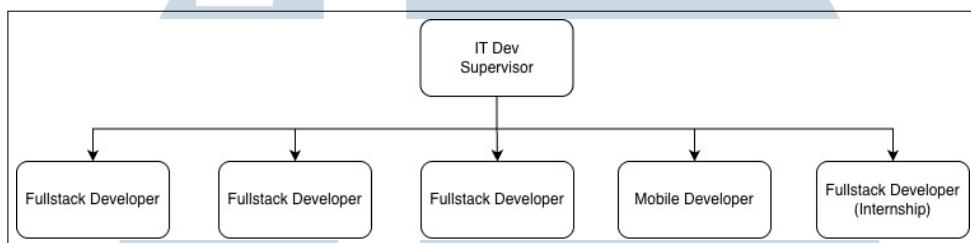


BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi



Gambar 3.1. Kedudukan dan koordinasi

Selama pelaksanaan kerja magang di PT Jala Lintas Media, sesuai pada Gambar 3.1 posisi yang diberikan adalah Fullstack Developer yang bertanggung jawab dalam proses pengembangan webiste HRGO. Kegiatan kerja magang ini dilaksanakan di bawah bimbingan langsung dari Bapak Budi Ramadhan, selaku pembimbing lapangan bagi mahasiswa magang.

3.2 Tugas yang Dilakukan

Selama masa pelaksanaan magang di PT Jala Lintas Media, tanggung jawab yang diberikan adalah proses pengembangan dan implementasi proyek aplikasi internal perusahaan, yaitu HRGO. aplikasi tersebut dikembangkan dengan tujuan untuk meningkatkan efisiensi operasional perusahaan melalui digitalisasi sistem administrasi sumber daya manusia dan pengelolaan dokumen. Selama mengerjakan proyek HRGO, tanggung jawab yang diberikan seperti melakukan pembaruan dan pengembangan fitur untuk meningkatkan kinerja, keamanan, dan kenyamanan penggunaan sistem. Adapun tugas-tugas yang dilakukan meliputi:

1. Pengembangan modul absensi berbasis *face recognition*.

Melakukan peningkatan sistem absensi yang sebelumnya masih berat di sisi *frontend* dengan memindahkan proses verifikasi wajah ke sisi *backend* menggunakan bahasa pemrograman Python. Pendekatan ini membuat proses absensi menjadi lebih ringan, stabil, dan akurat, karena beban komputasi utama kini dijalankan di server.

2. Pengembangan modul pengajuan cuti.

Membangun sistem pengajuan cuti dengan mekanisme dua lapisan persetujuan (*two-layer approval*), di mana permohonan cuti diverifikasi oleh atasan langsung sebelum disetujui oleh pihak HRD. Sistem ini meningkatkan transparansi dan efisiensi dalam pengelolaan izin karyawan.

3. Pengembangan sistem penggajian (payroll system).

Modul penggajian dikembangkan agar terintegrasi langsung dengan data absensi karyawan, sehingga perhitungan gaji, tunjangan, dan potongan dapat dilakukan secara otomatis. Selain itu, fitur *preview payroll* ditambahkan untuk memudahkan pengecekan hasil perhitungan sebelum dilakukan finalisasi.

4. Penyederhanaan navigasi dan perancangan ulang halaman login.

Melakukan perbaikan pada struktur navigasi agar antarmuka lebih intuitif serta merancang ulang halaman login dengan tampilan yang responsif dan autentikasi berbasis email serta kata sandi. Pembaruan ini bertujuan untuk meningkatkan pengalaman pengguna dalam mengakses sistem.

5. Pembuatan API untuk aplikasi mobile.

Membuat *Application Programming Interface* (API) agar aplikasi HRGO dapat diakses melalui platform mobile. API ini dirancang untuk menghubungkan data dari server utama ke aplikasi mobile secara aman dan efisien.

6. Integrasi dan pengujian aplikasi di server produksi.

Setelah seluruh modul dikembangkan, dilakukan proses integrasi ke server PT Jala Lintas Media. Kegiatan ini meliputi pengujian fungsional, optimasi performa sistem, serta memastikan seluruh fitur berjalan dengan baik pada lingkungan produksi.

7. Pengembangan aplikasi Software as a Service (SaaS)

Pengembangan HRGO dilanjutkan ke tahap SaaS untuk memberikan efisiensi yang lebih baik, sekarang HRGO memiliki 2 server produksi untuk pelanggan berbeda yang dimana ini tidak efisien. Pengembangan ke tahap SaaS bertujuan untuk menghemat resource yang dimiliki.

Secara keseluruhan, kegiatan magang ini memberikan pengalaman langsung kepada dalam mengelola proyek pengembangan aplikasi berbasis web di

lingkungan industri profesional. Melalui keterlibatan pada proyek HRGO dan DRMS, dalam kegiatan ini banyak diperoleh pemahaman yang lebih mendalam mengenai integrasi sistem, keamanan data, serta praktik pengembangan perangkat lunak yang efisien dan berorientasi pada kebutuhan pengguna.

3.3 Uraian Pelaksanaan Magang

Periode magang dimulai pada 25 Agustus 2025. Kegiatan utama mencakup pengembangan frontend, pengembangan backend, testing, hingga deployment.

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke-	Pekerjaan yang dilakukan
1	Pengenalan lingkungan kantor, penjelasan alur kerja dan standar pengembangan, pemaparan ruang lingkup proyek HRGO, serta perkenalan dengan tim pengembang dan <i>supervisor</i> .
2	Implementasi halaman login yang aman dan responsif menyiapkan dan menguji API dasar (autentikasi, manajemen pengguna) sebagai fondasi integrasi modul HRGO.
3	Pengembangan API lanjutan dan perancangan modul payroll integrasi awal payroll dengan data kehadiran sebagai dasar perhitungan gaji, tunjangan, dan potongan.
4	Pengembangan modul absensi berbasis <i>face recognition</i> (deteksi, verifikasi, penyimpanan), serta uji fungsional awal.
5	Pengembangan aplikasi lebih lanjut sesuai kebutuhan klien (penyempurnaan alur absensi, penyelarasan aturan payroll, integrasi awal modul cuti dan penyesuaian antarmuka).
6	Pengembangan aplikasi lebih lanjut sesuai kebutuhan klien (penyempurnaan alur absensi, penyelarasan aturan payroll, integrasi awal modul cuti dan penyesuaian antarmuka).
7	Melakukan proses unggah aplikasi HRGO ke server produksi, melakukan pengujian sistem secara menyeluruh, serta mengembangkan fitur pratinjau (<i>preview</i>) pada modul payroll. Selain itu, dilakukan pula penyesuaian fitur dan antarmuka aplikasi sesuai kebutuhan klien serta pengisian data awal pengguna ke dalam sistem.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang (lanjutan)

Minggu Ke-	Pekerjaan yang dilakukan
8	Menyiapkan lingkungan server untuk aplikasi DRMS dan mulai melakukan eksplorasi terhadap struktur serta fungsi utama sistem sebagai tahap awal pengembangan.
9	Melakukan penerapan sistem <i>payment gateway</i> DOKU Wallet pada aplikasi DRMS agar proses transaksi dan pembayaran internal dapat berlangsung secara otomatis, aman, dan sesuai dengan standar perusahaan.
10	Menerapkan sistem penyimpanan dokumen berbasis <i>cloud</i> dengan menggunakan layanan Nextcloud pada aplikasi DRMS, menggantikan metode penyimpanan lokal agar data lebih mudah diakses dan memiliki tingkat keamanan serta ketersediaan yang lebih tinggi.
11	Melakukan pengembangan lanjutan serta optimisasi performa pada aplikasi DRMS, termasuk perbaikan struktur data, efisiensi proses unggah dokumen, dan peningkatan stabilitas sistem secara keseluruhan.
12	Melakukan perancangan awal transformasi aplikasi HRGO menjadi platform Software as a Service (SaaS), termasuk analisis kebutuhan multi-tenant, struktur database berbasis company id, serta penyesuaian arsitektur aplikasi agar mendukung isolasi data antarklien.
13	Menyusun rancangan modul otentikasi dan manajemen pengguna pada skema SaaS dengan basis company id, mencakup pengelompokan hak akses pengguna per perusahaan, alur login, serta mekanisme pembatasan akses data agar setiap akun hanya dapat melihat data pada company id yang dimilikinya.
14	Melakukan pengembangan tahap pertama untuk implementasi struktur multi-tenant berbasis company id pada modul inti HRGO, termasuk penyesuaian model data, penambahan filter pada query, dan konfigurasi controller agar seluruh proses pengambilan data selalu mempertimbangkan company id terkait.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang (lanjutan)

Minggu Ke-	Pekerjaan yang dilakukan
15	Melanjutkan pengembangan untuk implementasi struktur multi-tenant berbasis company id pada modul inti HRGO, termasuk penyesuaian model data, penambahan filter pada query, dan konfigurasi middleware agar seluruh proses pengambilan data selalu mempertimbangkan company id terkait.
16	Mengubah integrasi cloud project DRMS ke server baru, mengubah struktur data penyimpanan pada cloud dengan menggunakan <i>Provisioning API</i> pada webdav.
17	Mengubah integrasi cloud project DRMS ke server baru, mengubah struktur data penyimpanan pada cloud dengan menggunakan <i>Provisioning API</i> pada Webdav.
18	Melanjutkan pengembangan untuk implementasi struktur multi-tenant berbasis company id pada modul inti HRGO, termasuk penyesuaian model data, penambahan filter pada query, dan konfigurasi middleware agar seluruh proses pengambilan data selalu mempertimbangkan company id terkait.

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

3.3.1 Pengembangan Modul Absensi Berbasis Face Recognition

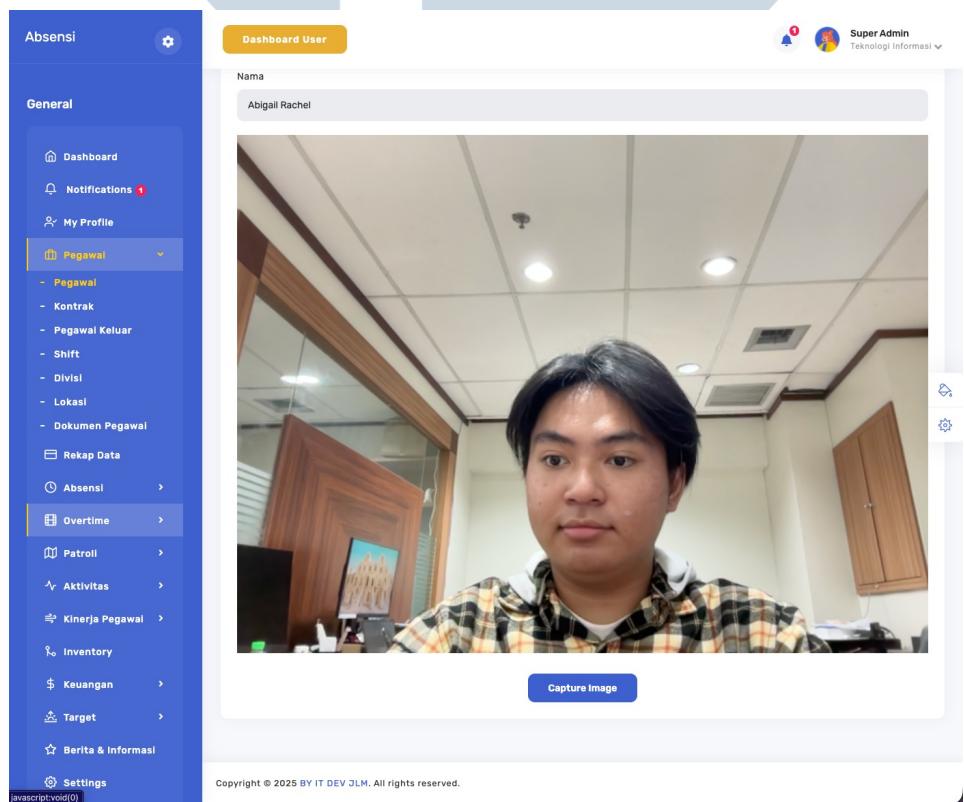
Pengembangan modul absensi berbasis *face recognition* dilakukan pada minggu ke-4 pelaksanaan magang. Tahap awal difokuskan pada analisis performa sistem absensi sebelumnya yang dinilai masih berat karena seluruh proses verifikasi wajah dijalankan di sisi *frontend*. Hal ini menyebabkan penggunaan sumber daya perangkat meningkat serta hasil deteksi wajah tidak selalu konsisten antar pengguna.

