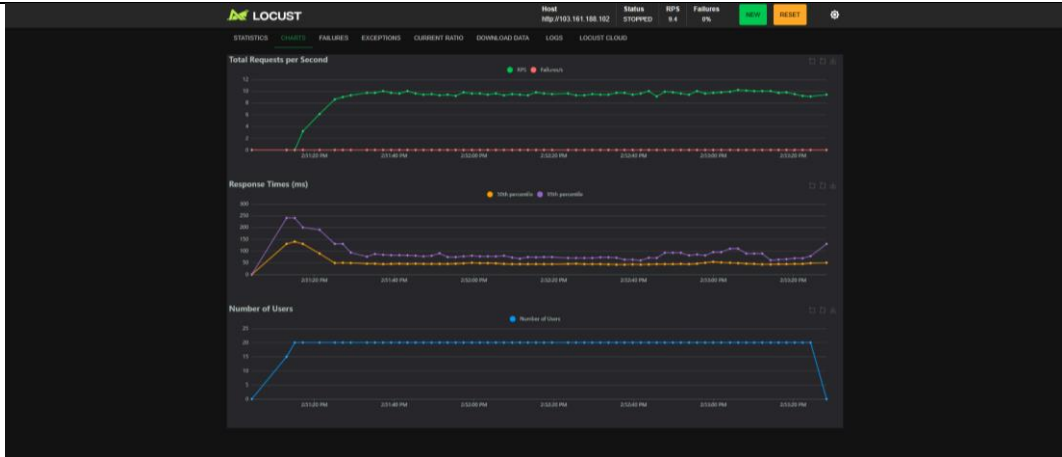
Pada tahap ini dilakukan pengujian *non-fungsional* untuk memastikan bahwa seluruh keperluan nonfungsional perangkat lunak bekerja sesuai dengan kebutuhan.