Sebagai solusi, dilakukan pengembangan ulang sistem dengan memindahkan proses inferensi *face recognition* ke sisi *backend*. Implementasi dilakukan menggunakan bahasa pemrograman Python dan pustaka pengenalan wajah yang diintegrasikan dengan sistem HRGO. Dengan pendekatan ini, beban komputasi dialihkan ke server sehingga proses absensi menjadi lebih ringan dan stabil. Setelah tahap implementasi selesai, dilakukan pengujian performa untuk

memastikan kecepatan verifikasi dan akurasi deteksi wajah berjalan sesuai standar yang ditetapkan perusahaan. Sistem yang telah diperbarui juga diintegrasikan langsung dengan modul payroll agar hasil kehadiran otomatis memengaruhi perhitungan gaji.

Secara garis besar, alur kerja modul absensi berbasis *face recognition* pada sistem HRGO terdiri dari tiga tahapan utama, yaitu pendaftaran wajah karyawan oleh administrator, Proses absensi mandiri oleh karyawan melalui halaman presensi, Serta pencatatan dan pemantauan data absensi pada dasbor administrator.

Sebelum karyawan dapat melakukan absensi menggunakan biometrik wajah, administrator perlu mendaftarkan data wajah karyawan melalui menu *Pegawai*. Proses dimulai dengan memilih data karyawan yang bersangkutan, kemudian sistem akan menampilkan antarmuka pengambilan gambar yang terhubung dengan kamera perangkat. Karyawan diminta untuk memposisikan wajah tepat di tengah bingkai kamera, lalu administrator menekan tombol *Capture Image* untuk mengambil citra referensi, sebagaimana ditunjukkan pada Gambar 3.2.

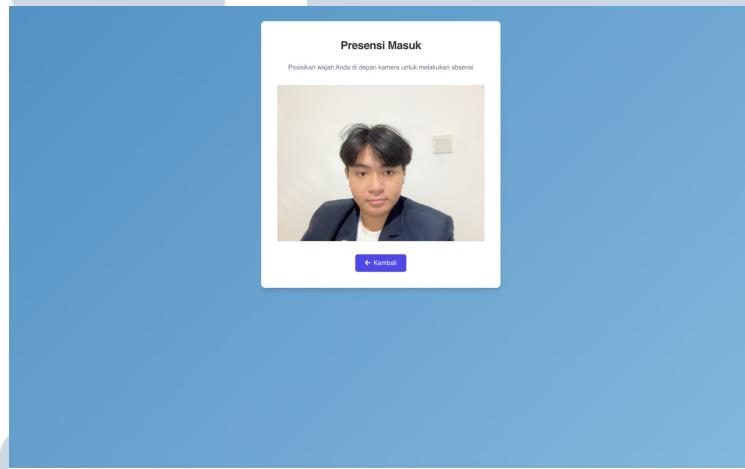


Gambar 3.2. Halaman pendaftaran wajah karyawan oleh admin pada modul HRGO.

Citra wajah yang diambil tidak hanya disimpan sebagai berkas gambar statis, melainkan langsung dikirim ke sisi *backend* HRGO. Gambar tersebut diproses

menggunakan pustaka InsightFace untuk melalui tahapan *face detection, alignment*, dan ekstraksi *embedding* wajah. Vektor *embedding* inilah yang disimpan ke dalam basis data sebagai identitas unik biometrik karyawan dan digunakan sebagai acuan komparasi saat proses absensi berlangsung.

Setelah data wajah terdaftar, karyawan dapat melakukan absensi melalui menu presensi yang tersedia setelah masuk ke dalam sistem. Saat halaman *Presensi Masuk* diakses, sistem secara otomatis mengaktifkan kamera dan menampilkan pratinjau wajah karyawan seperti terlihat pada Gambar 3.3. Karyawan cukup memposisikan wajah di depan kamera, dan sistem akan menangkap citra wajah untuk dikirimkan ke *backend* melalui API, bersamaan dengan data pendukung seperti jenis absensi dan titik koordinat geolokasi.



Gambar 3.3. Tampilan halaman presensi masuk berbasis kamera dan *face recognition*.

Alur logika teknis pada sisi sistem selama proses absensi digambarkan melalui flowchart pada Gambar 3.4. Rincian tahapan proses tersebut adalah sebagai berikut:

1. Akses antarmuka pengguna.

Karyawan mengakses halaman presensi pada aplikasi HRGO dan sistem menampilkan pratinjau kamera secara *real-time*.

2. Pengambilan citra.

Saat proses absensi diinisiasi, sistem mengambil satu bingkai (*frame*) wajah dari kamera sebagai data masukan.

3. Pengiriman permintaan ke server.

Citra wajah, informasi akun pengguna, dan data geolokasi dikirimkan ke server melalui permintaan API (*API Request*).

4. Penerimaan data di backend.

Server menerima data tersebut, menyimpan citra sementara, dan meneruskannya ke modul pemrosesan wajah.

5. Deteksi dan penyesuaian wajah.

Menggunakan pustaka InsightFace, sistem mendeteksi keberadaan wajah dalam citra dan melakukan *alignment* untuk menormalkan posisi wajah guna meningkatkan akurasi pengenalan.

6. Ekstraksi dan pencocokan.

Sistem mengekstraksi *embedding* dari wajah yang dideteksi dan membandingkannya dengan vektor *embedding* referensi yang tersimpan di basis data.

7. Validasi kecocokan wajah.

Jika tingkat kemiripan berada di bawah ambang batas yang ditentukan, sistem menolak absensi. *Backend* akan mengirimkan respon galat (Error/400) dan antarmuka akan menampilkan pesan kegagalan kepada pengguna.

8. Validasi geolokasi.

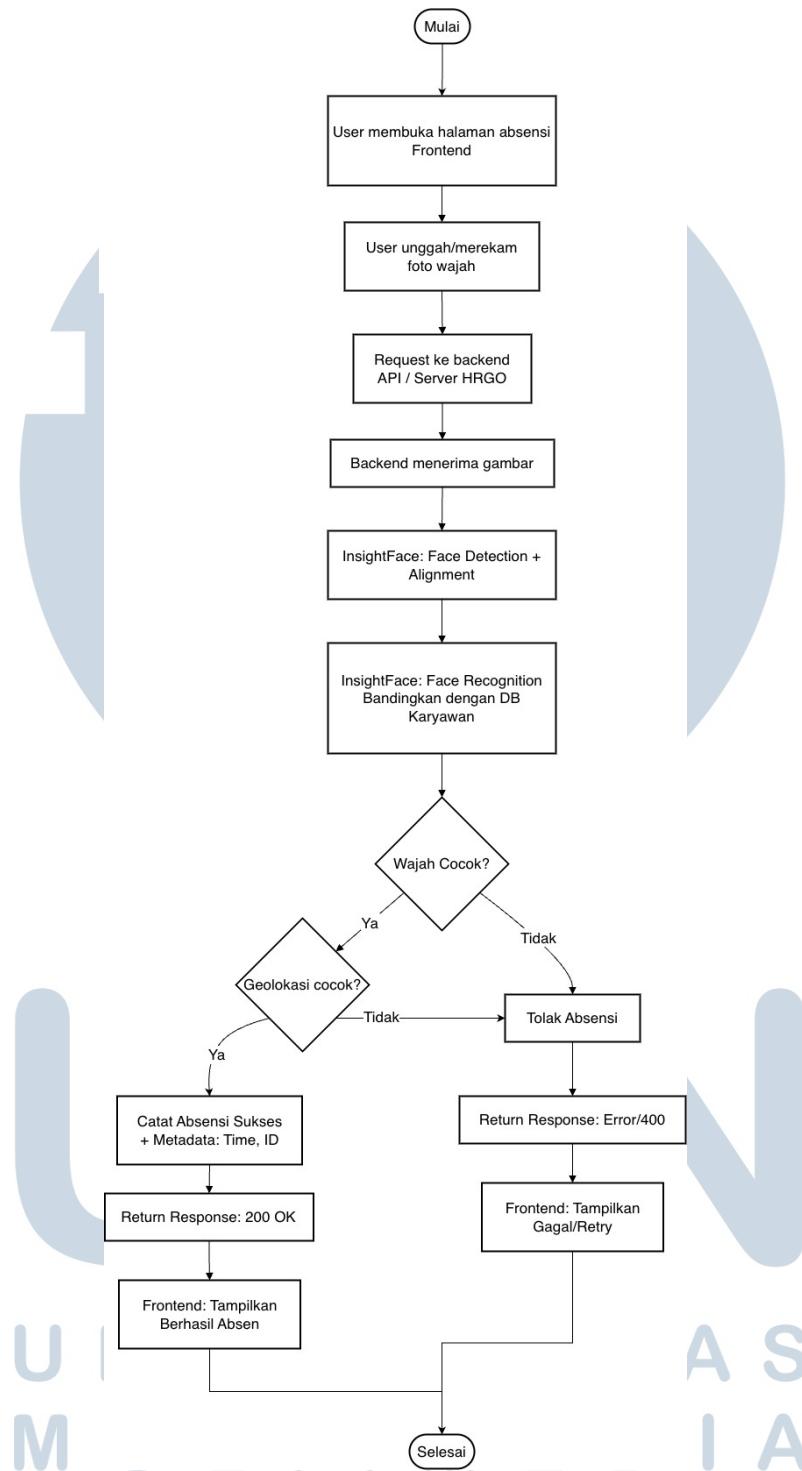
Jika wajah dinyatakan cocok, sistem melanjutkan ke pemeriksaan lokasi. Koordinat perangkat dibandingkan dengan koordinat kantor. Apabila jarak melebihi radius toleransi yang diizinkan, absensi akan ditolak.

9. Pencatatan data.

Apabila validasi wajah dan geolokasi berhasil, sistem mencatat data absensi ke dalam basis data, meliputi ID karyawan, waktu, jenis absensi, foto bukti, dan jarak lokasi. Server mengembalikan kode status sukses (200 OK).

10. Notifikasi berhasil.

Aplikasi menampilkan umpan balik visual kepada karyawan yang menyatakan bahwa proses absensi telah berhasil dicatat.



Gambar 3.4. Flowchart proses absensi berbasis *face recognition* pada HRGO.

Data kehadiran yang valid akan tersimpan secara terpusat dan dapat dipantau oleh administrator melalui modul *Data Absen*. Visualisasi antarmuka modul ini dapat dilihat pada Gambar 3.5. Halaman ini menyajikan tabel rekapitulasi yang

memuat informasi komprehensif, termasuk nama karyawan, tanggal, pembagian waktu kerja (*shift*), jam masuk dan pulang, keterangan keterlambatan, lokasi presensi, serta foto bukti kehadiran.