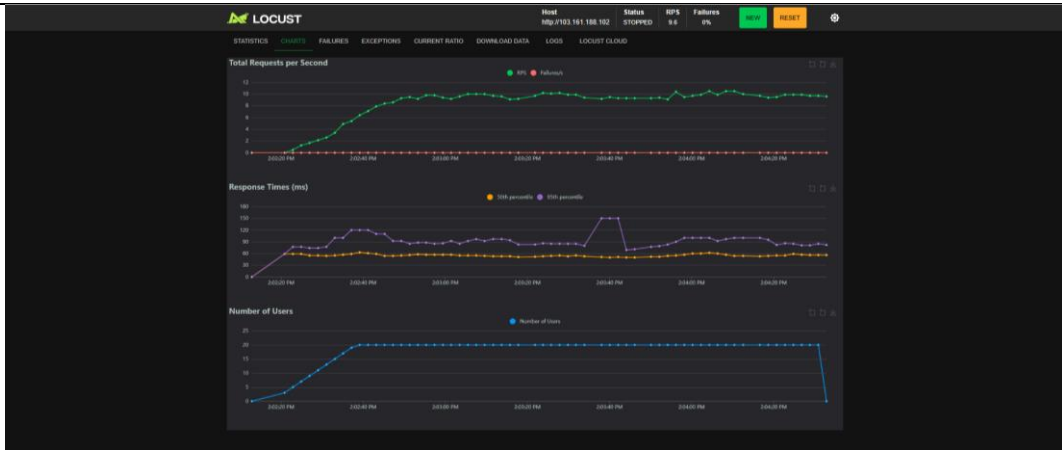
1. Performance Testing

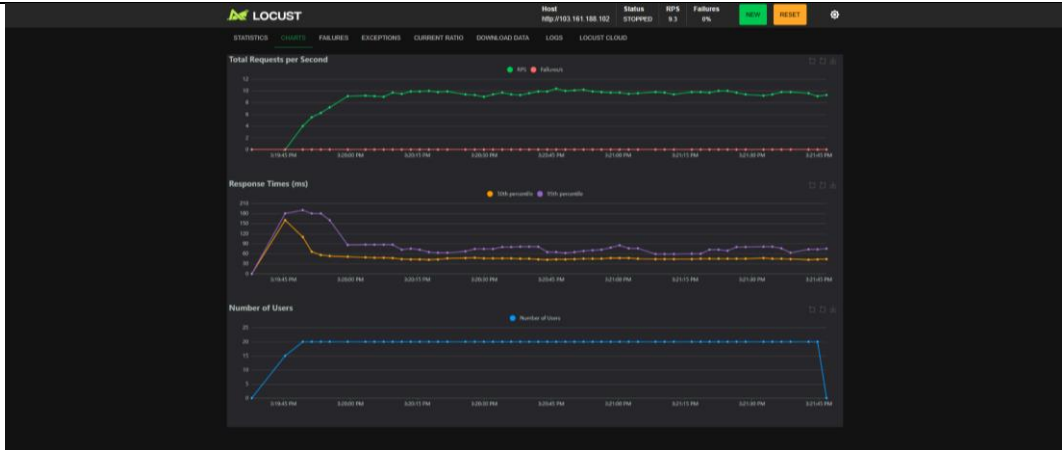
Tabel 4.30 Performance Testing

Komponen	Rincian	
Tes Case ID	TC01-NF01	
Type	Performance Testing	
Judul	Sistem harus mampu menangani peningkatan jumlah permintaan secara bertahap tanpa mengalami penurunan kinerja yang signifikan.	
Tujuan	Memastikan sistem dapat beroperasi dengan baik di bawah berbagai beban kerja tanpa penurunan performa.	
Lingkungan	Server pengujian menggunakan spesifikasi <i>hardware</i> dan <i>software</i> yang menyerupai lingkungan produksi, termasuk CPU, RAM, disk, sistem operasi, dan konfigurasi jaringan. <i>Database</i> serta web server dikonfigurasi identik dengan produksi, sementara koneksi jaringan disimulasikan dengan kecepatan, <i>latensi</i> , dan jitter realistis untuk mencerminkan kondisi pengguna akhir.	
Alat	Locust	
Metode	1. Simulasi perilaku pengguna dengan <i>Skenario</i> umum seperti <i>login</i> , melakukan absensi, mengajukan cuti, mengajukan lembur, dan menambahkan tugas tugas. 2. Uji <i>load</i> testing dengan jumlah pengguna bertahap sebanyak 30 <i>user</i> 3. Mencatat anomali atau penurunan performa.	
Ekspektasi	Sistem dapat merespons permintaan dalam waktu yang konsisten, dengan konsumsi sumber daya yang efisien dan tanpa <i>crash</i> saat mencapai beban puncak.	
Hasil Uji		
ID	LOT01	
Nama Pengujian	Load Test – Endpoint <i>Login</i>	
Deskripsi Pengujian	Pengujian ini dilakukan pada <i>endpoint login</i> untuk mengukur waktu <i>respons</i> , jumlah permintaan yang di proses, serta kestabilan sistem saat menerima beban pengguna yang tinggi secara bersamaan.	
Method	POST	
Skenario Uji	Total Users	20
	Ramp Up	5

	Time (durasi)	2 menit
Response Time (ms)	Min (ms)	42
	Average (ms)	81.33
	Max (ms)	365
	95%ile (ms)	150
Requests	Total Requests	1542
	Success Requests	1542
	Failed Requests	0
Error rate (%)	0%	
Chart Pengujian		
Hasil Uji		
ID	LOT02	
Nama Pengujian	Load Test – Endpoint Absensi	
Deskripsi Pengujian	Pengujian ini dilakukan pada <i>endpoint</i> melakukan absensi untuk mengukur waktu respons, jumlah permintaan yang di proses, serta kestabilan sistem saat menerima beban pengguna yang tinggi secara bersamaan.	
Method	POST	
Skenario Uji	Total Users	20
	Ramp Up	5
	Time (durasi)	2 menit
	Min (ms)	30

Response Time (ms)	Average (ms)	53.98
	Max (ms)	694
	95%ile (ms)	95
Requests	Total Requests	1268
	Success Requests	1268
	Failed Requests	0
Error rate (%)	0%	
Chart Pengujian		
Hasil Uji		
ID	LOT03	
Nama Pengujian	Load Test – Endpoint Mengajukan Cuti	
Deskripsi Pengujian	Pengujian ini dilakukan pada <i>endpoint</i> mengajukan cuti untuk mengukur waktu respons, jumlah permintaan yang di proses, serta kestabilan sistem saat menerima beban pengguna yang tinggi secara bersamaan.	
Method	POST	
Skenario Uji	Total Users	20
	Ramp Up	5
	Time (durasi)	2 menit
Response Time (ms)	Min (ms)	30
	Average (ms)	59.02
	Max (ms)	341

	95%ile (ms)	92
Requests	Total Requests	1201
	Success Requests	1201
	Failed Requests	0
Error rate (%)	0%	
Chart Pengujian		
Hasil Uji		
ID	LOT04	
Nama Pengujian	Load Test – Endpoint Mengajukan Lembur	
Deskripsi Pengujian	Pengujian ini dilakukan pada <i>endpoint</i> mengajukan lembur untuk mengukur waktu respons, jumlah permintaan yang di proses, serta kestabilan sistem saat menerima beban pengguna yang tinggi secara bersamaan.	
Method	POST	
Skenario Uji	Total Users	20
	Ramp Up	5
	Time (durasi)	2 menit
Response Time (ms)	Min (ms)	29
	Average (ms)	59.01
	Max (ms)	531
	95%ile (ms)	100
Requests	Total Requests	1098

	Success Requests	1098
	Failed Requests	0
Error rate (%)	0%	
Chart Pengujian		
Hasil Uji		
ID	LOT06	
Nama Pengujian	Load Test – Endpoint Menambahkan Tugas	
Deskripsi Pengujian	Pengujian ini dilakukan pada <i>endpoint</i> menambahkan tugas untuk mengukur waktu respons, jumlah permintaan yang di proses, serta kestabilan sistem saat menerima beban pengguna yang tinggi secara bersamaan.	
Method	POST	
Skenario Uji	Total Users	20
	Ramp Up	5
	Time (durasi)	2 menit
Response Time (ms)	Min (ms)	49
	Average (ms)	300.62
	Max (ms)	2094
	95%ile (ms)	690
Requests	Total Requests	1082
	Success Requests	1082
	Failed Requests	0

Error rate (%)	0%
Chart Pengujian	

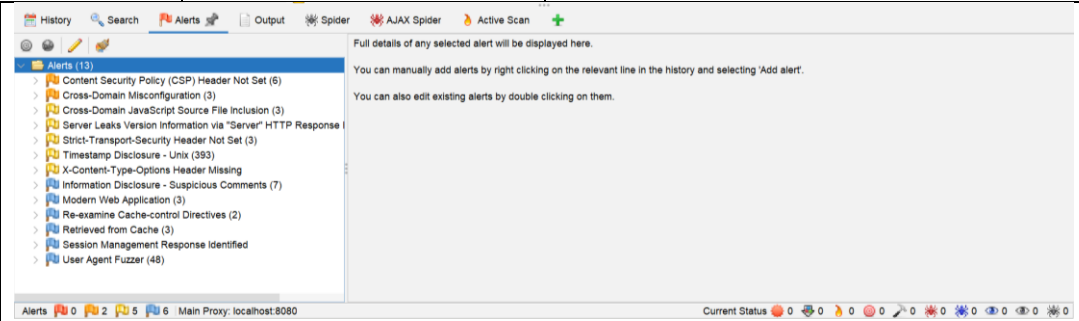
2. Security Testing

Tabel 4.31 Security Testing

Komponen		Rincian	
Tes <i>Case</i> ID		TC02-NF02	
<i>Type</i>		Security Testing	
Tujuan		Memastikan sistem terlindungi dari ancaman keamanan seperti injeksi SQL, <i>cross-site scripting</i> , dan akses tidak sah.	
Lingkungan		Pengujian pada server pengembangan dengan akses terbatas untuk mencegah kebocoran data.	
Alat		OWASP ZAP	
Metode		1. Melakukan pengujian penetrasi (<i>penetration testing</i>) terhadap fitur aplikasi. 2. Mengidentifikasi kerentanan dan melakukan <i>eksploitasi</i> terkontrol. 3. Menganalisis keamanan data saat transit dan penyimpanan.	
Ekspektasi		Sistem bebas dari kerentanan utama, data terenkripsi dengan baik, dan akses hanya dapat dilakukan oleh pihak yang berwenang.	
Hasil Uji			
Jumlah Alert		13	
No	Temuan	Tingkat Risiko	Penjelasan