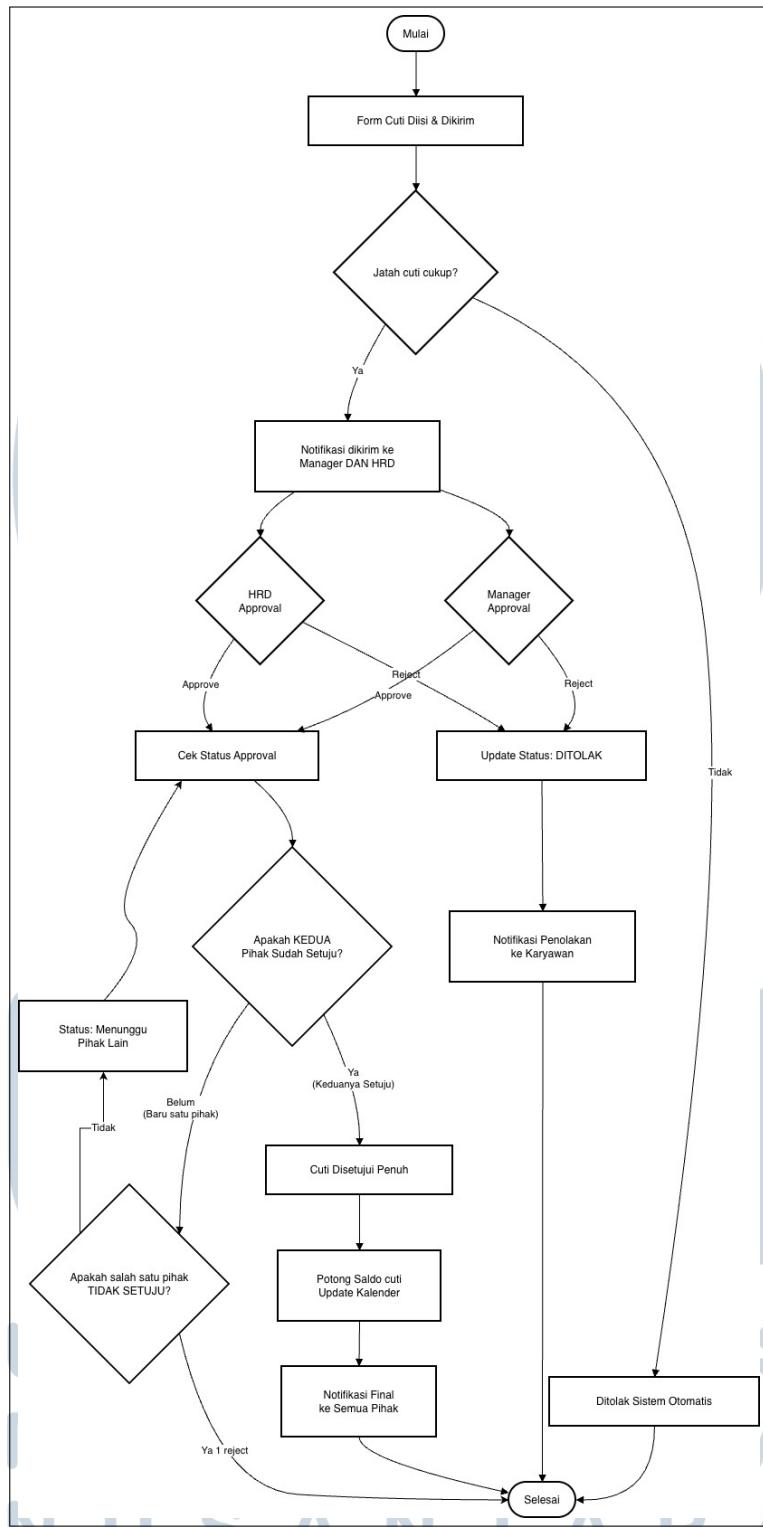
No.	Nama Pegawai	Shift	Tanggal	Jam Masuk	Telat	Lokasi Masuk	Foto Masuk	Keterangan Masuk	Jam Pulang	Pulang Cepat	Lokasi Pulang	Foto Pulang	Keterangan Pulang	Status Absen	Actions
1.	A. Wisnu Nugroho	-	-	-	-	-	-	-	-	-	-	-	-	-	Detail
2.	Abigail Rachel	Office (08:00 - 17:00)	2025-12-03	10:41	2 Jam 41 Menit	14071 Meter		12:56	12:56	00 Menit	1184687 Meter		Massa	Edit Delete	
3.	Adel Nugraha	Office (08:00 - 17:00)	2025-12-03	-	-	-	-	-	-	-	-	-	-	Edit Delete	
4.	Alya Amanda	Office (08:00 - 17:00)	2025-12-03	-	-	-	-	-	-	-	-	-	-	Edit Delete	
5.	Anindya Ramadhan	Office (08:00 - 17:00)	2025-12-03	-	-	-	-	-	-	-	-	-	-	Edit Delete	
6.	Azizi Asadel	Office (08:00 - 17:00)	2025-12-03	-	-	-	-	-	-	-	-	-	-	Edit Delete	
7.	Callista Alilia	Office (08:00 - 17:00)	2025-12-03	-	-	-	-	-	-	-	-	-	-	Edit Delete	

Gambar 3.5. Dashboard *Data Absen* untuk pemantauan kehadiran karyawan.

Administrator dilengkapi dengan fitur pencarian dan penyaringan data berdasarkan nama atau periode waktu tertentu untuk mempermudah pengawasan. Selain itu, data yang tersimpan pada modul ini terintegrasi secara otomatis dengan modul penggajian (*payroll*), sehingga rekapitulasi kehadiran dapat langsung digunakan sebagai dasar perhitungan gaji karyawan tanpa perlu input ulang data secara manual.

3.3.2 Pengembangan Modul Pengajuan Cuti

Pengembangan modul pengajuan cuti dilakukan pada minggu ke-5 bersamaan dengan tahap penyempurnaan antarmuka pengguna HRGO. penyempurnaan alur pengajuan cuti dilakukan dengan merancang sistem dua lapisan persetujuan (*two-layer approval*), di mana setiap permohonan cuti diverifikasi oleh atasan langsung dan pihak HRD.



Gambar 3.6. Flowchart modul pengajuan cuti

Gambar 3.6 menggambarkan tahapan ini meliputi pembuatan form pengajuan cuti, daftar riwayat pengajuan, halaman verifikasi untuk atasan, dan

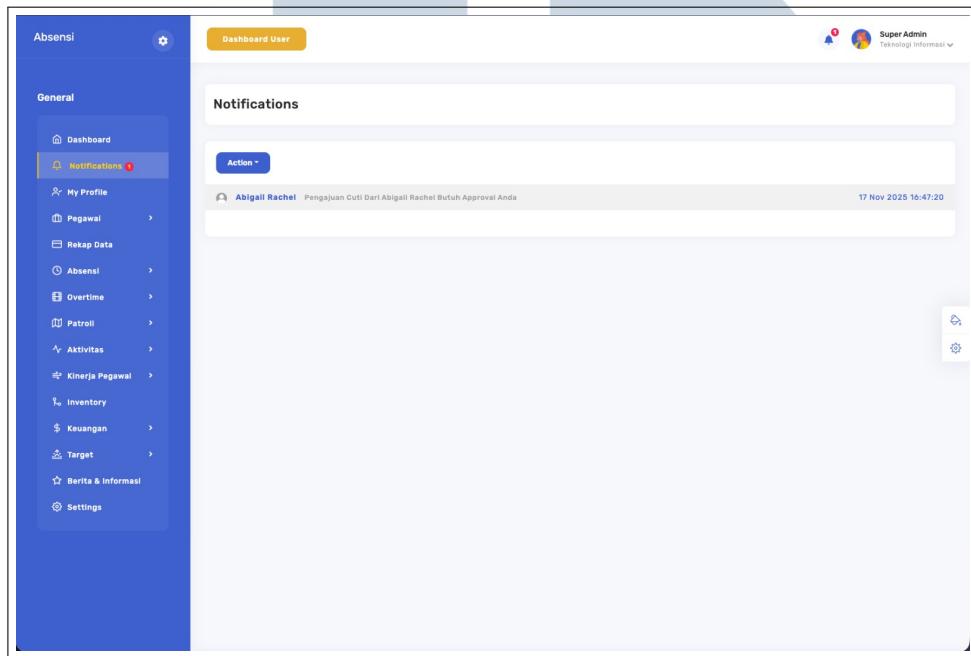
dashboard HRD untuk menyetujui atau menolak permohonan. Seluruh proses pengajuan disimpan dalam basis data yang terintegrasi, sehingga riwayat cuti dapat dilacak dengan mudah. Pengujian dilakukan untuk memastikan status pengajuan dan notifikasi berfungsi sesuai logika persetujuan yang telah ditentukan.

Gambar 3.7. Gambar formulir pengajuan cuti

Pada modul pengajuan cuti, alur dimulai ketika karyawan mengakses halaman permintaan cuti melalui antarmuka web pada Gambar 3.7. Karyawan mengisi formulir yang berisi jenis cuti, tanggal mulai, tanggal akhir, alasan pengajuan, serta bukti pendukung dalam bentuk berkas gambar jika diperlukan. Sistem kemudian memeriksa kelengkapan tanggal pengajuan; apabila salah satu dari tanggal mulai atau tanggal akhir tidak diisi, sistem akan menyamakan nilai keduanya sehingga pengajuan tetap memiliki rentang tanggal yang valid. Setelah itu, sistem membentuk deret tanggal harian berdasarkan rentang yang diajukan. Setiap tanggal dalam rentang tersebut akan diproses sebagai satu entri cuti yang berdiri sendiri, sehingga pengajuan cuti beberapa hari berturut-turut tersimpan sebagai beberapa baris data, masing-masing dengan tanggal dan status tersendiri.

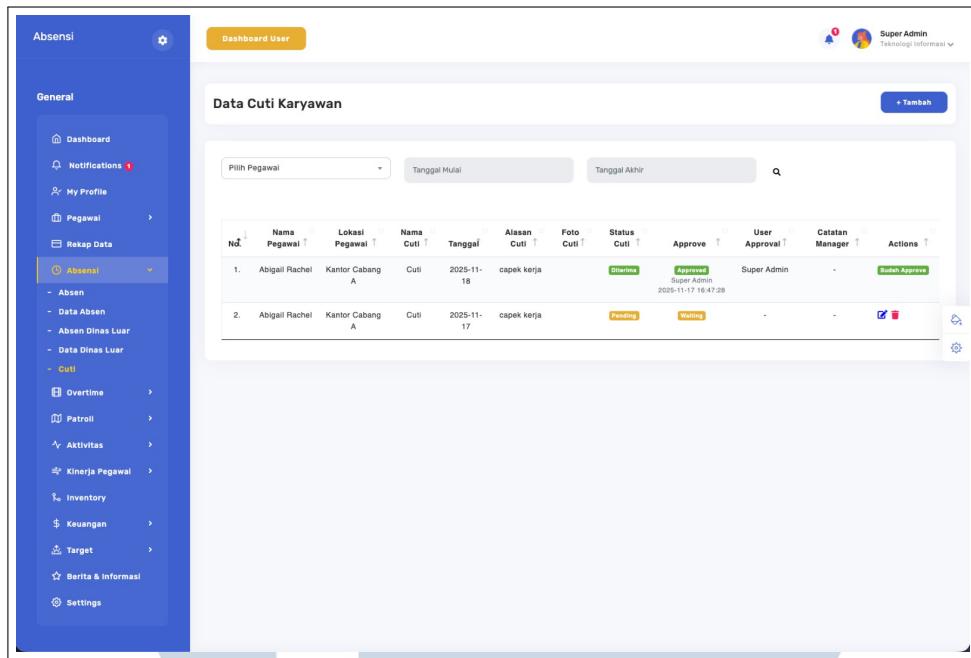
Sebelum data disimpan, sistem melakukan validasi terhadap input yang diberikan karyawan, seperti memastikan nama jenis cuti terisi, tanggal pengajuan valid, serta status awal pengajuan ditetapkan sebagai *Pending*. Apabila karyawan melampirkan bukti berupa foto atau dokumen, berkas tersebut diunggah ke penyimpanan server dan lokasi penyimpanannya direkam bersama data cuti. Setiap

entri cuti juga diberi identitas karyawan dan lokasi kerja sesuai informasi pada akun yang sedang masuk. Setelah seluruh entri pada rentang tanggal berhasil dibuat, karyawan diarahkan kembali ke halaman riwayat cuti dengan notifikasi bahwa pengajuan telah berhasil disimpan dan menunggu persetujuan.



Gambar 3.8. Gambar notifikasi pengajuan cuti

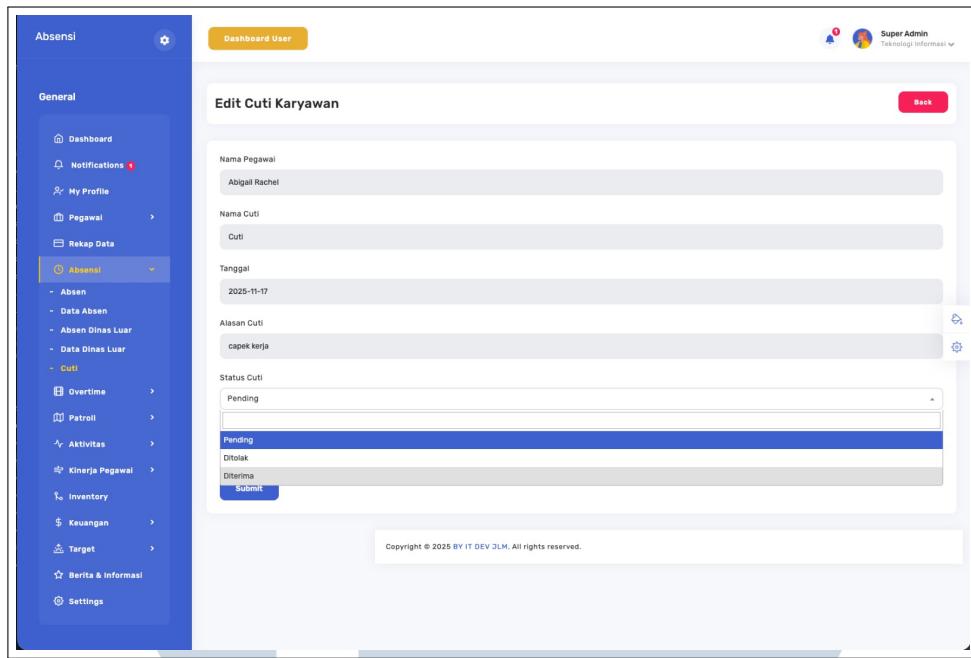
Gambar 3.8 memperlihatkan distribusi notifikasi kepada pihak yang berwenang menyetujui cuti. Sistem secara otomatis mengumpulkan daftar pengguna yang memiliki peran sebagai pengelola, seperti admin, manajer, kepala cabang, serta peran lain yang ditetapkan perusahaan sebagai atasan atau pengelola sumber daya manusia. Untuk kepala cabang, sistem membatasi cakupan hanya pada karyawan di lokasi yang sama, sedangkan untuk manajer, sistem membatasi hanya pada karyawan dengan jabatan yang berada di bawah tanggung jawabnya. Setiap pihak yang termasuk dalam daftar approver akan menerima pesan pemberitahuan bahwa terdapat pengajuan cuti yang membutuhkan tindakan, beserta tautan langsung menuju halaman data cuti yang relevan. Selain melalui tampilan di dalam aplikasi, notifikasi juga disalurkan melalui mekanisme internal sistem pemberitahuan, sehingga approver dapat dengan mudah memonitor pengajuan baru.