1	<i>Content Security Policy (CSP) Header Not Set</i>	Menengah	Tidak adanya <i>Content-Security-Policy</i> meningkatkan risiko serangan <i>Cross Site Scripting</i> (XSS) dan injeksi data. CSP berfungsi sebagai lapisan keamanan tambahan untuk mendeklarasikan sumber konten yang diizinkan seperti JavaScript, CSS, <i>fonts</i> , dan <i>images</i> .
2	<i>Cross-Domain Misconfiguration (CORS)</i>	Menengah	Konfigurasi CORS yang tidak tepat mengizinkan permintaan lintas domain dari pihak ketiga arbitrary. Ini berpotensi memungkinkan penyerang mengakses data yang tersedia secara tidak terautentikasi.
3	<i>Cross-Domain JavaScript Source File Inclusion</i>	Menengah	Aplikasi memuat file JavaScript dari domain pihak ketiga (unpkg.com) yang dapat menimbulkan risiko keamanan jika sumber tidak terpercaya atau dapat dikontrol oleh <i>end user</i> .
4	<i>Server Leaks Version Information via "Server" HTTP Response Header</i>	Rendah	<i>Server</i> mengungkapkan informasi versi melalui header HTTP response. Informasi ini dapat membantu penyerang mengidentifikasi kerentanan spesifik yang terkait dengan versi <i>server</i> tersebut.
5	<i>Strict-Transport-Security Header Not Set</i>	Rendah	Tidak adanya HTTP <i>Strict Transport Security</i> (HSTS) <i>header</i> dapat memungkinkan koneksi tidak aman. HSTS memastikan browser hanya berinteraksi menggunakan koneksi HTTPS yang aman.

6	<i>Timestamp Disclosure - Unix</i>	Rendah	Aplikasi mengungkapkan timestamp Unix yang berpotensi digunakan untuk mengidentifikasi pola atau informasi sensitif yang dapat dieksploitasi.
7	<i>X-Content-Type-Options Header Missing</i>	Rendah	<i>Header X-Content-Type-Options</i> tidak disetel ke 'nosniff', memungkinkan MIME-sniffing pada browser versi lama yang dapat menyebabkan interpretasi konten berbahaya.
8	<i>Suspicious Comments</i>	<i>Informational</i>	Response mengandung komentar mencurigakan yang berpotensi memberikan informasi kepada penyerang tentang struktur atau konfigurasi aplikasi.
9	<i>Modern Web Application</i>	<i>Informational</i>	Aplikasi teridentifikasi sebagai modern web application. Untuk eksplorasi otomatis lebih efektif disarankan menggunakan Ajax Spider daripada standard spider.
10	<i>Incomplete or No Cache-control Header Set</i>	<i>Informational</i>	<i>Cache-control header</i> tidak disetel dengan benar, memungkinkan browser dan proxy menyimpan konten. Perlu review untuk memastikan tidak ada konten sensitif yang di-cache.
11	<i>Retrieved from Cache</i>	<i>Informational</i>	Konten diambil dari <i>shared cache</i> yang berpotensi membocorkan informasi sensitif atau sesi user jika konfigurasi <i>caching server</i> tidak tepat.
12	<i>Session Management Response Identified</i>	<i>Informational</i>	Response teridentifikasi mengandung token manajemen sesi. Ini adalah <i>alert</i>

			informasional untuk <i>auto-detect session management method</i> , bukan <i>vulnerability</i> .
13	<i>User Agent Fuzzer</i>	<i>Informational</i>	Pemeriksaan perbedaan <i>response</i> berdasarkan <i>User Agent</i> yang di-fuzz untuk mendeteksi perlakuan berbeda terhadap <i>mobile browser</i> atau <i>search engine crawler</i> .
Screenshot			