Gambar 3.9. Gambar halaman pengajuan cuti

Pada sisi pengelola, proses persetujuan dilakukan melalui halaman khusus data cuti karyawan seperti pada Gambar 3.9. Ketika seorang pengelola membuka halaman ini, sistem terlebih dahulu menerapkan penyaringan berdasarkan peran dan kewenangannya. Kepala cabang hanya dapat melihat pengajuan dari karyawan yang bekerja di lokasi yang sama, sedangkan manajer hanya dapat mengakses pengajuan dari karyawan yang berada pada jabatan di bawah struktur yang dikelolanya. Setelah memilih salah satu pengajuan, pengelola diarahkan ke halaman detail cuti seperti pada Gambar 3.10 yang menampilkan informasi lengkap, meliputi jenis cuti, tanggal, alasan, status terkini, serta kolom catatan untuk diisi oleh pihak yang memproses pengajuan.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.10. Gambar halaman persetujuan pengajuan cuti

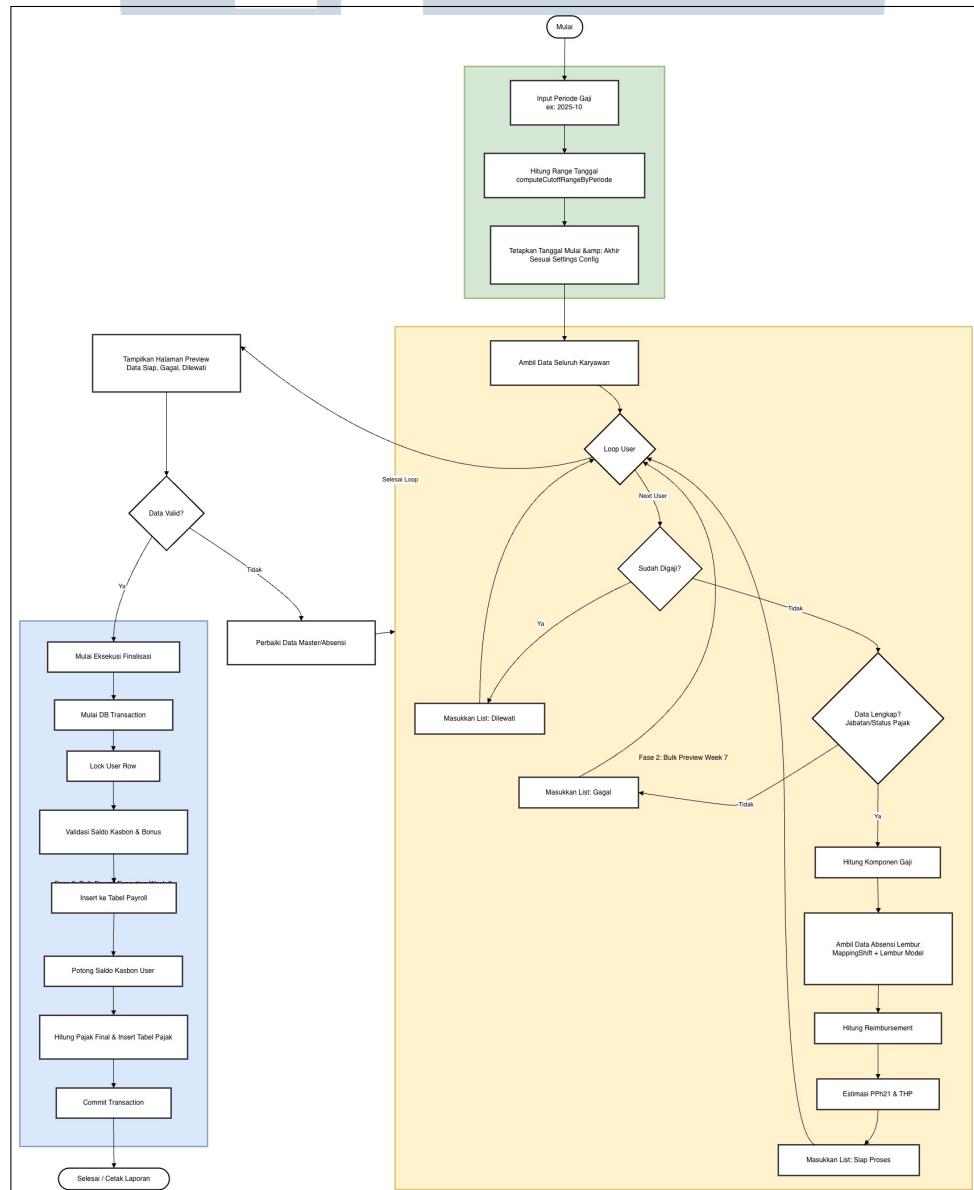
Sebelum perubahan disimpan, sistem kembali melakukan validasi terhadap input yang diberikan pengelola. Pengelola menentukan status akhir pengajuan, apakah tetap *Pending*, *Diterima*, atau *Ditolak*, serta dapat menambahkan catatan penjelasan atas keputusan yang diambil. Sistem kemudian mencatat identitas pengguna yang melakukan persetujuan sebagai penanggung jawab tindakan tersebut. Jika pengajuan dinyatakan diterima, sistem akan melakukan serangkaian pembaruan lanjutan, seperti mengurangi jatah cuti atau kuota izin yang dimiliki karyawan sesuai jenis cuti yang disetujui, serta menyesuaikan status kehadiran pada data penjadwalan dan absensi. Pada kasus tertentu, seperti izin telat atau izin pulang cepat, sistem juga memperbarui informasi jam hadir atau jam pulang, koordinat lokasi, dan bukti foto yang berkaitan agar konsisten dengan kebijakan yang berlaku.

Apabila pengajuan cuti berstatus diterima, sistem mengirimkan notifikasi kepada karyawan bahwa permohonannya telah disetujui oleh atasan. Notifikasi ini disertai tautan menuju halaman riwayat cuti sehingga karyawan dapat melihat kembali detail pengajuan yang telah diproses. Sebaliknya, jika pengajuan ditolak, sistem juga mengirimkan pesan penolakan kepada karyawan dengan mencantumkan nama pihak yang menolak serta catatan penjelasan bila disediakan. Dengan mekanisme ini, alur komunikasi antara karyawan dan pengelola menjadi lebih transparan, karena setiap keputusan persetujuan maupun penolakan tercatat dengan jelas di dalam sistem.

Secara keseluruhan, alur pengajuan dan persetujuan cuti ini memastikan

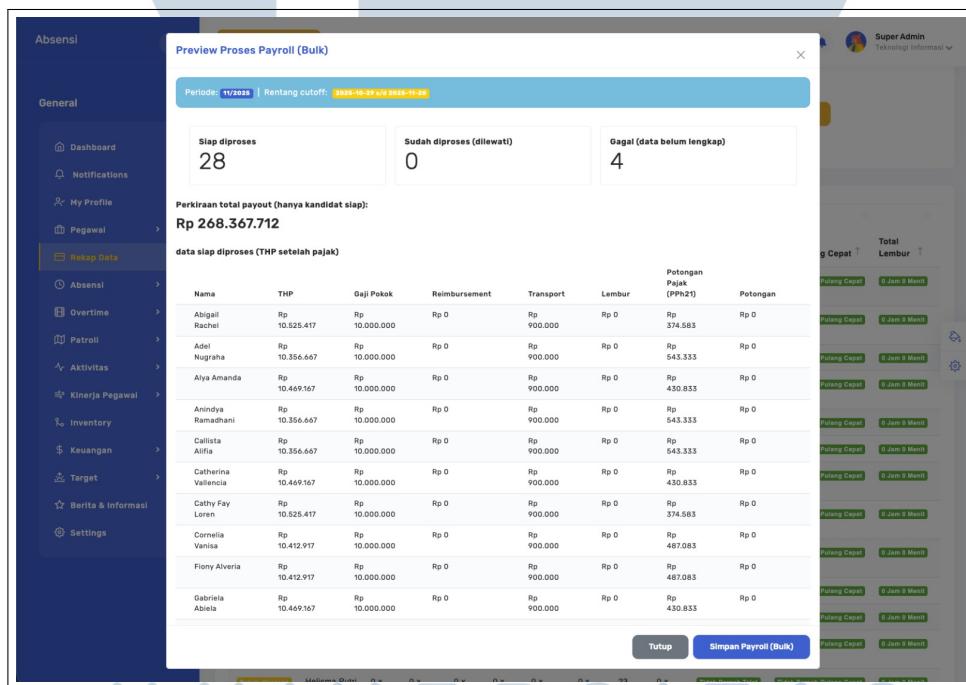
bahwa setiap permohonan diproses melalui tahapan yang terstruktur, mulai dari pengisian formulir oleh karyawan, validasi dan penyimpanan data oleh sistem, distribusi notifikasi kepada pengelola yang berwenang, hingga pembaruan kuota cuti dan data kehadiran setelah keputusan akhir diambil. Struktur ini tidak hanya mendukung akuntabilitas dan keterlacakkan setiap keputusan, tetapi juga membantu perusahaan menjaga konsistensi antara data absensi, hak cuti karyawan, dan kebijakan sumber daya manusia yang berlaku.

3.3.3 Pengembangan Sistem Payroll



Gambar 3.11. flowchart payroll

Pada minggu ke-3 pelaksanaan magang, kegiatan difokuskan pada pengembangan modul penggajian atau *payroll system* yang terintegrasi dengan data kehadiran karyawan. Pada tahap ini, diimplementasikan mekanisme *bulk payroll*, yaitu proses penghitungan gaji untuk seluruh karyawan dalam satu bulan secara serentak. Perancangan sistem ini meliputi pembentukan struktur data payroll bulanan, pengambilan data absensi harian dari modul presensi berbasis pengenalan wajah, serta integrasi dengan komponen perhitungan tunjangan, potongan, dan hak-hak karyawan lainnya lebih lengkapnya ada pada gambar 3.11. Sistem *bulk* ini dirancang agar seluruh proses perhitungan dapat dilakukan secara otomatis, terstandarisasi, dan bebas dari kesalahan perhitungan manual. Dengan adanya mekanisme tersebut, bagian administrasi perusahaan dapat memproses penggajian secara lebih cepat dan efisien tanpa perlu melakukan perhitungan terpisah untuk setiap karyawan dalam satu periode penggajian.