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, pengembangan, dan pengujian aplikasi Human Resource Information System di Perusahaan Penyedia Jasa Teknologi Informasi (IT), maka dapat disimpulkan bahwa tujuan proyek akhir ini telah tercapai. Kesimpulan utama yang diperoleh adalah sebagai berikut:

1. Sistem tugas dan absensi karyawan berbasis GPS dan *video* berhasil dirancang dan dibangun, sehingga karyawan dapat melakukan presensi di kantor maupun penugasan di luar kantor dengan bukti lokasi dan video yang akurat dan dapat divalidasi oleh bagian HR.
2. Sistem absensi yang terintegrasi dengan perhitungan potongan pajak berhasil dikembangkan, sehingga pencatatan kehadiran, perhitungan potongan gaji, serta rekapitulasi data dapat dilakukan secara otomatis, lebih efisien, dan dapat dijadikan dasar yang valid dalam proses penggajian.
3. Fitur pengingat servis kendaraan berkala berhasil diimplementasikan, dengan mekanisme notifikasi otomatis melalui email sehingga membantu perusahaan mengurangi risiko keterlambatan servis kendaraan operasional.
4. Sistem pengajuan dan verifikasi cuti serta lembur berhasil dibangun, sehingga proses permohonan, pengecekan, dan persetujuan dapat dilakukan secara lebih terstruktur, terdokumentasi, dan dapat diverifikasi oleh manajer secara langsung.

Dengan terealisasinya keempat tujuan tersebut, sistem HRIS yang dikembangkan mampu mendukung proses administrasi perusahaan secara lebih efektif, akurat, dan terotomatisasi.

5.2 Saran

Berdasarkan hasil pengembangan dan pengujian aplikasi HRIS yang telah dilakukan, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan sistem di masa mendatang:

1. Sistem absensi dapat dilengkapi dengan fitur pengingat otomatis apabila pengguna belum melakukan absensi masuk atau keluar pada waktu yang telah ditentukan, sehingga dapat meminimalkan keterlambatan atau kelalaian dalam melakukan absensi.
2. Untuk pengguna yang login melalui lebih dari satu perangkat, disarankan agar sistem menyediakan mekanisme penyimpanan *device token* FCM secara terpisah di *database*, sehingga seluruh perangkat yang digunakan dapat menerima notifikasi sistem secara konsisten.
3. Perlu ditambahkan fitur pengajuan lupa absensi, sehingga ketika karyawan lupa melakukan absensi masuk atau keluar, mereka dapat membuat permohonan koreksi absensi yang selanjutnya dapat diverifikasi atau disetujui oleh pihak HR.
4. Fitur penugasan dapat dikembangkan agar satu tugas dapat diberikan kepada beberapa pengguna sekaligus, sehingga proses penugasan lebih fleksibel dan efisien untuk pekerjaan lapangan yang melibatkan banyak personel.

DAFTAR PUSTAKA

- Abdul Rahman Ismail. (2025). Penerapan Metode Agile Pada Perancangan Sistem Informasi Pengajuan Nomor Surat di Pemerintahan Desa. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 7(2), 285–290.
- Abdullah, M. S., Shaddiq, S., & Sudirman, S. (2024). Pengaruh Dan Tantangan Dalam Penggunaan Aplikasi Human Resource Information (HRIS) Terhadap Produktivitas Pegawai. *Journal of Management Branding*, 1(1), 1–9. <https://doi.org/10.71326/jmb.v1i1.5>
- Ainah, S., Nur Chusnul Khotimah, Y., Maharani, A., Noor Kamala Sari, N., & Handrianus Pranatawijaya, V. (2024). Implementasi Framework Flutter Untuk Pengembangan Aplikasi Restoran Dengan Penerapan Api Chatgpt. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(3), 3802–3809. <https://doi.org/10.36040/jati.v8i3.9804>
- ANDRY ANDARU. (2020). Pengertian *Database* Secara Umum. *Proceedings of the 1970 25th Annual Conference on Computers and Crisis: How Computers Are Shaping Our Future, ACM 1970*, 1–7.
- Ariffin, M., & Sitabuana, T. H. (2022). Sistem Perpajakan Di Indonesia. *Serina IV Untar*, 28, 523–534.
- Aris Sunandar, & Henri Septanto. (2023). Perancangan Human Resource Information System Berbasis Web Di Pt the Master Steel. *Jurnal Informatika Dan Tekonologi Komputer (JITEK)*, 3(2), 164–172. <https://doi.org/10.55606/jitek.v3i2.1846>
- Bailoul, Y., Anoir, L., Khaldi, M., & Khaldi, M. (2025). Analysis of the Existing E-Mailing Management Tools and Systems: Methodology for the Study of the Existing Situation. *International Journal of Advanced Research*, 13(05), 872–877. <https://doi.org/10.21474/ijar01/20971>
- Desma Aipina, & Harry Witriyono. (2022). Pemanfaatan Framework Laravel Dan Framework Bootstrap Pada Pembangunan Aplikasi Penjualan Hijab Berbasis Web. *Jurnal Media Infotama*, 18(1), 36–42.
- Edhy Permata, R. (2019). Human Resources Information System (Hris) Di Pt. Sarmiento Parakantja Timber Berbasis Web. *Jurnal Penelitian Dosen Fikom (UNDA)*, 10(1).