Gambar 3.12. Gambar preview bulk proses payroll

Pada minggu ke-7, pengembangan modul payroll dilanjutkan dengan penambahan sistem *cutoff* dan fitur *preview* sesuai yang terlihat pada Gambar 3.12 untuk proses payroll bulanan. Sistem *cutoff* dirancang untuk menentukan batas waktu pengambilan data kehadiran yang akan dihitung pada suatu periode penggajian, sehingga hanya data absensi dalam rentang tanggal tertentu yang digunakan dalam perhitungan gaji. Dengan adanya mekanisme ini, proses payroll menjadi lebih tertata karena data yang digunakan telah terkunci pada periode

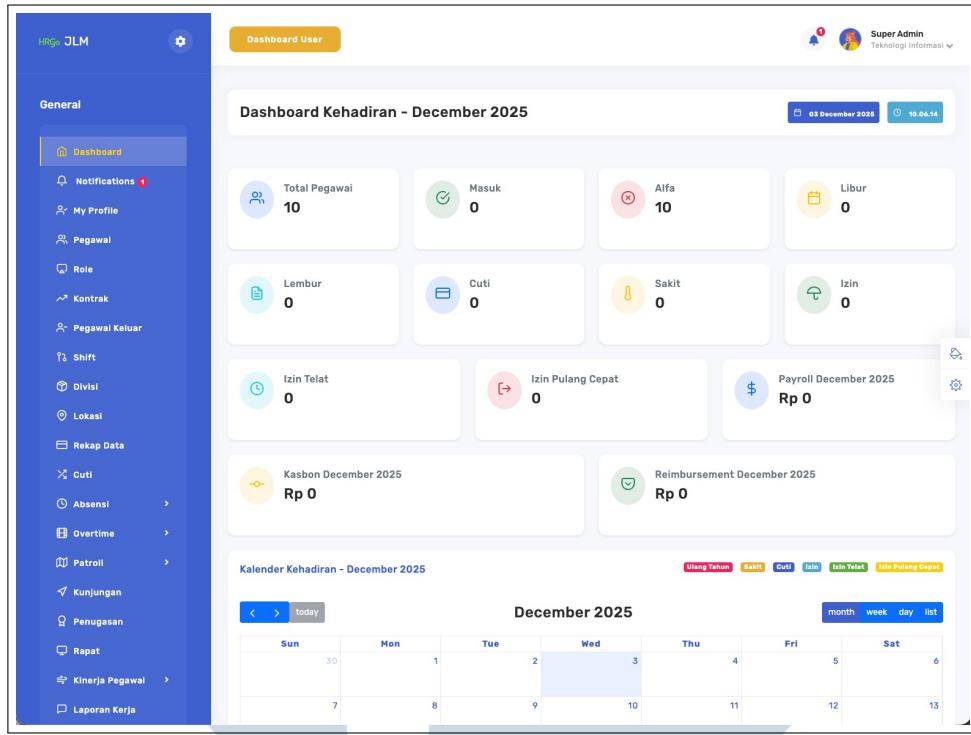
tertentu dan tidak berubah di tengah proses perhitungan. Selain itu, fitur *preview bulk payroll* ditambahkan untuk memberi kesempatan bagi tim HR atau pengelola keuangan untuk melakukan pemeriksaan awal terhadap hasil perhitungan gaji seluruh karyawan sebelum dilakukan finalisasi. Melalui tampilan pratinjau ini, pengguna dapat memeriksa kesesuaian perhitungan, mendeteksi potensi kesalahan data, serta memastikan bahwa seluruh parameter seperti absensi, tunjangan, lembur, dan potongan telah terhitung dengan benar. Setelah data dipastikan akurat, barulah proses penggajian dapat difinalisasi dan diunggah sebagai payroll resmi untuk periode tersebut.

Dengan adanya rangkaian pengembangan ini, modul payroll menjadi sistematis, dapat diandalkan, dan mampu mengurangi beban kerja administratif secara signifikan. Alur perhitungan yang semula banyak dilakukan secara manual kini dapat berjalan secara otomatis, terstruktur, dan sesuai standar operasional perusahaan.

3.3.4 Penyederhanaan Navigasi dan Perbaikan Antarmuka Pengguna

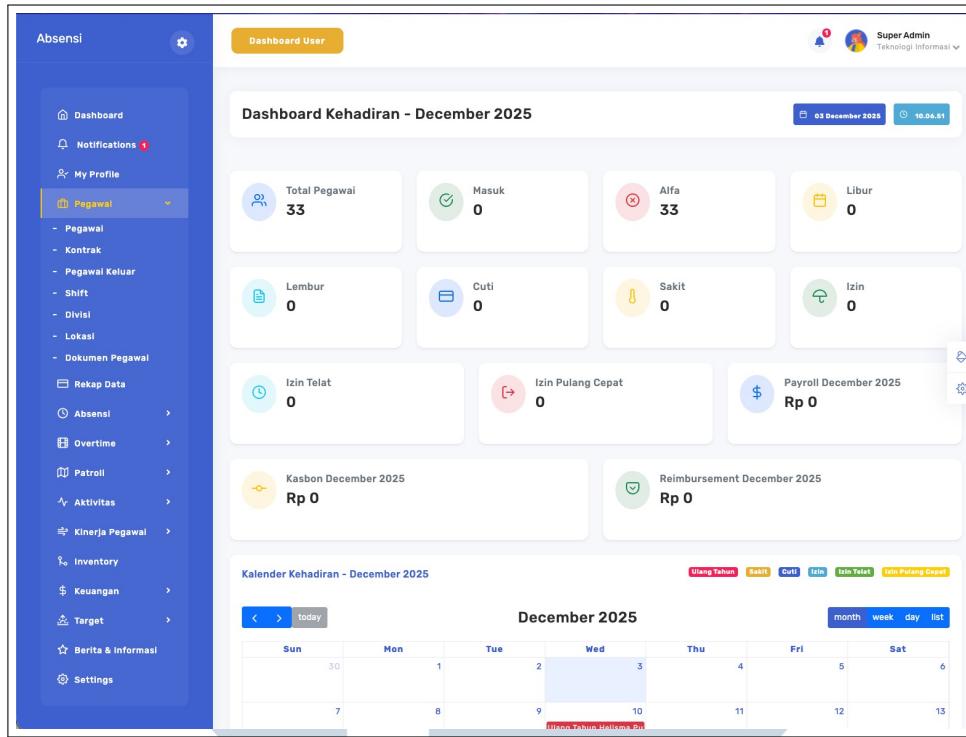
Untuk meningkatkan kenyamanan pengguna, dilakukan penyederhanaan navigasi pada sistem HRGO agar struktur menu menjadi lebih mudah dipahami dan akses ke fitur utama dapat dilakukan dengan cepat. Menu dan halaman diatur ulang sesuai kategori fungsi, seperti administrasi, kepegawaian, dan laporan.





Gambar 3.13. Gambar navigasi side bar lama

Gambar 3.13 menyajikan preview dari navigasi sistem HRGO versi sebelum perancangan ulang. Pada tampilan ini, menu navigasi utama yang terletak di sisi kiri menunjukkan daftar item yang cukup panjang dan padat. Struktur menu terlihat kurang terorganisir dengan pengelompokan yang minim, di mana banyak fitur utama (Dashboard, My Profile, Pegawai, Kontrak, Shift, hingga Laporan Kerja) terdaftar pada tingkat hirarki yang hampir sama. Hal ini dapat mengurangi efisiensi akses dan berpotensi menyebabkan kebingungan bagi pengguna baru, karena mereka harus memindai daftar yang panjang untuk menemukan fungsi yang diinginkan. Kelemahan ini menjadi salah satu pendorong utama dilakukannya proyek perancangan ulang guna menyederhanakan struktur menu dan meningkatkan kenyamanan pengguna.

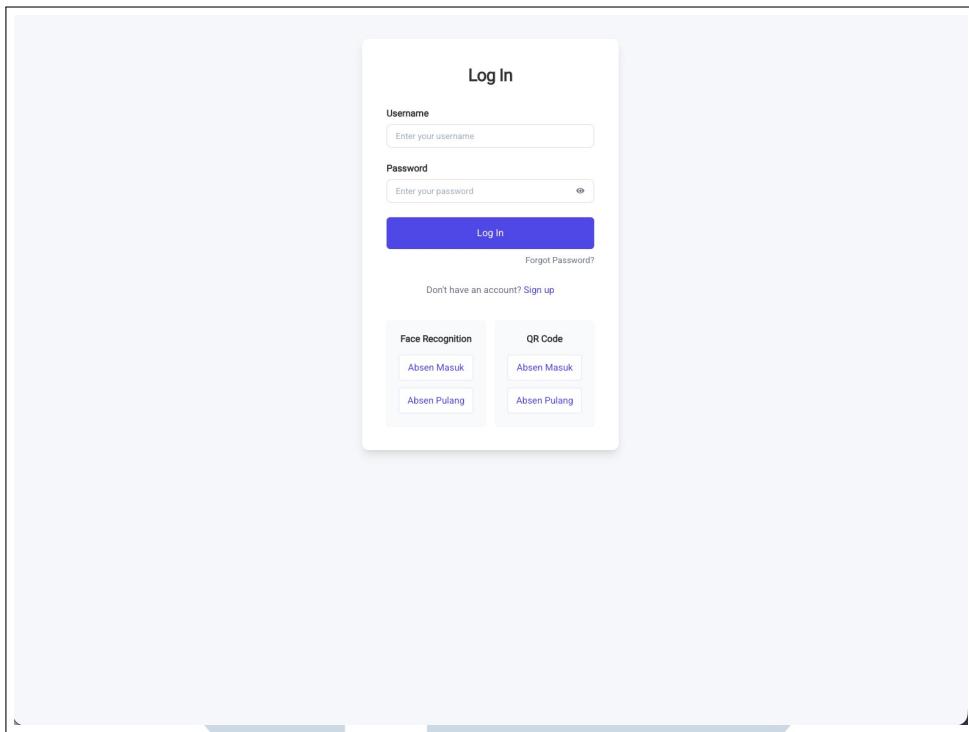


Gambar 3.14. Gambar navigasi side bar lama

Gambar 3.14 menunjukkan preview navigasi sistem HRGO setelah dilakukan penyederhanaan dan perbaikan antarmuka pengguna. Perubahan paling signifikan adalah pengelompokan item menu ke dalam kategori fungsional yang lebih logis, seperti Pegawai (yang di dalamnya terdapat sub-menu Kontrak dan Pegawai Keluar), Rekap Data, Absensi, Patroli, Aktivitas, Keuangan, dan Target. Penyederhanaan ini membuat struktur menu menjadi lebih ringkas dan intuitif, sejalan dengan tujuan meningkatkan kenyamanan pengguna. Dengan mengelompokkan fitur berdasarkan kategori fungsinya, pengguna kini dapat menemukan fitur utama dengan lebih cepat dan dengan alur yang lebih mudah dipahami, memastikan akses yang cepat ke fitur administrasi, kepegawaian, dan laporan, sesuai dengan reorganisasi yang direncanakan.

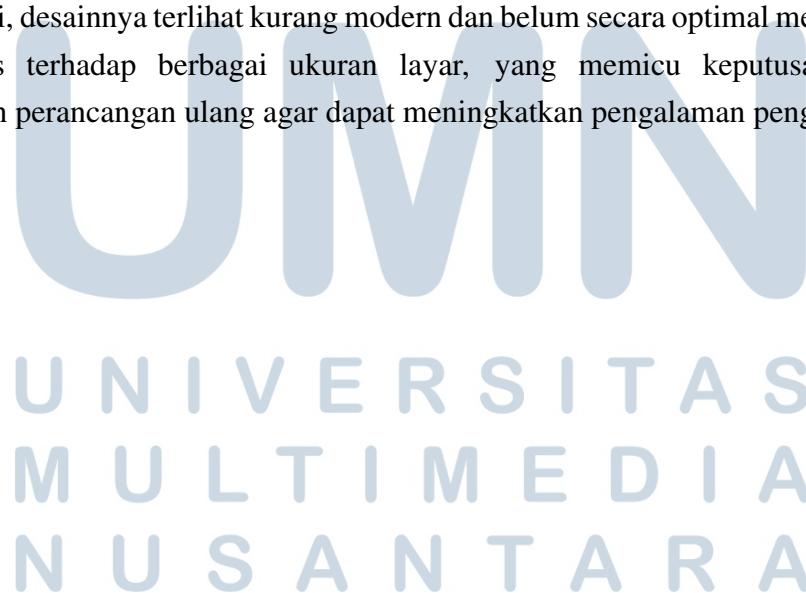
3.3.5 Perancangan Ulang Tampilan Halaman Login

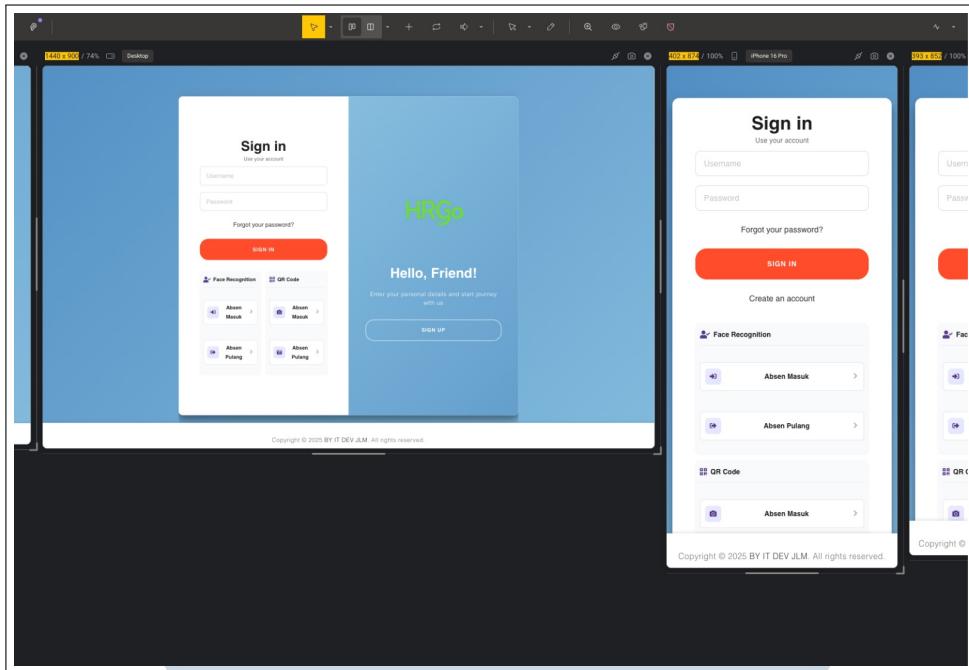
Halaman login menjadi salah satu aspek penting dalam peningkatan keamanan sistem. Dilakukan juga perancangan ulang tampilan halaman login dengan antarmuka yang lebih berwarna dan modern.