- Ery Hartati. (2022). Sistem Informasi Transaksi Gudang Berbasis Website Pada Cv. Asyura. *Klik - Jurnal Ilmu Komputer*, 3(1), 12–18. <https://doi.org/10.56869/klik.v3i1.323>
- Hikmah, N., Suradika, A., & Ahmad Gunadi, R. A. (2021). Metode Agile Untuk Meningkatkan Kreativitas Guru Melalui Berbagi Pengetahuan (Knowledge Sharing) (Studi Kasus: Sdn Cipulir 03 Kebayoran Lama, Jakarta. *Instruksional*, 3(1), 30. <https://doi.org/10.24853/instruksional.3.1.30-39>
- Ii, B. A. B., & Pustaka, T. (2015). *Politeknik Negeri Sriwijaya*. 7–23.
- Ilyas, M., & Sari, R. (2024). Rancang Bangun Sistem Informasi Manajemen Karyawan Berbasis Web. *Reputasi: Jurnal Rekayasa Perangkat Lunak*, 5(1), 62–68. <https://doi.org/10.31294/reputasi.v5i1.3318>
- Jovanka, R. (2025). *Perancangan HRIS Berbasis Website untuk Meningkatkan Efektivitas Management pada Sebuah Perusahaan Swasta di Kota Batam Pendahuluan*. 24, 151–160.
- M Sahyudi, & Erliyan Redy Susanto. (2025). Analisis Implementasi Sistem Keamanan Basis Data Berbasis Role-Based Access Control (RBAC) pada Aplikasi Enterprise Resource Planning. *SATESI: Jurnal Sains Teknologi Dan Sistem Informasi*, 5(1), 105–116. <https://doi.org/10.54259/satesi.v5i1.3997>
- Mirji, H., & Kapoor, R. (2022). Human Resource Information System (Hris)-Scope, Challenges and Benefits. *International Journal of Engineering Science Invention Research & Development*, 2(August).
- Mulang, H. (2023). Pengaruh Kompetensi dan Motivasi Terhadap Kinerja Pegawai. *Jesya*, 6(1), 38–51. <https://doi.org/10.36778/jesya.v6i1.894>
- Nasich, A. K., Wicaksono, S. A., & Saputra, M. C. (2025). *Implementasi Role Based Access Control (RBAC) dalam Sistem Informasi Manajemen Pelanggan dan Pembayaran Air Berbasis Web (studi pada PT Tirta Wangi Sejahtera)*. 9(9), 1–9.
- Nisa, A. Z., & Putra, R. S. (2022). 478-Article Text-1423-1-10-20220713. *Mizania: Jurnal Ekonomi Dan Akuntansi*, 2(1), 175–189.
- Nusantoko, Y. R., & Prapanca, A. (2025). Pengembangan Aplikasi Human Resource Information System Berbasis *Mobile* Dengan Absensi Wajah Menggunakan Metode CNN Pada BMT Bahtera Pekalongan. *Journal of Informatics and Computer Science*, 06, 1020–1038.