Gambar 3.15. Gambar preview login page lama

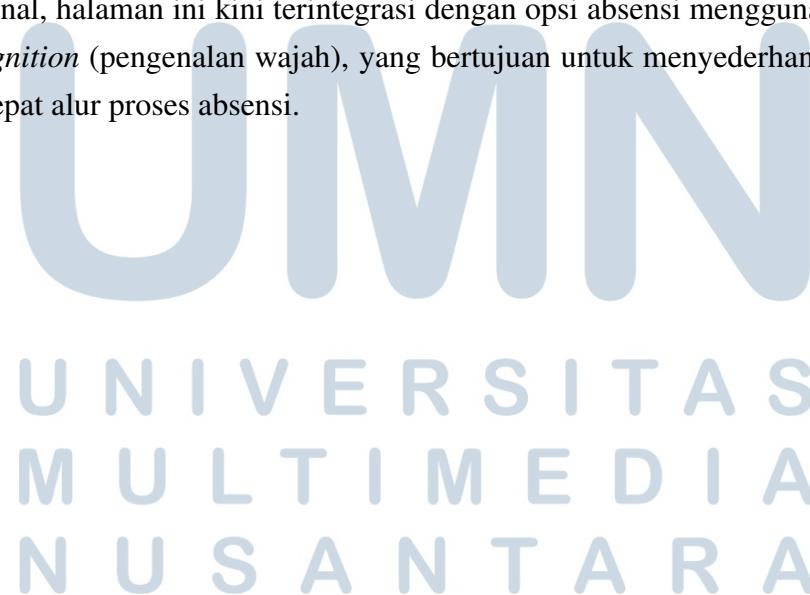
Gambar 3.15 menampilkan halaman login versi lama dari sistem. Antarmuka ini berfungsi sebagai gerbang masuk utama bagi pengguna, meminta Username dan Password sebagai kredensial. Meskipun memenuhi fungsi dasar autentikasi, desainnya terlihat kurang modern dan belum secara optimal mendukung adaptivitas terhadap berbagai ukuran layar, yang memicu keputusan untuk melakukan perancangan ulang agar dapat meningkatkan pengalaman pengguna.

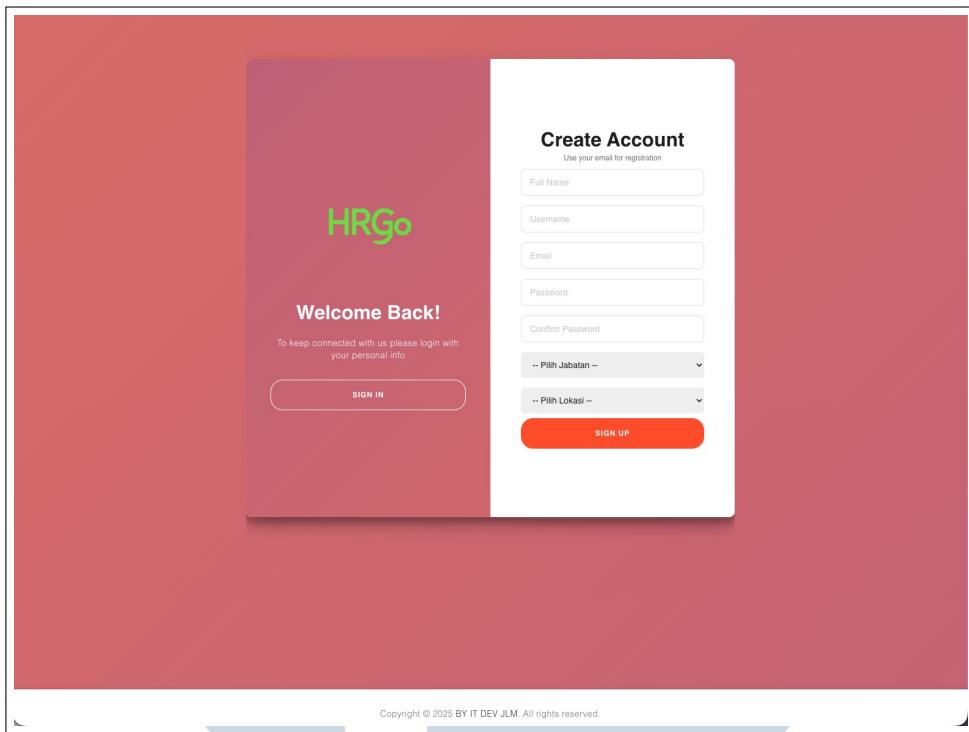




Gambar 3.16. Gambar preview login page baru

Gambar 3.16 menunjukkan preview halaman login yang telah diperbarui. Perancangan ulang ini menghasilkan tampilan yang lebih modern, bersih, dan yang terpenting, adaptif (responsive) terhadap berbagai ukuran layar (desktop dan mobile), memastikan pengguna dapat mengakses sistem dengan lancar. Selain login konvensional, halaman ini kini terintegrasi dengan opsi absensi menggunakan fitur *face recognition* (pengenalan wajah), yang bertujuan untuk menyederhanakan dan mempercepat alur proses absensi.





Gambar 3.17. Gambar preview register page baru

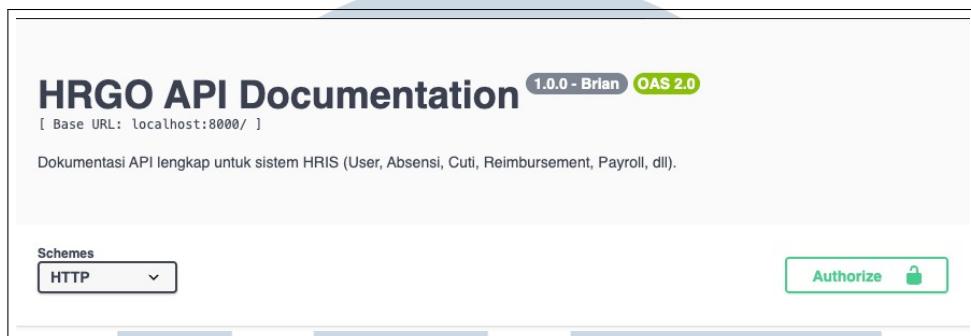
Gambar 3.17 menyajikan preview halaman *Register* (Pendaftaran Akun Baru) yang selaras dengan desain login baru. Halaman ini dirancang untuk memandu pengguna baru dalam memasukkan detail pendaftaran, termasuk Full Name, Username, Email, Password, dan informasi penting lainnya seperti Posisi/Jabatan. Konsistensi desain antara halaman login dan register ini memastikan pengguna mendapatkan pengalaman antarmuka yang terpadu dan intuitif saat pertama kali berinteraksi dengan sistem.

3.3.6 Pengembangan API untuk Aplikasi Mobile

Sebagai upaya untuk memperluas jangkauan aksesibilitas sistem HRGO, dilakukan pengembangan *Application Programming Interface* (API) yang berfungsi sebagai jembatan penghubung antara basis data utama pada sisi *backend* dengan aplikasi *mobile* yang digunakan oleh karyawan. API ini dikembangkan dengan arsitektur RESTful yang mengembalikan respons dalam format JSON, sehingga memastikan pertukaran data yang ringan dan efisien.

Seluruh *controller* API telah dilengkapi dengan dokumentasi yang dibuat dalam format file (.YAML) lalu dikompilasi menggunakan Swagger UI. Dokumentasi API ini sangat krusial untuk mempermudah proses pengembangan

dan serah terima (*hand over*) kepada tim *mobile developer*. Tampilan antarmuka dokumentasi Swagger dapat dilihat pada Gambar 3.18.



Gambar 3.18. Tampilan Header Dokumentasi API (Swagger)

Secara spesifik, terdapat 11 *controller* API yang dibangun untuk menangani berbagai modul fungsionalitas dalam aplikasi. Setiap *controller* bertanggung jawab atas *endpoint* tertentu yang memisahkan logika berdasarkan fitur yang tersedia. Berikut adalah rincian dari kesebelas API yang telah dikembangkan:

- *UsersController*: Mengelola data profil pengguna, termasuk pembaruan informasi pribadi, foto profil, dan pengaturan akun lainnya. Daftar *endpoint* untuk modul ini dapat dilihat pada Gambar 3.19.



Gambar 3.19. Daftar Endpoint *UsersController*

- *AuthApiController*: Mengelola proses autentikasi keamanan, termasuk *login*, *logout*, dan validasi token pengguna untuk memastikan hanya pengguna terdaftar yang dapat mengakses sistem. Detail *endpoint* autentikasi disajikan pada Gambar 3.20.

Auth		
POST	/register	Register User
POST	/login	General Login
POST	/login-admin	Login Admin
POST	/login-user	Login User
POST	/forgot-password	Forgot Password Request
POST	/reset-password	Reset Password Action
GET	/master-data	Get Master Data for Registration
POST	/profile	Get Profile
POST	/update-profile	Update Profile
POST	/change-password	Change Password
POST	/logout	Logout
POST	/switch/{id}	Switch User (Admin)

Gambar 3.20. Daftar Endpoint AuthApiController

- DashboardApiController: Menyediakan data ringkasan statistik kehadiran dan informasi penting lainnya yang ditampilkan pada halaman utama (*home screen*) aplikasi *mobile*. Struktur *endpoint* dashboard terlihat pada Gambar 3.21.

Dashboard		
GET	/dashboard	Get Dashboard Data

Gambar 3.21. Daftar Endpoint DashboardApiController

- FaceApiController: Menangani proses inti dari absensi biometrik, meliputi penerimaan gambar wajah dari kamera ponsel, pengiriman ke layanan pengenalan wajah, serta verifikasi pencocokan identitas. Implementasi *endpoint* ini ditunjukkan pada Gambar 3.22.

Face Recognition		
POST	/face/upload-photo	Upload Face Photo
POST	/face/get-photo	Get Face Photo
POST	/face/delete-photo	Delete Face Photo

Gambar 3.22. Daftar Endpoint FaceApiController

- BeritaApiController: Berfungsi untuk mendistribusikan informasi, pengumuman perusahaan, atau berita internal terkini kepada seluruh karyawan melalui aplikasi. Rincian *endpoint* berita dapat dilihat pada Gambar 3.23.

Berita		
POST	/berita	Handle Berita

Gambar 3.23. Daftar Endpoint BeritaApiController

- CutiApiController: Memfasilitasi proses pengajuan cuti, memuat riwayat cuti karyawan, serta menampilkan sisa kuota cuti tahunan yang tersedia. Dokumentasi *endpoint* cuti terdapat pada Gambar 3.24.

Cuti		
POST	/cuti/list	List Cuti
POST	/cuti/submit	Submit Cuti
POST	/cuti/approval	Approval Cuti

Gambar 3.24. Daftar Endpoint CutiApiController

- LemburApiController: Mengelola data terkait jam kerja tambahan, mulai dari pengajuan perintah lembur hingga rekapitulasi durasi lembur yang telah dilakukan. Daftar *endpoint* lembur ditunjukkan pada Gambar 3.25.

Lembur		
POST	/lembur/request	Request Lembur
POST	/lembur/history	History Lembur
POST	/lembur/approval	Approval Lembur

Gambar 3.25. Daftar Endpoint LemburApiController

- KasbonApiController: Menangani fitur pengajuan pinjaman karyawan (*cash advance*) serta memantau status persetujuan dan riwayat pemotongan gaji terkait. Detail *endpoint* kasbon dapat dilihat pada Gambar 3.26.