- Odun-Ayo, I., & Aina, A. (2020). *Development of a Cloud-Based Payroll Management System*. 1–7.
- Panjaitan, E. H. H. (2023). Implementing Human Resource Information System (HRIS) for Efficient Human Resource Management. *International Journal of Science and Society*, 5(2), 128–139. <https://doi.org/10.54783/ijssoc.v5i2.676>
- Putra, A. B., & Nita, S. (2019). Perancangan dan Pembangunan Sistem Informasi E-Learning Berbasis Web (Studi Kasus Pada Madrasah Aliyah Kare Madiun). *Seminar Nasional Teknologi Informasi Dan Komunikasi 2019*, 1(1), 81–85.
- Raudha Allisya, D., & Sari, R. E. (2023). *Rancang Bangun Aplikasi Human Resource Information System pada PT. Sistem Teknologi Regularisasi Elektronik Aktivitas Millenial Design and Build the Human Resource Information System Application at PT. Millennial Activity Electronic Regularization Technol*. 1(3), 834.
- Rinaldiansyah Nugraha, R., & Purnama, G. (2023). Web-Based Payroll Application Development in Higher Education Institutions (Case Study at XYZ University). *Jtsi*, 4(2), 335–345.
- Rioni, Y. S., & Syauqi, T. radhifan. (2020). Analisis Peningkatan Kepatuhan Wajib Pajak Dalam Pembuatan NPWP UKM di Kebun Lada Kecamatan Hinai Kabupaten Langkat. *Jurnal Perpajakan*, 1(2), 28–37. <http://jurnal.pancabudi.ac.id/index.php/jurnalperpajakan/article/view/805>
- Seran, N., Mau, S. D. I., & Ayu, M. P. (2023). Implementasi Absensi Online Berbasis Sistem GPS Pada Pegawai Lapangan DP5A. *Jurnal Teknik Informatika Dan Sistem Informasi (Jatsi)*, 10(4), 183–193.
- Suwarno, S., & Jesselyn, J. (2024). Pengembangan Aplikasi Human Resource Information System (Hris) Pada Pt. Xyz. *Computer Based Information System Journal*, 12(2), 40–50. <https://doi.org/10.33884/cbis.v12i2.9392>
- Unconfirmed 662121.crdownload*. (n.d.).
- Wijoyo, dkk. hadion. (2022). Sistem Informasi Manajemen: Tujuan Sistem Informasi Manajemen. In CV. Pena Persada (Issue April).
- Yunianti, D. S., & SE.MM, H. S. (2025). Pengaruh Absensi Dan Tunjangan Kerja Terhadap Kinerja Pegawai Pada Badan Kepegawaian Dan Pengembangan Sumber Daya Manusia (Bkpsdm) Kota Tangerang Selatan. *Jurnal Manajemen Dan Bisnis*, 7(1), 39–47. <https://doi.org/10.47080/jmb.v7i1.3641>

Yuricha, I. K. P. (2023). The Implementation of Role Based Access Control in a Cloud-Based Supply Chain Management System. *Donesian Journal of Machine Learning and Computer Science*, 3(2), 339–348.
<https://journal.irpi.or.id/index.php/malcom/article/view/1259>

DAFTAR LAMPIRAN

Lampiran A. Dokumen Proses Pengumpulan Requirement

Dokumen Work Process

<https://drive.google.com/drive/folders/1ODiTfLCLQgDMxeVaX9XnTNspdCEoR1F9?usp=sharing>

Dokumen Postmen

<https://documenter.getpostman.com/view/33834742/2sB3dTto1h>

Lampiran B. Link Produk

<https://github.com/Greyari/HR>

<https://github.com/HaikalTaufiq/hris>

Lampiran C. Dokumen Pengujian UAT

Performance Testing

https://drive.google.com/drive/folders/114w6FXQVAkYVn_xVJfR1OZschgf2hZW9?usp=sharing

Security Testing

<https://drive.google.com/drive/folders/1C52faYBh6OiDZFOXY5PvOx0idll7IJV5?usp=sharing>

Dokumen User Acceptance Testing (UAT)

<https://drive.google.com/drive/folders/1zmwKL9KbvKjjUsOzDcy9VkiykPwfFovc?usp=sharing>