Kasbon		
GET	/kasbon	List Kasbon
POST	/kasbon	Create Kasbon
GET	/kasbon/{id}	Show Kasbon
PUT	/kasbon/{id}	Update Kasbon
DELETE	/kasbon/{id}	Delete Kasbon

Gambar 3.26. Daftar Endpoint KasbonApiController

- ReimbursementApiController: Menyediakan *endpoint* untuk pengajuan klaim penggantian biaya operasional karyawan beserta unggahan bukti transaksi. Struktur API ini terlihat pada Gambar 3.27.

Reimbursement		
POST	/reimbursement/list	List Reimbursement
POST	/reimbursement/create	Create Reimbursement
POST	/reimbursement/detail	Detail Reimbursement
POST	/reimbursement/approval	Approve/Reject Reimbursement

Gambar 3.27. Daftar Endpoint ReimbursementApiController

- TunjanganApiController: Menampilkan rincian tunjangan-tunjangan yang berhak diterima oleh karyawan berdasarkan jabatan atau performa kerja. Dokumentasi *endpoint* tunjangan disajikan pada Gambar 3.28.



Gambar 3.28. Daftar Endpoint TunjanganApiController

- PayrollApiController: Mengintegrasikan data penggajian sehingga karyawan dapat melihat rincian slip gaji (*payslip*) bulanan secara transparan dan privat melalui ponsel. Rincian *endpoint* payroll dapat dilihat pada Gambar 3.29.



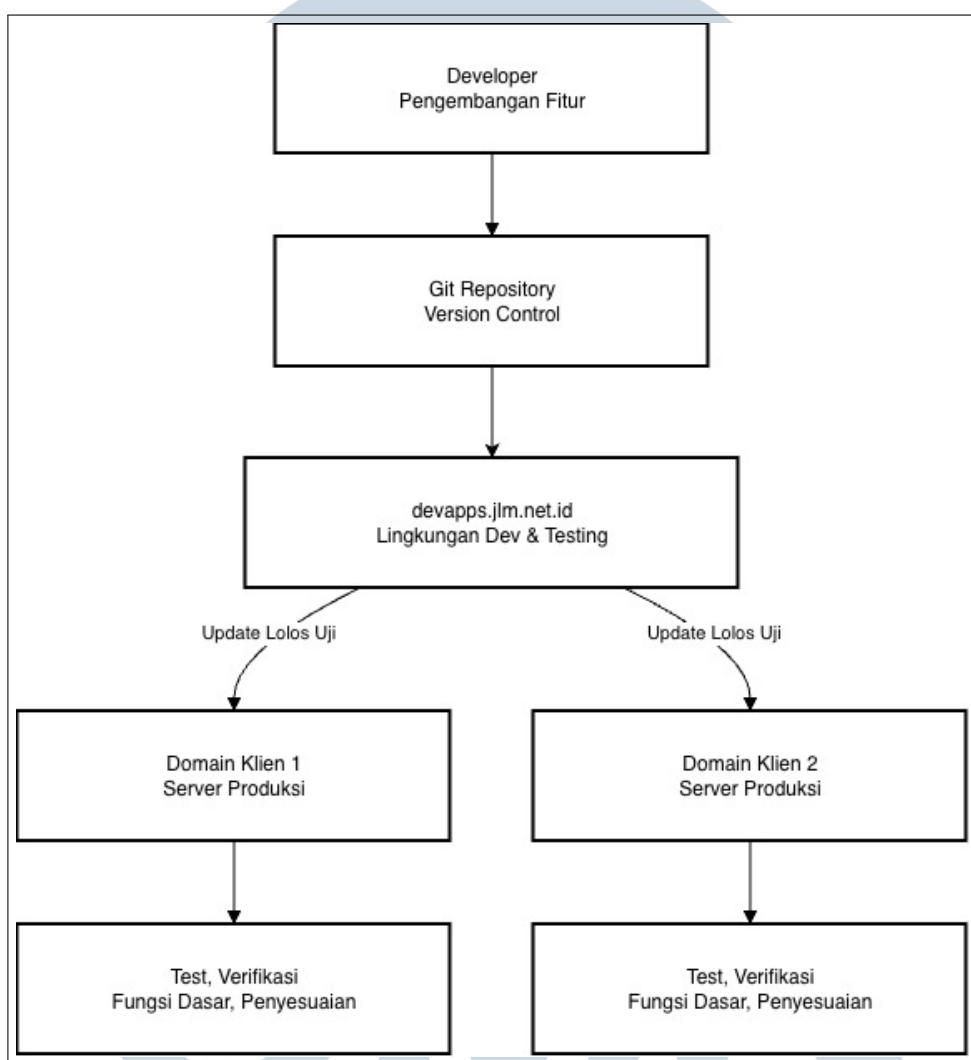
Gambar 3.29. Daftar Endpoint PayrollApiController

3.3.7 Pipeline Deployment dan Update Sistem HRGO

Dalam proses pemeliharaan dan pengembangan lanjutan, sistem HRGO dikelola menggunakan beberapa lingkungan server yang memiliki peran berbeda. Pendekatan ini digunakan agar setiap pembaruan fitur dapat diuji terlebih dahulu tanpa mengganggu kestabilan aplikasi yang sudah digunakan klien. Secara umum, terdapat empat domain utama yang dikelola, yaitu:

1. HRGO.jlm.net.id sebagai aplikasi HRGO utama yang berfungsi sebagai *baseline* atau acuan versi stabil. Server ini diperbarui secara manual dan digunakan sebagai referensi ketika membandingkan perilaku sistem.
2. devapps.jlm.net.id sebagai lingkungan pengembangan dan pengujian. Server ini adalah satu-satunya lingkungan yang terhubung langsung dengan repositori versi kontrol (Git) untuk menarik pembaruan kode secara otomatis atau melalui perintah terminal.
3. Domain klien 1 sebagai server produksi yang digunakan oleh perusahaan klien, yang menjalankan HRGO dengan konfigurasi spesifik. Pembaruan dilakukan secara manual melalui akses cPanel atau FTP.

4. Domain klien 2 sebagai server produksi untuk klien lainnya dengan pola pengelolaan yang serupa dengan domain klien 1.



Gambar 3.30. Gambar pipeline update HRGO

Mengingat perbedaan konfigurasi server tersebut, pada Gambar 3.30 alur pembaruan sistem HRGO menerapkan metode hibrida yang menggabungkan versi kontrol modern dengan manajemen berkas konvensional. Tahapan deployment adalah sebagai berikut:

1. Pengembangan dan komit kode

Pengembangan fitur baru atau perbaikan bug dilakukan pada lingkungan pengembangan lokal. Setelah perubahan selesai, kode dikirim (*push*) ke repositori Git untuk melacak perubahan setiap pembaruan.

2. Sinkronisasi ke devapps.jlm.net.id via Git

Kode terbaru dari repositori ditarik (*pull*) ke server devapps.jlm.net.id.

Karena server ini terintegrasi dengan Git, proses pembaruan dapat dilakukan dengan cepat untuk kebutuhan pengujian intensif.

3. Pengujian dan validasi modul

Seluruh modul utama diuji pada server devapps.jlm.net.id. Pengujian ini bertujuan untuk memastikan fitur berjalan lancar dan bebas dari bug. Pembaruan ke tahap selanjutnya ditahan hingga modul dinyatakan sempurna dan stabil sepenuhnya di lingkungan ini.

4. Distribusi manual ke server produksi (FTP/cPanel)

Setelah modul pada server pengembangan dipastikan stabil dan tidak memiliki bug, proses pembaruan ke server klien dilakukan secara berkala. Proses ini tidak menggunakan Git, melainkan dilakukan secara manual dengan cara mengunggah berkas yang telah diperbarui melalui File Manager di cPanel atau menggunakan protokol FTP (seperti FileZilla) dengan metode *drag and drop*. Pendekatan ini memberikan kontrol penuh terhadap berkas apa saja yang akan ditimpa di server produksi.

5. Verifikasi operasional

Setelah proses unggah berkas selesai, dilakukan pengecekan fungsionalitas dasar (*smoke test*) pada domain klien untuk memastikan proses penyalinan data berjalan sukses dan sistem dapat digunakan kembali oleh pengguna secara normal.

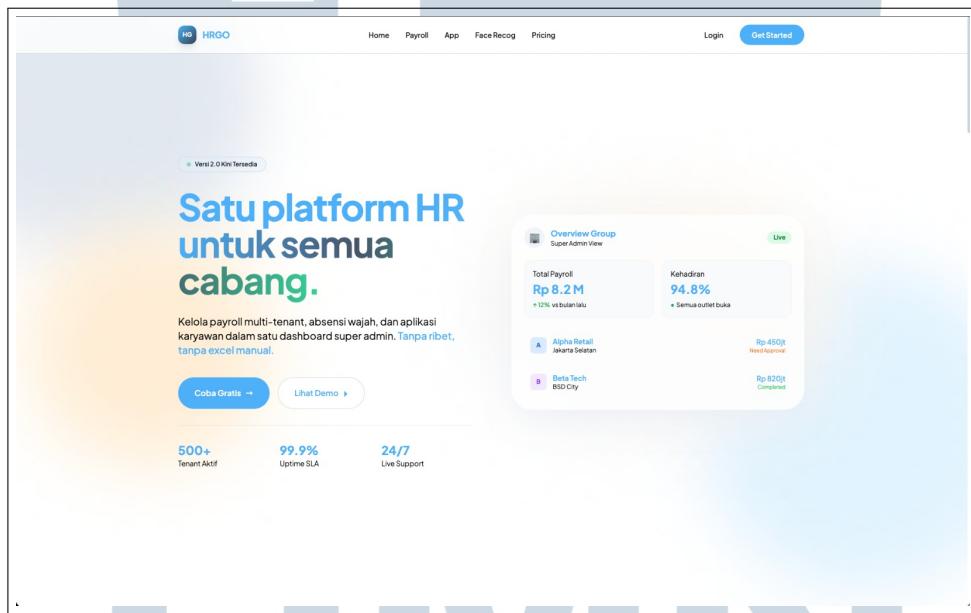
3.3.8 Pengembangan Aplikasi ke SaaS

Pada minggu ke-12 sampai minggu ke-16, pengembangan aplikasi HRGO memasuki tahap transformasi menuju arsitektur *Software as a Service* (SaaS). Tahap ini meliputi perancangan sistem multi-tenant berbasis *company id*, penyesuaian arsitektur frontend dan backend, serta pengujian fungsional untuk memastikan seluruh modul bekerja secara terisolasi antar perusahaan.

A Implementasi Frontend pada Arsitektur SaaS

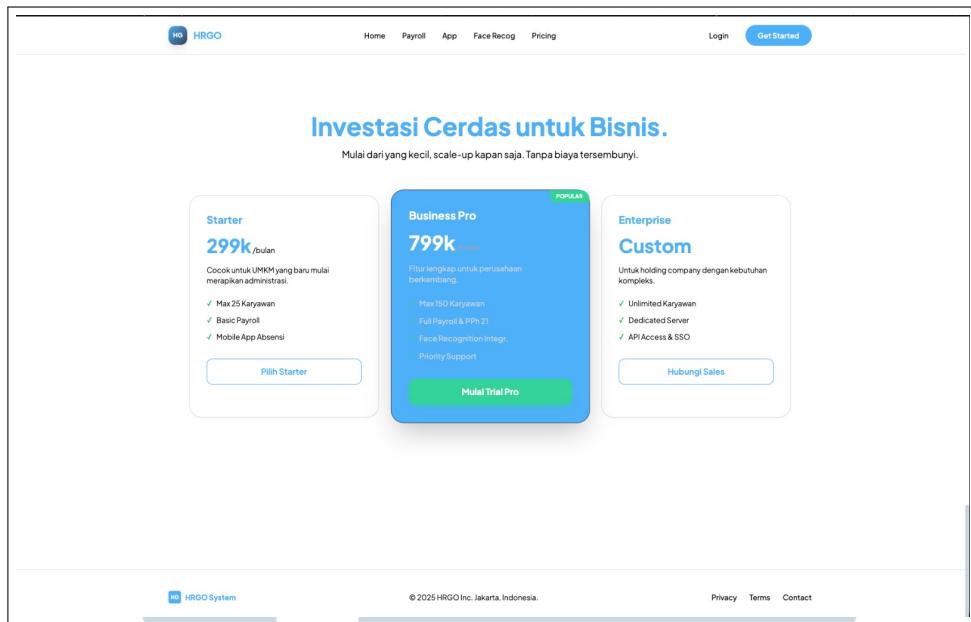
Transformasi sistem menuju model *Software as a Service* (SaaS) menuntut adanya pembaruan signifikan pada sisi antarmuka, terutama untuk memfasilitasi

akses publik dan pendaftaran mandiri. Pengembangan sisi depan (*frontend*) ini dimulai dengan pendekatan inkremental atau bertahap, di mana modernisasi teknologi diterapkan pada modul-modul baru terlebih dahulu. Mengingat aplikasi utama HRGO dibangun menggunakan kerangka kerja Bootstrap yang mulai tertinggal dari segi fleksibilitas desain modern, maka untuk pengembangan halaman publik (*landing page*), diputuskan untuk menggunakan teknologi yang lebih mutakhir. Pustaka Tailwind CSS dan GSAP (*GreenSock Animation Platform*) diimplementasikan melalui *Content Delivery Network* (CDN). Pendekatan ini dipilih untuk menghadirkan tampilan yang estetis, ringan, dan interaktif tanpa perlu melakukan perombakan total pada basis kode *legacy* yang sudah ada.



Gambar 3.31. Tampilan landing page yang dibangun menggunakan Tailwind CSS dan GSAP

Salah satu komponen vital yang dikembangkan dengan teknologi baru tersebut adalah halaman penawaran harga (*pricing*) dan alur pembayaran. Antarmuka pada bagian ini dirancang untuk menyajikan informasi paket berlangganan secara komparatif, sehingga memudahkan calon penyewa dalam memilih layanan yang sesuai dengan kebutuhan perusahaan. Setelah pemilihan paket, pengguna diarahkan melalui antarmuka pembayaran yang terintegrasi. Fokus utama pada perancangan bagian ini adalah menciptakan pengalaman pengguna yang mulus (*seamless*) mulai dari pengenalan fitur hingga penyelesaian transaksi, sehingga proses akuisisi klien baru dapat berjalan secara otomatis melalui sistem tanpa intervensi manual yang berlebihan.



Gambar 3.32. Antarmuka halaman pemilihan paket dan pembayaran

Terkait dengan antarmuka aplikasi utama yang digunakan oleh klien, desain visual yang seragam tetap dipertahankan untuk menjaga konsistensi pengalaman pengguna. Meskipun arsitektur belakang layar telah berubah menjadi multi-penyewa, tampilan antarmuka bagi pengguna akhir di setiap perusahaan dibuat identik. Perbedaan mendasar terletak pada logika penyajian data di sisi *frontend*, di mana sistem secara otomatis memfilter informasi berdasarkan identitas perusahaan (*company ID*). Dengan demikian, isolasi data tetap terjaga dengan ketat meskipun seluruh penyewa mengakses antarmuka aplikasi dari satu pintu gerbang yang sama.

B Perancangan Backend untuk SaaS

Dari sisi arsitektur, kunci utama penerapan model *Software as a Service* (SaaS) pada HRGO berada pada lapisan backend. Pada lapisan inilah logika pemisahan data antar perusahaan diatur, sehingga setiap tenant memiliki ruang data yang sepenuhnya terisolasi meskipun dijalankan pada server dan aplikasi yang sama. Konsep utama yang digunakan adalah pola *multi-tenant* berbasis *company_id*, di mana setiap data bisnis selalu dikaitkan dengan satu entitas perusahaan tertentu.

Pemisahan ini dimulai dengan perancangan tabel *companies* sebagai tabel induk yang merepresentasikan setiap perusahaan yang menggunakan HRGO. Tabel *companies* memiliki kolom *primary key* berupa *company_id* yang kemudian

dijadikan *foreign key* pada tabel-tabel lain, seperti tabel `users`, `employees`, `payroll`, `pengajuan`, dan modul-modul inti HRGO lainnya. Dengan desain ini, setiap baris data di tabel operasional selalu terhubung ke satu perusahaan tertentu melalui `company_id`, sehingga secara struktur database sudah tercipta batas yang jelas antara data milik satu perusahaan dan perusahaan lainnya.

Dari sisi alur sistem, ketika pengguna melakukan proses login, backend akan mengidentifikasi perusahaan mana yang terkait dengan akun tersebut. Informasi `company_id` kemudian disimpan pada sesi, *token* autentikasi, atau konteks permintaan (*request context*) yang digunakan sepanjang interaksi pengguna. Pada tahap perancangan juga didefinisikan bahwa `company_id` tidak boleh dipercaya begitu saja dari input pengguna, tetapi harus selalu diambil dari identitas pengguna yang sudah terautentikasi. Pendekatan ini mencegah kemungkinan pengguna mengubah `company_id` secara manual untuk mengakses data milik perusahaan lain.

Untuk memperjelas rancangan, dibuat pula diagram relasi pada Gambar 3.33 yang menunjukkan hubungan antara tabel `companies` dengan tabel-tabel utama pada HRGO. Diagram ini membantu memastikan bahwa tidak ada tabel penting yang terlewat dari skema multi-tenant dan seluruh modul konsisten menggunakan `company_id` sebagai dasar pemisahan data.





Gambar 3.33. Diagram relasi tabel companies dan pemisahan data berbasis company_id

C Pengembangan Backend untuk SaaS

Setelah rancangan multi-tenant ditetapkan, tahap berikutnya adalah mengimplementasikan konsep tersebut dalam kode backend. Implementasi berfokus pada dua hal utama, yaitu cara memisahkan setiap perusahaan pada level query, dan cara memfilter data agar setiap permintaan hanya memproses data milik perusahaan yang tepat. Seluruh lapisan backend, mulai dari *controller* hingga

service layer dan akses ke database, disesuaikan untuk selalu mempertimbangkan `company_id`.

```
class InventoryController extends Controller
{
    public function index()
    {
        $title = 'Inventory';
        $search = request()->input('search');

        $currentUser = auth()->user();
        $isSuperAdmin = $currentUser && $currentUser->hasRole('admin') && is_null($currentUser->company_id);

        $inventoriesQuery = Inventory::query()
            ->when($search, function ($query) use ($search) {
                $query->where('nama_barang', 'LIKE', '%' . $search . '%')
                    ->orWhere('kode_barang', 'LIKE', '%' . $search . '%');
            });

        // Scope per company untuk selain super admin (role admin + company_id null)
        if (!$isSuperAdmin) {
            $companyId = $currentUser ? $currentUser->company_id : null;

            if ($companyId) {
                // Asumsikan Inventory berelasi ke Jabatan yang memiliki company_id
                $inventoriesQuery->whereHas('jabatan', function ($q) use ($companyId) {
                    $q->where('company_id', $companyId);
                });
            } else {
                // Jika tidak ada company_id, kosongkan hasil
                $inventoriesQuery->whereRaw('1 = 0');
            }
        }
    }
}
```

Gambar 3.34. Cuplikan kode pemisahan data berbasis `company_id`

Pada Gambar 3.34 lapisan *controller* dan *service*, implementasi difokuskan untuk memastikan seluruh query ke database selalu difilter berdasarkan `company_id`. Secara praktis, setiap fungsi pengambilan data (misalnya daftar karyawan, daftar slip gaji, atau daftar pengajuan) akan menambahkan kondisi filter `company_id` yang diambil dari konteks pengguna. Pola ini diterapkan secara konsisten di semua modul sehingga tidak ada endpoint yang mengambil data tanpa filter perusahaan. Untuk mempermudah pemeliharaan, sering kali dibuat abstraksi berupa fungsi bantu atau *base repository* yang secara otomatis menambahkan filter `company_id` pada setiap query, sehingga risiko pengembang lupa menambahkan filter tersebut dapat diminimalkan.

Pengembangan backend juga mencakup penyesuaian *model* dan *database* agar semua operasi CRUD (Create, Read, Update, Delete) selalu bekerja dalam ruang lingkup satu perusahaan. Misalnya, ketika super admin menambah perusahaan baru, sistem akan membuat entri baru di tabel `companies` dan memastikan bahwa pengguna-pengguna yang didaftarkan setelahnya selalu terkait dengan `company_id` tersebut. Sebaliknya, ketika pengguna biasa melakukan operasi pada modul HR, seluruh tindakan mereka tetap terkurung pada data perusahaannya masing-masing. Dengan pendekatan ini, tampilan dan fitur aplikasi dapat tetap

sama untuk semua pengguna, tetapi secara backend data sudah terpisah rapi per perusahaan.

D Integrasi dan Pengujian Akhir Backend SaaS

Pada tahap integrasi dan pengujian, fokus utama adalah memastikan bahwa seluruh mekanisme pemisahan dan pemfilteran data berbasis `company_id` berjalan dengan benar di semua modul. Backend diuji dengan mensimulasikan beberapa perusahaan yang berbeda, masing-masing dengan pengguna, karyawan, dan data transaksi yang berbeda pula. Skenario pengujian meliputi akses normal oleh pengguna dalam satu perusahaan, akses lintas modul, hingga percobaan akses yang seharusnya ditolak, misalnya ketika pengguna mencoba mengakses data yang bukan milik perusahaannya melalui manipulasi URL atau parameter.

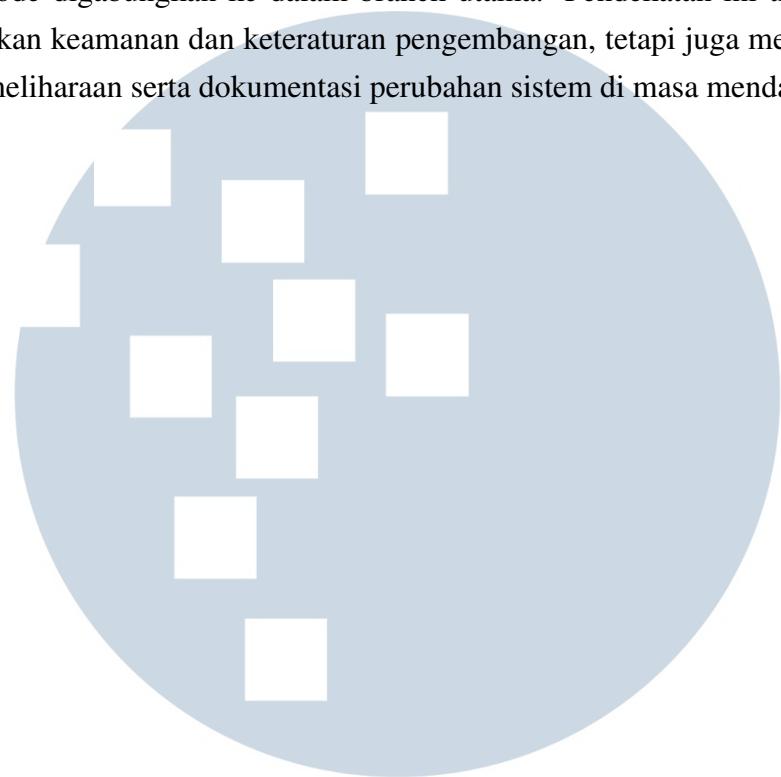
Selain itu, dilakukan juga verifikasi konsistensi antara data yang dikembalikan dan yang ditampilkan di frontend. Hal ini memastikan bahwa filter `company_id` benar-benar diterapkan secara end-to-end, bukan hanya di satu sisi saja. Selama proses uji coba, setiap bug yang berpotensi menyebabkan kebocoran data antar perusahaan menjadi prioritas utama untuk diperbaiki. Pengujian performa juga dilakukan untuk memastikan bahwa penambahan filter `company_id` pada query tidak menimbulkan beban berlebih pada server meskipun sistem menangani banyak perusahaan secara bersamaan. Melalui rangkaian integrasi dan pengujian ini, backend HRGO dipastikan siap mendukung model SaaS dengan isolasi data yang kuat dan perilaku sistem yang stabil.

3.4 Kendala dan Solusi yang Ditemukan

Selama pelaksanaan magang, kendala pertama yang dihadapi adalah tidak adanya sistem version control pada proyek yang dikembangkan. Setiap pembaruan kode atau fitur baru langsung diterapkan ke server produksi tanpa melalui proses versioning yang terstruktur. Kondisi ini menimbulkan risiko terjadinya konflik kode, kesulitan dalam melakukan rollback, serta menyulitkan proses kolaborasi antar pengembang karena tidak ada riwayat perubahan yang terdokumentasi dengan baik.

Sebagai solusi, setiap proyek disarankan untuk dibuatkan repositori Git agar seluruh perubahan kode dapat dilacak secara sistematis. Dengan penerapan Git, tim pengembang dapat melakukan branching untuk fitur baru, melakukan

commit secara berkala, serta memanfaatkan pull request sebagai sarana review sebelum kode digabungkan ke dalam branch utama. Pendekatan ini tidak hanya meningkatkan keamanan dan keteraturan pengembangan, tetapi juga memudahkan proses pemeliharaan serta dokumentasi perubahan sistem di masa mendatang.



UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA