

BAB 3

PENGALAMAN MAGANG

3.1 Kedudukan dan Koordinasi

Selama menjalani kegiatan magang di PT Buana Selaras Globalindo, penulis diberikan kepercayaan untuk mengisi peran sebagai satu-satunya *Fullstack Developer* dalam proyek pengembangan sistem informasi internal perusahaan. Penempatan ini tidak hanya memberikan pengalaman praktis dalam dunia kerja profesional, tetapi juga menantang kemampuan penulis dalam mengelola seluruh aspek teknis pengembangan perangkat lunak secara mandiri.

Secara struktural, penulis berada di bawah Divisi Teknologi Informasi dan bekerja langsung di bawah arahan pembimbing lapangan dari perusahaan. Dalam proses pengembangan sistem, penulis aktif berkoordinasi dengan berbagai pihak terkait, seperti bagian HR, keuangan, dan operasional, untuk memahami kebutuhan fungsional yang harus diakomodasi oleh sistem yang dikembangkan.

Dalam pelaksanaan tugas, penulis menggunakan beberapa teknologi terkini untuk mendukung pengembangan aplikasi. Di sisi *frontend*, penulis mengandalkan *React* dan *Tailwind CSS* untuk membangun antarmuka pengguna yang intuitif dan responsif. Penulis juga menggunakan *Ionic Framework* agar sistem dapat berjalan optimal di berbagai perangkat, termasuk *mobile*, sesuai kebutuhan operasional perusahaan.

Untuk sisi *backend*, penulis memanfaatkan framework *Laravel* yang dikombinasikan dengan pembuatan *RESTful API*, sehingga data dapat diolah secara efisien dan diakses dengan aman oleh antarmuka pengguna. Seluruh proses pengembangan dilakukan secara bertahap dan terdokumentasi, mulai dari analisis kebutuhan, perancangan, implementasi, pengujian, hingga penyempurnaan berdasarkan umpan balik pengguna.

Meskipun bekerja secara mandiri dalam aspek teknis, komunikasi dan koordinasi tetap dilakukan secara rutin melalui pertemuan langsung, diskusi daring, dan laporan perkembangan berkala kepada pembimbing. Dengan demikian, seluruh proses pengembangan berjalan sesuai dengan rencana dan kebutuhan yang telah ditetapkan.

Pengalaman ini tidak hanya memperdalam keterampilan teknis penulis, tetapi juga membentuk pemahaman penulis tentang pentingnya koordinasi lintas

divisi dalam pengembangan sistem informasi yang efektif dan tepat guna.

3.2 Tugas yang Dilakukan

Selama masa magang di PT Buana Selaras Globalindo, Penulis mengemban berbagai tugas dan tanggung jawab yang mencakup seluruh proses pengembangan aplikasi berbasis web dan *mobile*. Tugas-tugas tersebut dilakukan secara bertahap dan terintegrasi untuk mendukung kebutuhan sistem informasi internal perusahaan. Berikut rincian tugas yang penulis lakukan:

1. Analisis Kebutuhan Sistem

Melakukan identifikasi dan pengumpulan informasi terkait kebutuhan pengguna (*user requirements*) melalui wawancara informal dan observasi langsung terhadap proses bisnis di divisi-divisi terkait, seperti HR, keuangan, dan operasional.

2. Perancangan Antarmuka (UI/UX)

Merancang struktur halaman dan antarmuka pengguna menggunakan sketsa dan *wireframe*. Desain dibuat untuk memastikan kemudahan penggunaan dan pengalaman pengguna (*user experience*) yang optimal.

3. Pengembangan Frontend Aplikasi

Mengimplementasikan desain antarmuka menggunakan *React JS* yang dipadukan dengan *Tailwind CSS* untuk membangun tampilan yang responsif, konsisten, dan mudah diakses. Untuk mendukung perangkat *mobile*, penulis menggunakan *Ionic Framework* agar aplikasi dapat berjalan dengan baik di berbagai ukuran layar.

4. Pengembangan Backend dan API

Menggunakan *Laravel* untuk mengembangkan logika bisnis dan pengelolaan data. penulis juga membangun *RESTful API* untuk menghubungkan *frontend* dengan *database* secara efisien dan aman.

5. Pengujian dan Debugging

Melakukan pengujian fungsionalitas dari setiap modul aplikasi untuk memastikan sistem berjalan sesuai dengan yang diharapkan. Bug atau kesalahan yang ditemukan selama proses pengujian diperbaiki secara berkala.

6. **Deployment dan Dokumentasi**

Melakukan proses *deployment* ke server perusahaan setelah sistem selesai dibangun dan diuji. penulis juga menyusun dokumentasi teknis serta panduan penggunaan aplikasi bagi pengguna akhir.

7. **Pemeliharaan dan Penyesuaian Sistem**

Melakukan penyesuaian terhadap fitur atau tampilan aplikasi berdasarkan masukan dari pengguna, serta memastikan sistem tetap berjalan dengan stabil dan sesuai kebutuhan operasional.

Tugas-tugas tersebut memberikan pengalaman nyata dalam mengelola proyek pengembangan perangkat lunak secara utuh, khususnya dalam membangun aplikasi Human Resource (HR Apps), serta memperkuat kemampuan teknis dan non-teknis yang penulis miliki. Dalam proyek ini, penulis terlibat langsung mulai dari proses analisis kebutuhan pengguna, perancangan antarmuka, hingga implementasi dan pengujian sistem yang digunakan oleh pihak internal perusahaan.

Dari sisi teknis, penulis mengembangkan backend menggunakan Laravel yang terhubung ke basis data perusahaan, dan membangun frontend dengan Ionic, React, serta Tailwind CSS untuk menciptakan tampilan antarmuka yang responsif, modern, dan mudah digunakan. penulis merancang serta mengimplementasikan berbagai fitur penting seperti sistem login, manajemen data karyawan, pengajuan dan persetujuan cuti, pinjaman, lembur, hingga permintaan kendaraan dinas. penulis juga mengelola proses validasi input, otorisasi berdasarkan peran pengguna (role-based access control), serta integrasi API antara frontend dan backend.

Dari sisi non-teknis, penulis meningkatkan keterampilan dalam menyusun dokumentasi teknis, menyampaikan hasil kerja kepada supervisor, serta berkolaborasi dengan rekan tim dan pihak terkait untuk memastikan sistem berjalan sesuai kebutuhan bisnis. penulis belajar menyusun rencana kerja, mengatur waktu pengerjaan sesuai target, serta menyelesaikan tantangan teknis dengan pendekatan yang sistematis dan efisien.

Melalui keterlibatan penulis dalam proyek HR Apps ini, penulis tidak hanya meningkatkan keterampilan pemrograman dan perancangan sistem, tetapi juga memperoleh wawasan nyata tentang bagaimana membangun solusi teknologi yang berdampak langsung terhadap proses kerja di lingkungan perusahaan.

Tabel 3.1. Rangkuman Aktivitas Mingguan Selama Magang

Minggu	Aktivitas
Minggu 1	Briefing awal, pengenalan perusahaan dan sistem HR yang akan dikembangkan
Minggu 2	Setup lingkungan kerja (Laravel, React, Ionic), konfigurasi database
Minggu 3	Implementasi halaman login dan sistem autentikasi menggunakan Laravel Sanctum
Minggu 4	Pembuatan dashboard utama dan navigasi role-based (admin dan karyawan)
Minggu 5	Pengembangan fitur input data karyawan dan validasi
Minggu 6	Pembuatan dan pengujian fitur CRUD data karyawan, divisi, posisi, entitas
Minggu 7	Pengembangan form pengajuan cuti dan sistem persetujuan
Minggu 8	Penambahan fitur tracking status cuti dan filtering data
Minggu 9	Pembuatan form pinjaman, model dan migration tabel pinjaman
Minggu 10	Sistem persetujuan pinjaman dan update status otomatis
Minggu 11	Fitur pengajuan lembur dan struktur data lembur
Minggu 12	Fitur pengajuan pinjam kendaraan dan pengecekan ketersediaan
Minggu 13	Desain UI lebih responsif menggunakan Tailwind CSS
Minggu 14	Pengujian fungsi form dan validasi setiap fitur HR apps
Minggu 15	Perbaikan bug dan optimasi performa aplikasi
Minggu 16	Penambahan fitur pencetakan dokumen dalam PDF
Minggu 17	Finalisasi alur sistem dan penyusunan dokumentasi teknis
Minggu 18	Penyusunan laporan magang dan evaluasi akhir bersama supervisor

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Tabel 3.2. Pekerjaan yang Dilakukan Tiap Minggu Selama Pelaksanaan Kerja Magang

Minggu Ke	Pekerjaan yang Dilakukan
1	Mengikuti briefing, mengenal struktur organisasi, memahami kebutuhan proyek HR
2	Instalasi Laravel, Ionic, React, konfigurasi database lokal dan koneksi GitHub
3	Membuat halaman login, API register/login/logout, implementasi Sanctum
4	Membangun halaman dashboard berdasarkan peran user dan routing role-based
5	Membuat halaman input data karyawan dan validasi field dengan Laravel
6	Menyusun model, controller, migration dan API untuk data jabatan, divisi, posisi
7	Mengembangkan form pengajuan cuti, validasi, dan sistem approval oleh atasan
8	Menambahkan status dinamis dan tampilan riwayat cuti
9	Membuat tabel pinjaman, form pinjaman, field nominal, tujuan, periode cicilan
10	Workflow approval pinjaman dan tampilan histori pinjaman
11	Mendesain alur lembur dan membuat formulir pengajuan lembur karyawan
12	Membuat form peminjaman kendaraan dinas dan fitur pengecekan ketersediaan
13	Refactor halaman menggunakan Tailwind agar lebih responsif dan konsisten
14	Menguji semua form (cuti, pinjaman, lembur, kendaraan) dan validasi field
15	Fixing bug tampilan dan logika, menyempurnakan alur alur approval
16	Implementasi fitur cetak PDF untuk data cuti dan pinjaman
17	Menyusun dokumentasi API dan flowchart alur sistem secara menyeluruh

3.3 Uraian Pelaksanaan Magang

Magang dilaksanakan selama enam bulan di PT Buana Selaras Globalindo. Selama periode tersebut, penulis berfokus pada pengembangan sistem informasi kepegawaian berbasis web yang dirancang untuk memudahkan pengelolaan data karyawan oleh divisi Human Resource (HR). Aplikasi ini dikembangkan sebagai solusi digital yang dapat diakses melalui berbagai perangkat, baik *desktop* maupun *mobile*.

Pengembangan sistem dilakukan dengan memanfaatkan *framework* modern, yaitu **Laravel** sebagai *backend* untuk membangun *REST API*, serta **Ionic Framework** sebagai *frontend* untuk menghasilkan tampilan antarmuka yang responsif dan menyerupai aplikasi *mobile*. Dengan pendekatan *Progressive Web App* (PWA), aplikasi ini mampu memberikan pengalaman pengguna yang optimal di berbagai ukuran layar.

Beberapa fitur utama yang berhasil dikembangkan selama magang meliputi:

1. Manajemen Informasi Karyawan

Modul ini berfungsi untuk menampilkan dan mengelola data pribadi karyawan, seperti nama lengkap, tanggal lahir, nomor telepon, alamat, serta informasi administratif seperti tanggal kontrak, NIK, NPWP, nomor BPJS Kesehatan dan Ketenagakerjaan, serta nomor rekening bank. Seluruh data ini dapat diakses dan diperbarui oleh pihak admin melalui *dashboard* yang disediakan.

2. Fitur Pengajuan Cuti

Karyawan dapat melakukan pengajuan cuti secara daring melalui antarmuka aplikasi. Dalam proses pengajuan, pengguna dapat menentukan jenis cuti, tanggal mulai dan selesai cuti, serta mengisi informasi mengenai karyawan yang ditunjuk sebagai pengganti sementara. Permohonan cuti yang diajukan akan masuk ke sistem dan menunggu persetujuan dari pihak HR. Riwayat cuti juga dapat dilihat oleh karyawan melalui halaman khusus.

3. Fitur Tambahan Pendukung

Untuk melengkapi sistem, penulis juga mengembangkan sejumlah fitur tambahan, seperti:

- *Authentication* pengguna dengan *Laravel Sanctum*.

- Halaman *dashboard* interaktif yang menampilkan data statistik pegawai dan riwayat aktivitas.
- Modul manajemen struktur organisasi yang mencakup divisi, jabatan, dan entitas perusahaan.
- Fitur pengumuman internal yang memungkinkan HR menyebarkan informasi secara langsung kepada seluruh karyawan.

Pengembangan dilakukan melalui tahapan yang sistematis, dimulai dari analisis kebutuhan pengguna, perancangan tampilan menggunakan *Figma*, pembuatan *API backend*, integrasi antarmuka menggunakan *Ionic* dan *Tailwind CSS*, hingga proses pengujian dengan *Postman* serta pengujian langsung oleh pengguna.

Dengan pendekatan teknologi web yang bersifat *mobile-friendly*, aplikasi ini diharapkan dapat menjadi alat bantu yang efektif dalam pengelolaan sumber daya manusia di lingkungan perusahaan secara efisien dan modern.

3.3.1 Tahapan Pengerjaan Aplikasi HR

Pengembangan aplikasi HR dilakukan secara terstruktur sesuai tahapan metode *Waterfall*. Berikut adalah penjabaran tahapan yang dilakukan selama proses magang:

3.3.2 Analisis Kebutuhan

Penulis melakukan observasi terhadap sistem manual yang digunakan oleh tim HR serta wawancara informal dengan staf terkait untuk mengidentifikasi kebutuhan utama.

Kebutuhan Fungsional:

- Formulir input dan edit data karyawan.
- Manajemen divisi, jabatan, dan entitas.
- Sistem login dan *dashboard* admin.
- Pengajuan cuti dan pinjaman.
- Pengumuman internal.

Kebutuhan Non-Fungsional:

- Aplikasi bersifat *responsive*.
- Sistem aman dan stabil.
- Tampilan sederhana dan mudah digunakan.

3.3.3 Perancangan Sistem

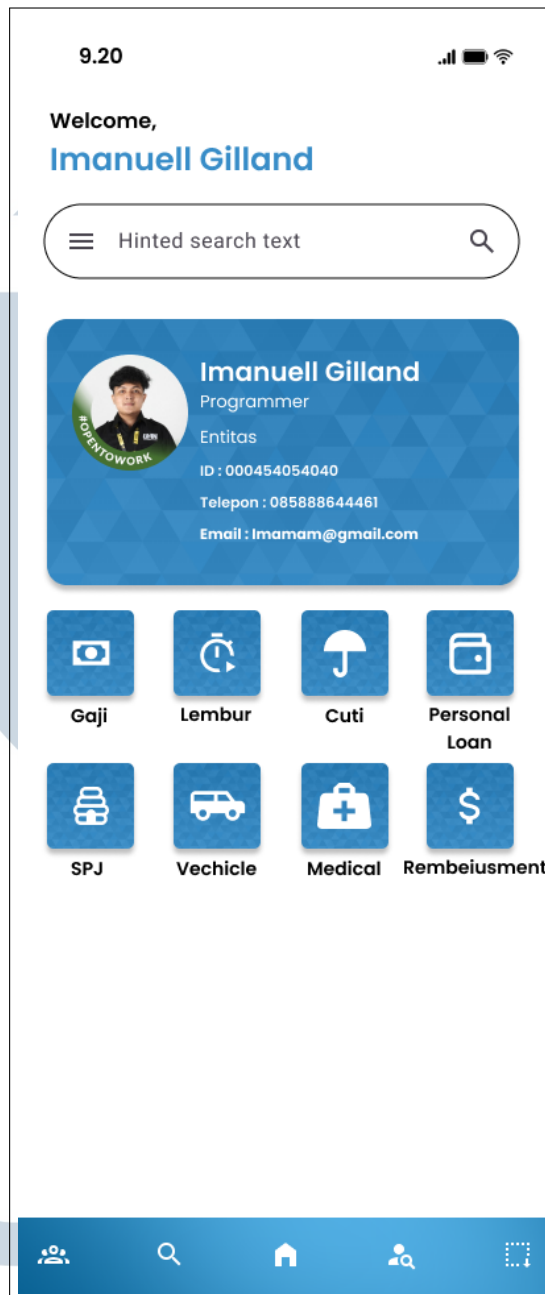
Setelah tahap analisis kebutuhan selesai, tahapan selanjutnya adalah perancangan sistem. Tujuan tahap ini adalah mengubah kebutuhan fungsional menjadi rancangan teknis yang dapat diimplementasikan. Perancangan mencakup tiga aspek utama: antarmuka pengguna (*User Interface*), alur sistem (*flow*), dan struktur basis data (*database*).

A Perancangan Antarmuka (*User Interface*)

Desain antarmuka sistem dilakukan menggunakan **Figma**, yang memungkinkan pembuatan *wireframe* dan *mockup* secara cepat dan kolaboratif. Pendekatan yang digunakan adalah *mobile-first*, karena aplikasi dikembangkan menggunakan **Ionic Framework** yang mengedepankan pengalaman pengguna di perangkat *mobile*, namun tetap optimal di *desktop*.

Desain mempertimbangkan prinsip-prinsip UI/UX seperti konsistensi *layout*, kejelasan elemen navigasi, dan kemudahan interaksi. Beberapa elemen yang dirancang meliputi:

- Halaman login yang sederhana dan aman.
- Formulir input data pegawai yang dibagi dalam beberapa tahap (pribadi, pekerjaan, keluarga).
- *Dashboard* admin yang menampilkan statistik jumlah pegawai, cuti, dan pinjaman.
- Halaman pengajuan cuti dan pinjaman yang intuitif.
- Modul pengumuman internal.



Gambar 3.1. Contoh Desain UI Halaman *Dashboard*

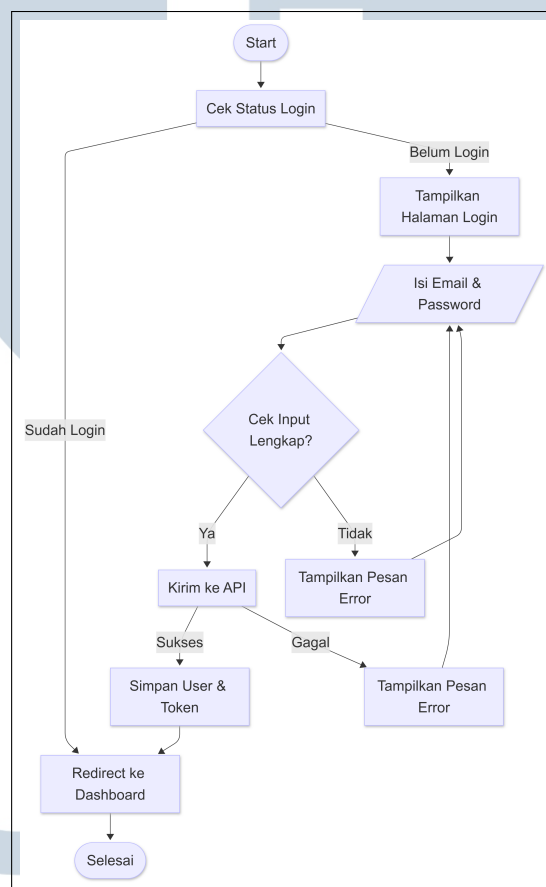
B Perancangan Alur Sistem (*Flowchart* dan Navigasi)

Untuk memastikan alur sistem berjalan secara logis dan efisien, dibuat beberapa *flowchart* yang menggambarkan urutan interaksi pengguna dengan sistem pada berbagai fitur utama. *Flowchart* ini digunakan sebagai acuan dalam implementasi sistem agar proses pengembangan lebih terarah dan kesalahan logika

dapat diminimalkan.

B.1 Alur Login dan Autentikasi Pengguna

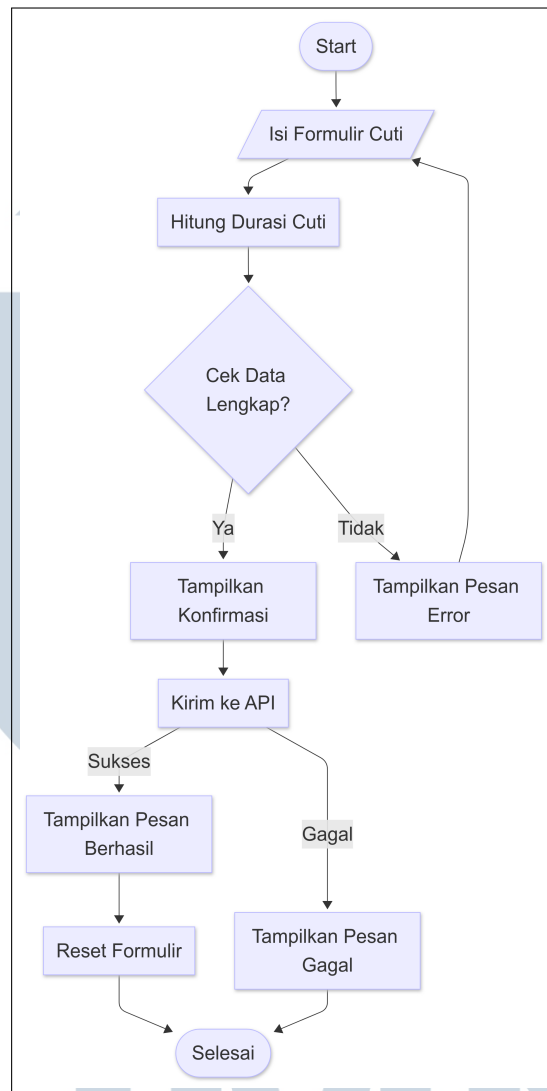
Pengguna memasukkan *email* dan *password*. Sistem melakukan validasi terhadap kredensial. Jika valid, pengguna diarahkan ke halaman *dashboard* sesuai peran (admin, karyawan, atau manajer). Jika tidak valid, sistem memberikan pesan kesalahan.



Gambar 3.2. *Flowchart* Alur Login

B.2 Alur Pengajuan dan Persetujuan Cuti

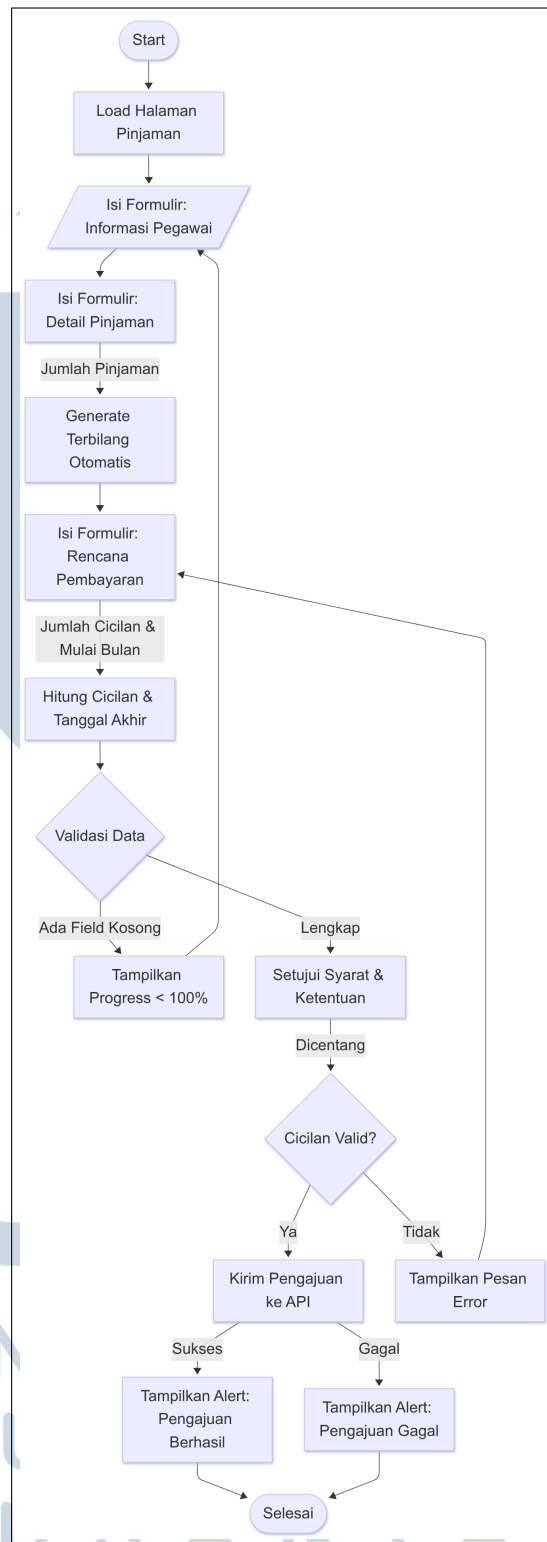
Karyawan mengisi formulir cuti dengan detail tanggal dan alasan. Pengajuan dikirim ke atasan untuk disetujui atau ditolak. Status cuti akan diperbarui dan ditampilkan pada sistem.



Gambar 3.3. *Flowchart* Alur Pengajuan Cuti

B.3 Alur Pengajuan dan Persetujuan Pinjaman

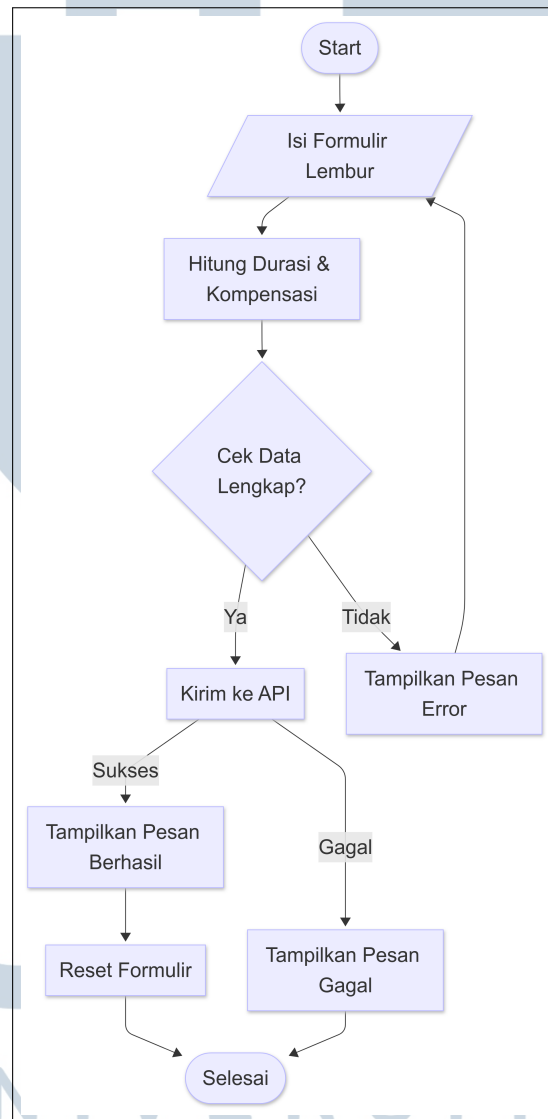
Karyawan mengajukan pinjaman dengan mengisi nominal, alasan, dan periode pengembalian. Sistem mencatat pengajuan dan mengirimkan notifikasi ke pihak HR atau keuangan untuk disetujui.



Gambar 3.4. Flowchart Alur Pengajuan Pinjaman

B.4 Alur Pengajuan dan Persetujuan Lembur

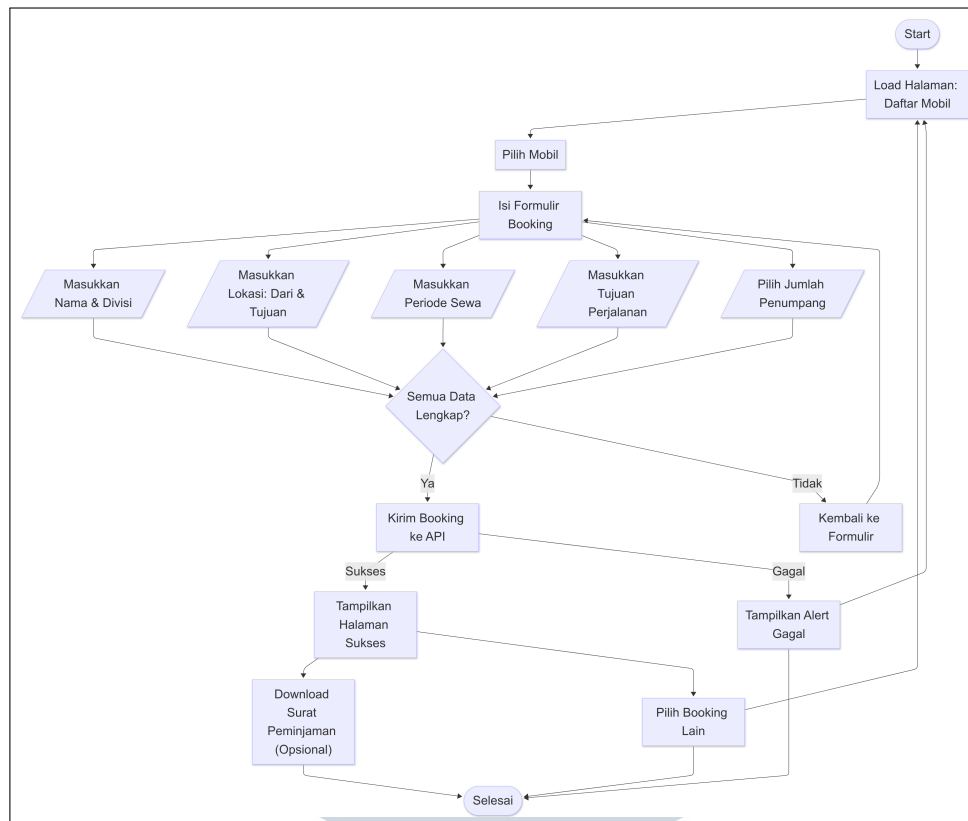
Proses pengajuan lembur dimulai dari permintaan lembur oleh karyawan, kemudian disetujui oleh atasan. Sistem menghitung jumlah jam dan menyimpan data sebagai referensi penggajian atau kompensasi.



Gambar 3.5. Flowchart Alur Pengajuan Lembur

B.5 Alur Pengajuan dan Persetujuan Pinjam Mobil

Karyawan yang memerlukan kendaraan operasional dapat mengajukan permintaan pinjam mobil melalui sistem. Admin akan mengecek ketersediaan kendaraan dan memberikan persetujuan jika memungkinkan.



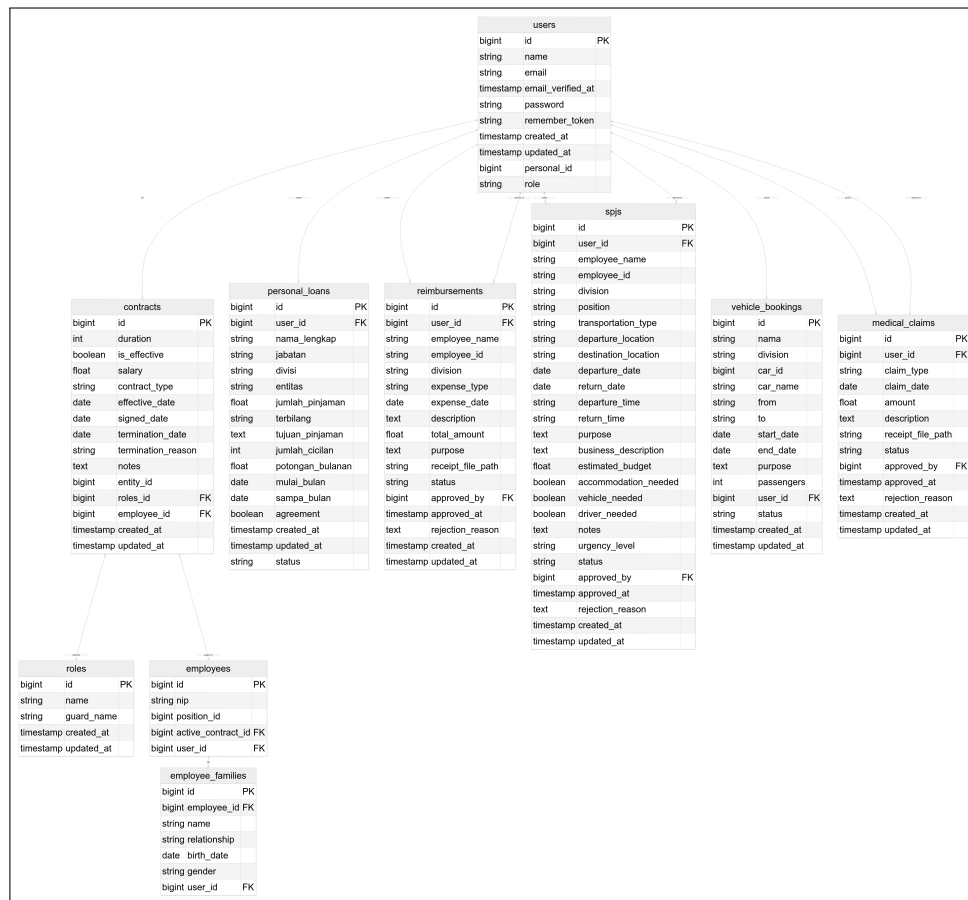
Gambar 3.6. Flowchart Alur Pengajuan Pinjam Mobil

Flowchart di atas memberikan gambaran sistematis terhadap jalur logika aplikasi dan memungkinkan tim pengembang serta *stakeholder* untuk memahami mekanisme kerja sistem. Dengan dokumentasi visual ini, risiko miskomunikasi dan kesalahan implementasi dapat diminimalkan.

C Skema Basis Data

Untuk mendukung fungsi-fungsi utama dalam aplikasi HR, dirancang sebuah skema basis data yang mencakup berbagai entitas dan relasi yang relevan, seperti pengguna, data personal, data karyawan, kontrak kerja, keluarga, pinjaman, lembur, serta permintaan cuti.

Struktur relasional antara tabel-tabel tersebut divisualisasikan dalam bentuk diagram entitas-relasi (ERD) berikut:



Gambar 3.7. Skema Basis Data Aplikasi HR

3.3.4 Tugas yang Dilakukan

Selama masa magang di PT Buana Selaras Globalindo, penulis mengemban berbagai tugas dan tanggung jawab yang mencakup seluruh proses pengembangan aplikasi berbasis web dan *mobile*. Tugas-tugas tersebut dilakukan secara bertahap dan terintegrasi untuk mendukung kebutuhan sistem informasi internal perusahaan. Berikut rincian tugas yang penulis lakukan:

1. Analisis Kebutuhan Sistem

Melakukan identifikasi dan pengumpulan informasi terkait kebutuhan pengguna (*user requirements*) melalui wawancara informal dan observasi langsung terhadap proses bisnis di divisi-divisi terkait, seperti HR, keuangan, dan operasional.

2. Perancangan Antarmuka (UI/UX)

Merancang struktur halaman dan antarmuka pengguna menggunakan sketsa dan *wireframe*. Desain dibuat untuk memastikan kemudahan penggunaan dan pengalaman pengguna (*user experience*) yang optimal.

3. Pengembangan *Frontend* Aplikasi

Mengimplementasikan desain antarmuka menggunakan *React JS* yang dipadukan dengan *Tailwind CSS* untuk membangun tampilan yang responsif, konsisten, dan mudah diakses. Untuk mendukung perangkat *mobile*, penulis menggunakan *Ionic Framework* agar aplikasi dapat berjalan dengan baik di berbagai ukuran layar.

4. Pengembangan *Backend* dan API

Menggunakan *Laravel* untuk mengembangkan logika bisnis dan pengelolaan data. penulis juga membangun *RESTful API* untuk menghubungkan *frontend* dengan *database* secara efisien dan aman.

5. Pengujian dan *Debugging*

Melakukan pengujian fungsionalitas dari setiap modul aplikasi untuk memastikan sistem berjalan sesuai dengan yang diharapkan. Bug atau kesalahan yang ditemukan selama proses pengujian diperbaiki secara berkala.

6. *Deployment* dan Dokumentasi

Melakukan proses *deployment* ke server perusahaan setelah sistem selesai dibangun dan diuji. penulis juga menyusun dokumentasi teknis serta panduan penggunaan aplikasi bagi pengguna akhir.

7. Pemeliharaan dan Penyesuaian Sistem

Melakukan penyesuaian terhadap fitur atau tampilan aplikasi berdasarkan masukan dari pengguna, serta memastikan sistem tetap berjalan dengan stabil dan sesuai kebutuhan operasional.

Tugas-tugas tersebut memberikan pengalaman nyata dalam mengelola proyek pengembangan perangkat lunak secara utuh dan memperkuat kemampuan teknis serta non-teknis yang penulis miliki.

3.3.5 Uraian Pelaksanaan Magang

Magang dilaksanakan selama enam bulan di PT Buana Selaras Globalindo. Selama periode tersebut, penulis berfokus pada pengembangan sistem informasi

kepegawaian berbasis web yang dirancang untuk memudahkan pengelolaan data karyawan oleh divisi Human Resource (HR). Aplikasi ini dikembangkan sebagai solusi digital yang dapat diakses melalui berbagai perangkat, baik *desktop* maupun *mobile*.

Pengembangan sistem dilakukan dengan memanfaatkan *framework* modern, yaitu **Laravel** sebagai *backend* untuk membangun *REST API*, serta **Ionic Framework** sebagai *frontend* untuk menghasilkan tampilan antarmuka yang responsif dan menyerupai aplikasi *mobile*. Dengan pendekatan *Progressive Web App* (PWA), aplikasi ini mampu memberikan pengalaman pengguna yang optimal di berbagai ukuran layar.

Beberapa fitur utama yang berhasil dikembangkan selama magang meliputi:

1. Manajemen Informasi Karyawan

Modul ini berfungsi untuk menampilkan dan mengelola data pribadi karyawan, seperti nama lengkap, tanggal lahir, nomor telepon, alamat, serta informasi administratif seperti tanggal kontrak, NIK, NPWP, nomor BPJS Kesehatan dan Ketenagakerjaan, serta nomor rekening bank. Seluruh data ini dapat diakses dan diperbarui oleh pihak admin melalui *dashboard* yang disediakan.

2. Fitur Pengajuan Cuti

Karyawan dapat melakukan pengajuan cuti secara daring melalui antarmuka aplikasi. Dalam proses pengajuan, pengguna dapat menentukan jenis cuti, tanggal mulai dan selesai cuti, serta mengisi informasi mengenai karyawan yang ditunjuk sebagai pengganti sementara. Permohonan cuti yang diajukan akan masuk ke sistem dan menunggu persetujuan dari pihak HR. Riwayat cuti juga dapat dilihat oleh karyawan melalui halaman khusus.

3. Fitur Tambahan Pendukung

Untuk melengkapi sistem, penulis juga mengembangkan sejumlah fitur tambahan, seperti:

- *Authentication* pengguna dengan *Laravel Sanctum*.
- Halaman *dashboard* interaktif yang menampilkan data statistik pegawai dan riwayat aktivitas.
- Modul manajemen struktur organisasi yang mencakup divisi, jabatan, dan entitas perusahaan.

- Fitur pengumuman internal yang memungkinkan HR menyebarkan informasi secara langsung kepada seluruh karyawan.

Pengembangan dilakukan melalui tahapan yang sistematis, dimulai dari analisis kebutuhan pengguna, perancangan tampilan menggunakan *Figma*, pembuatan *API backend*, integrasi antarmuka menggunakan *Ionic* dan *Tailwind CSS*, hingga proses pengujian dengan *Postman* serta pengujian langsung oleh pengguna.

Dengan pendekatan teknologi web yang bersifat *mobile-friendly*, aplikasi ini diharapkan dapat menjadi alat bantu yang efektif dalam pengelolaan sumber daya manusia di lingkungan perusahaan secara efisien dan modern.

3.3.6 Metodologi Pengembangan

Dalam pengembangan aplikasi HR, metode yang digunakan adalah **Waterfall**. Metode ini merupakan model pengembangan perangkat lunak yang bersifat linear dan berurutan, di mana setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya. Model ini dipilih karena kebutuhan sistem telah cukup jelas sejak awal dan pengembangan dilakukan secara bertahap dari analisis hingga pengujian.

Tahapan-tahapan dalam metode *Waterfall* meliputi:

1. **Analisis Kebutuhan**

Pengumpulan kebutuhan pengguna dan sistem melalui observasi dan wawancara.

2. **Perancangan Sistem**

Desain struktur aplikasi, antarmuka pengguna, serta perancangan *database*.

3. **Implementasi**

Proses penulisan kode dan integrasi sistem *frontend* dan *backend*.

4. **Pengujian**

Pengujian fungsional sistem terhadap kebutuhan dan spesifikasi.

5. **Pemeliharaan**

Penyesuaian dan perbaikan sistem pasca implementasi.

Metode Waterfall ini dipilih berdasarkan arahan dari manajer pengembangan sistem di PT Buana Selaras Globalindo yang menilai bahwa kebutuhan sistem telah terdefinisi dengan baik sejak awal.

3.3.7 Pengujian Sistem

Pengujian sistem merupakan tahap penting dalam pengembangan aplikasi HR untuk memastikan bahwa sistem memenuhi kebutuhan fungsional dan non-fungsional yang telah ditetapkan. Pengujian dilakukan menggunakan pendekatan **black-box testing**, yang berfokus pada kesesuaian hasil *output* terhadap *input* yang diberikan tanpa memeriksa kode internal aplikasi. Pendekatan ini dipilih karena memungkinkan pengujian dari perspektif pengguna akhir, sehingga memastikan bahwa sistem dapat digunakan dengan baik oleh pengguna tanpa memerlukan pengetahuan teknis tentang implementasi internal.

Pengujian dilakukan secara iteratif sepanjang siklus pengembangan, dengan setiap iterasi mencakup pengujian fungsional, pengujian antarmuka, dan pengujian performa untuk memastikan sistem stabil di berbagai kondisi. Berikut adalah contoh skenario pengujian yang dilakukan menggunakan pendekatan *black-box testing*:

No	Skenario Uji	Input	Output Diharapkan
1	Isi form pegawai dengan valid	Semua data diisi	Data berhasil disimpan
2	Form kosong (nama tidak diisi)	Nama: kosong	Error: “Nama wajib diisi”
3	Login dengan akun salah	Email tidak terdaftar	Error: “Login gagal”
4	Ajukan pinjaman valid	Jumlah, tujuan terisi	Berhasil disimpan di <i>database</i>
5	Pengajuan cuti dengan tanggal tidak valid	Tanggal selesai sebelum tanggal mulai	Error: “Tanggal cuti tidak valid”
6	Akses dashboard tanpa login	Tidak ada sesi aktif	Redirect ke halaman login
7	Pengajuan lembur dengan jam tidak valid	Jumlah jam: -1	Error: “Jumlah jam tidak valid”

Tabel 3.3. Contoh Pengujian *Black-box Testing*

Pengujian *RESTful API* dilakukan menggunakan **Postman** untuk membuat dan menjalankan *test cases* terhadap setiap *endpoint* API. Pengujian API mencakup verifikasi respons HTTP (misalnya, status kode 200 untuk sukses, 400 untuk kesalahan input, atau 401 untuk akses tidak sah), format data JSON, serta kebenaran

data yang dikembalikan. Sebagai contoh, pengujian *endpoint* untuk pengajuan cuti memastikan bahwa data yang dikirim dari *frontend* tersimpan dengan benar di tabel `leave_requests` dan menghasilkan respons sukses dengan pesan konfirmasi. Pengujian juga mencakup skenario kegagalan, seperti pengiriman data tanpa token autentikasi, yang diharapkan menghasilkan error 401.

Hasil pengujian menunjukkan bahwa aplikasi memenuhi 95% kebutuhan fungsional, dengan beberapa masalah kecil seperti penundaan respons API pada beban tinggi dan kesalahan kecil pada antarmuka pengguna di perangkat tertentu (misalnya, Safari versi lama). Masalah-masalah ini didokumentasikan dalam laporan pengujian dan dijadwalkan untuk diperbaiki pada tahap pemeliharaan. Dokumentasi hasil pengujian disusun dalam format yang jelas, mencakup metrik keberhasilan, daftar bug, dan rekomendasi perbaikan, untuk memudahkan tim perusahaan dalam memahami performa sistem dan merencanakan pengembangan lebih lanjut.

Pengujian sistem ini tidak hanya memastikan kualitas aplikasi, tetapi juga memberikan wawasan berharga tentang pentingnya pengujian berulang, kolaborasi dengan pengguna akhir, dan optimalisasi performa untuk menghasilkan sistem yang andal, efisien, dan ramah pengguna.

3.3.8 Pengembangan Sistem

Setelah tahap perancangan dan pengujian selesai dilakukan, sistem HR dikembangkan secara bertahap berdasarkan hasil evaluasi, kebutuhan pengguna, serta pengujian fungsional dan antarmuka yang telah dilakukan sebelumnya.

Pengembangan sistem melibatkan implementasi berbagai fitur utama untuk mendukung proses administrasi dan manajemen data karyawan. Aplikasi dikembangkan menggunakan Laravel untuk backend, serta React + Ionic + Tailwind CSS untuk frontend.

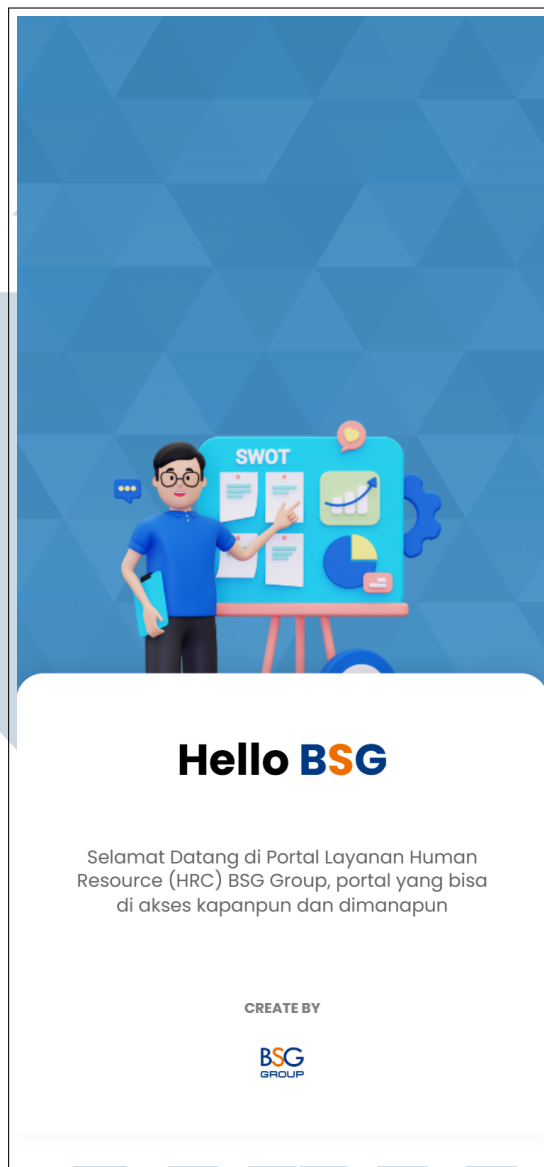
Struktur halaman utama dalam sistem dijelaskan pada bagian-bagian berikut:

A Homepage

Halaman awal atau *homepage* merupakan tampilan pembuka dari aplikasi HR BSG Group yang berfungsi sebagai pengenalan sistem kepada pengguna. Pada halaman ini ditampilkan nama aplikasi, sambutan singkat “Hello BSG”,

serta deskripsi yang menjelaskan bahwa portal ini merupakan layanan Human Resource (HRC) yang dapat diakses kapanpun dan di manapun. Desain halaman dibuat profesional dan modern dengan ilustrasi karakter yang sedang menjelaskan analisis SWOT di depan papan presentasi, menambah daya tarik visual sekaligus menggambarkan semangat kerja dan perencanaan. Warna biru dominan dan penggunaan font yang konsisten memberikan kesan korporat dan bersih. Di bagian bawah, terdapat informasi “Create by BSG Group” beserta logo perusahaan sebagai bentuk identitas pengembang. Secara keseluruhan, halaman ini menjadi titik awal yang informatif dan ramah pengguna sebelum melanjutkan ke proses login dan fitur-fitur lainnya.



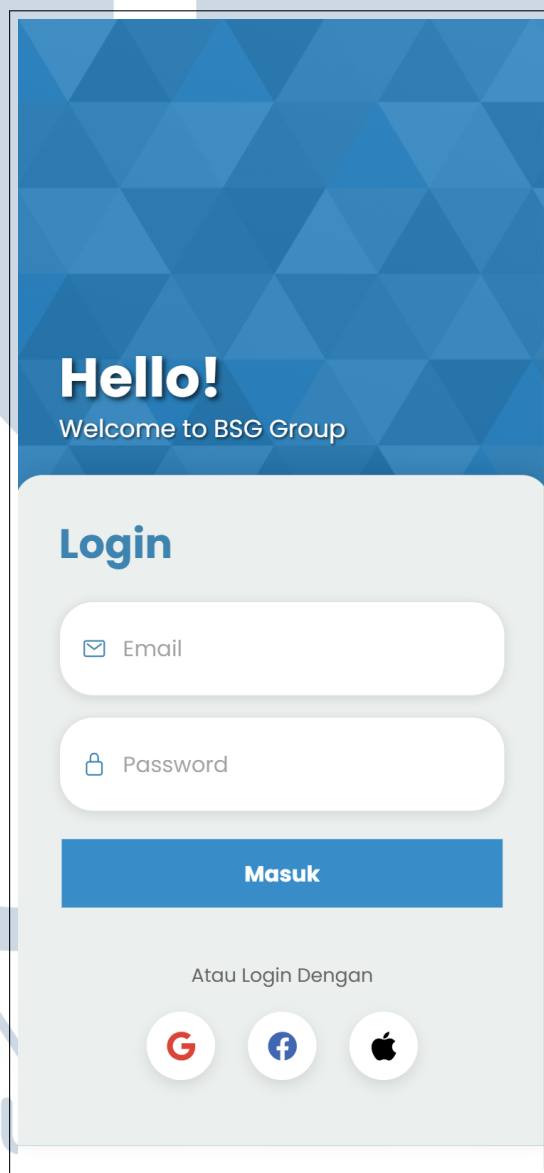


Gambar 3.8. Tampilan Homepage Aplikasi

B Login Page

Halaman *login* merupakan gerbang utama untuk mengakses sistem dan hanya dapat digunakan oleh pengguna yang telah memiliki akun resmi. Pada halaman ini, pengguna diminta untuk mengisi email dan kata sandi sebagai bentuk kredensial untuk proses autentikasi. Sistem menggunakan Laravel Sanctum dengan metode token-based authentication untuk memastikan keamanan dan validitas identitas pengguna, sekaligus melindungi sistem dari akses tidak sah. Apabila proses login berhasil, pengguna akan langsung diarahkan ke halaman *dashboard*

sesuai dengan peran dan hak akses yang telah ditentukan, seperti admin, HRD, atau karyawan. Desain halaman login dibuat minimalis, responsif, dan fokus pada kemudahan penggunaan, dengan tampilan input yang jelas serta validasi otomatis yang memberikan notifikasi apabila terjadi kesalahan, seperti email tidak ditemukan atau kata sandi salah. Hal ini bertujuan untuk meningkatkan kenyamanan pengguna serta menjaga integritas data sistem sejak tahap awal akses.



Hello!
Welcome to BSG Group

Login

Email

Password

Masuk

Atau Login Dengan

Google, Facebook, Apple

Gambar 3.9. Tampilan Halaman Login

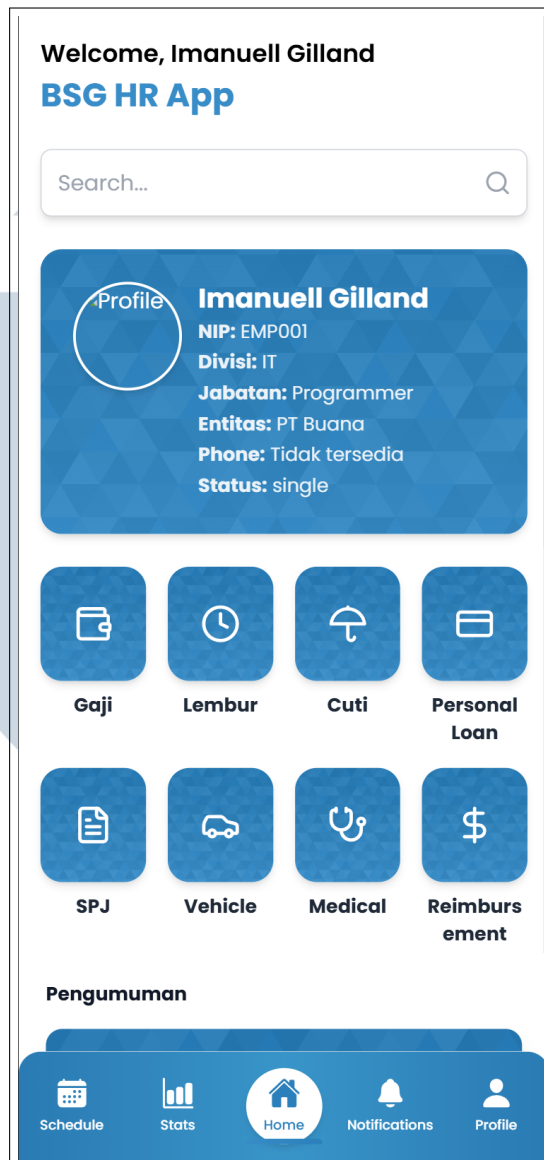
C Dashboard

Setelah berhasil login, pengguna diarahkan ke halaman *dashboard* yang berfungsi sebagai pusat informasi utama dalam aplikasi. Pada halaman ini, pengguna disambut dengan pesan personalisasi seperti “Welcome, [Nama Pengguna]” yang secara langsung menciptakan pengalaman yang lebih interaktif dan ramah. Di bawahnya, terdapat nama aplikasi “BSG HR App” yang mempertegas identitas sistem. Salah satu fitur penting yang langsung terlihat adalah kartu profil yang menampilkan informasi personal pengguna, seperti NIP, divisi, jabatan, entitas, nomor telepon, dan status pernikahan. Informasi ini ditampilkan secara ringkas namun jelas untuk memberikan gambaran identitas pengguna secara langsung tanpa perlu membuka halaman profil secara terpisah.

Dashboard ini juga menampilkan berbagai fitur utama dalam bentuk ikon-ikon besar dan jelas, yang masing-masing mewakili layanan HR penting seperti Gaji, Lembur, Cuti, Personal Loan, SPJ, Kendaraan, Layanan Medis, dan Reimbursement. Setiap ikon menggunakan latar biru bermotif khas yang konsisten dengan desain visual aplikasi secara keseluruhan, serta dilengkapi dengan label teks yang memudahkan identifikasi. Desain ini memudahkan pengguna untuk langsung mengakses layanan tertentu hanya dengan sekali klik.

Di bagian bawah terdapat menu navigasi tetap (fixed navigation bar) dengan lima ikon utama: Schedule, Stats, Home, Notifications, dan Profile. Navigasi ini memudahkan perpindahan antar halaman tanpa harus kembali ke menu utama, sehingga meningkatkan efisiensi penggunaan. Komponen UI seluruhnya dibangun dengan Tailwind CSS dan framework Ionic untuk memastikan tampilan responsif, modern, dan ramah pengguna di berbagai ukuran perangkat.

Secara keseluruhan, halaman *dashboard* dirancang tidak hanya sebagai pusat informasi, tetapi juga sebagai titik interaksi awal bagi pengguna dalam mengakses fitur-fitur HR secara praktis dan terorganisir.



Gambar 3.10. Tampilan Halaman Dashboard

C.1 Pengumuman

Fitur pengumuman digunakan untuk menampilkan informasi terbaru dari pihak manajemen atau HRD kepada seluruh karyawan. Informasi yang disampaikan dapat berupa pemberitahuan cuti bersama, perubahan kebijakan perusahaan, hingga agenda internal lainnya. Pengumuman ini tampil langsung di bawah menu utama dashboard agar mudah terlihat oleh semua pengguna.



Gambar 3.11. Tampilan Fitur Pengumuman

C.2 Personal Loan

Menu *Personal Loan* merupakan fitur yang disediakan untuk memfasilitasi karyawan dalam mengajukan pinjaman pribadi kepada perusahaan. Fitur ini bertujuan untuk memberikan bantuan keuangan internal yang dapat digunakan oleh karyawan dalam keadaan tertentu, seperti keperluan mendesak, biaya kesehatan, renovasi rumah, pendidikan, dan sebagainya. Pada halaman ini, karyawan diminta untuk mengisi formulir pengajuan pinjaman dengan informasi yang lengkap, seperti jumlah nominal pinjaman yang diajukan, tujuan atau keperluan pinjaman, jumlah cicilan per bulan, serta periode pemotongan gaji yang diinginkan.

Setelah data pengajuan diisi dan dikirimkan, sistem akan mencatat permohonan dan mengarahkan prosesnya ke jalur persetujuan internal. Biasanya, permohonan akan diperiksa oleh atasan langsung terlebih dahulu, lalu diteruskan ke bagian HRD dan keuangan untuk diverifikasi serta disetujui. Proses ini juga mempertimbangkan histori pinjaman sebelumnya, masa kerja, dan status kepegawaian. Jika disetujui, maka pinjaman akan dicairkan sesuai ketentuan perusahaan dan cicilan akan mulai dipotong dari gaji karyawan sesuai periode yang telah disepakati.

Tampilan fitur ini dibuat sederhana namun informatif, sehingga karyawan dapat dengan mudah memahami dan mengisi formulir secara mandiri. Selain itu, status pengajuan pinjaman dapat dipantau secara real-time melalui sistem, sehingga transparansi proses dapat terjaga. Dengan adanya fitur *Personal Loan*, perusahaan menunjukkan kepeduliannya terhadap kesejahteraan karyawan sekaligus menjaga

keteraturan administrasi keuangan.

Personal Loan Appl...

BSG

BSG Group

Smart Personal Loan Application System

Auto-Calculate Feature Enabled

Form Permohonan Pinjaman (Personal Loan)

Kode: **BSG-PL-2025-769411**

Detail Pemohon

Yang bertanda tangan di bawah ini:

Nama Lengkap *

Masukkan nama lengkap

Gambar 3.12. Tampilan Fitur Personal Loan (1) pada Dashboard

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Detail Pinjaman

Dengan ini mengajukan pinjaman (personal loan) dengan jumlah sebesar:

Jumlah Pinjaman (Rp) *

Masukkan jumlah pinja

Terbilang

Tujuan Pinjaman *

Jelaskan tujuan pinjaman secara detail...

Gambar 3.13. Tampilan Fitur Personal Loan (2) pada Dashboard

Rencana Pembayaran

Untuk melunasi pinjaman tersebut, saya bersedia dipotong gaji setiap bulan sebesar:

Jumlah Cicilan (Bulan) *

Masukkan jumlah cicilan

Potongan Bulanan (Rp) *

Masukkan jumlah potoi

Mulai Bulan *

April 2024
May 2024
June 2024
July
August

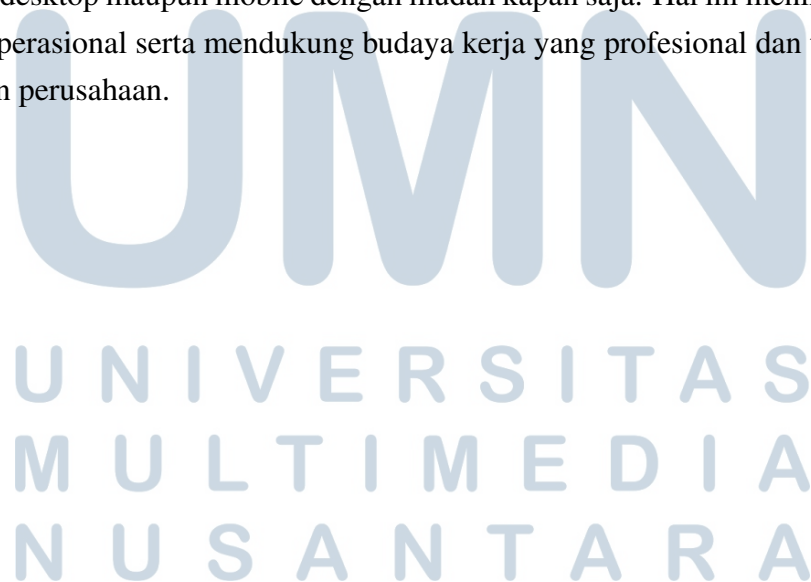
Sampai Bulan (Auto-Generated)

Gambar 3.14. Tampilan Fitur Personal Loan (3) pada Dashboard

C.3 Overtime

Fitur *Overtime* atau lembur merupakan salah satu komponen penting dalam sistem HR karena memberikan fleksibilitas kepada karyawan untuk mengajukan jam kerja tambahan di luar jam kerja normal. Fitur ini sangat bermanfaat dalam kondisi tertentu, seperti ketika terdapat tugas mendesak yang harus diselesaikan di luar waktu kerja atau ketika karyawan perlu hadir di luar hari kerja reguler. Melalui fitur ini, karyawan dapat melakukan pengajuan dengan mengisi form yang mencakup tanggal lembur, jam mulai dan jam selesai, serta memberikan deskripsi atau alasan lembur secara jelas. Sistem secara otomatis akan menghitung durasi lembur yang diajukan dan mencatatnya dalam basis data, sekaligus mengirimkan notifikasi ke atasan langsung untuk proses persetujuan.

Setelah diajukan, status lembur akan diperbarui secara real-time, sehingga karyawan dapat memantau apakah permohonannya disetujui, ditolak, atau masih dalam proses. Data lembur yang disetujui nantinya akan digunakan oleh bagian HR untuk keperluan perhitungan insentif lembur, pelaporan keuangan, atau dokumentasi jam kerja. Dengan adanya fitur ini, seluruh proses permintaan lembur menjadi lebih efisien, transparan, dan terdokumentasi dengan baik tanpa perlu pengajuan manual atau formulir kertas. Tampilan antar muka (UI) dibuat sederhana dan responsif sehingga karyawan dapat melakukan pengajuan lembur baik melalui perangkat desktop maupun mobile dengan mudah kapan saja. Hal ini meningkatkan efisiensi operasional serta mendukung budaya kerja yang profesional dan teratur di lingkungan perusahaan.



Overtime Applicati...

BSG Group

Smart Overtime Application System

Auto-Calculate Compensation

Form Pengajuan Lembur (Overtime)

Kode: **BSG-OT-2025-326628**

Detail Pemohon
Yang bertanda tangan di bawah ini:

Nama Lengkap *
Masukkan nama lengkap

Jabatan *
Masukkan jabatan

Divisi *
Pilih divisi

Gambar 3.15. Tampilan Fitur Form Lembur (1)

UMN

UNIVERSITAS

MULTIMEDIA

NUSANTARA

Detail Lembur

Dengan ini mengajukan lembur dengan detail sebagai berikut:

Tanggal Lembur *

June 2025

SUN	MO	TUE	WED	THU	FRI	SAT
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Jenis Lembur *

Pilih jenis lembur

Jam Mulai *

7:57

Gambar 3.16. Tampilan Fitur Form Lembur (2)

UMN

UNIVERSITAS

MULTIMEDIA

NUSANTARA



Gambar 3.17. Tampilan Fitur Form Lembur (3)

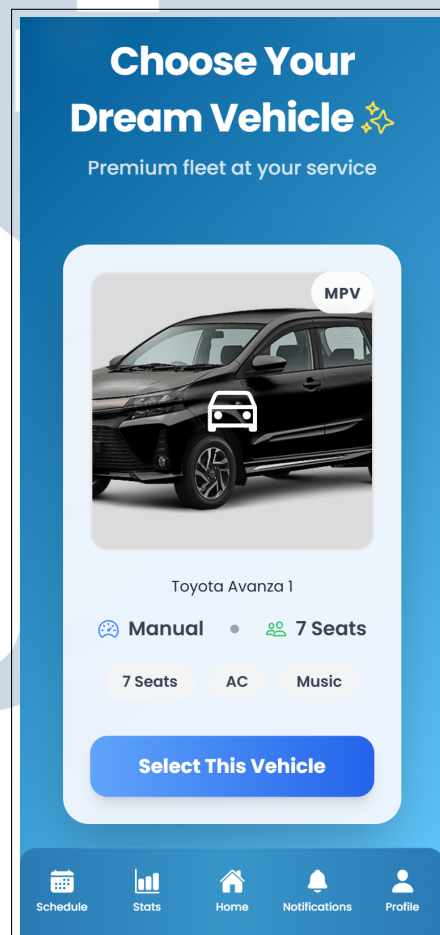
C.4 Vehicle

Menu *Vehicle* merupakan fitur yang digunakan untuk mengatur peminjaman kendaraan operasional perusahaan oleh karyawan. Fitur ini dirancang agar proses permintaan kendaraan menjadi lebih tertib, efisien, dan terdokumentasi secara digital. Pada halaman ini, karyawan terlebih dahulu diminta untuk memilih jenis kendaraan yang tersedia, seperti mobil dinas A, mobil operasional B, atau kendaraan khusus lainnya yang telah didaftarkan dalam sistem. Daftar kendaraan ini ditampilkan secara real-time sesuai dengan ketersediaan dan jadwal pemakaian sebelumnya, sehingga pengguna dapat memilih kendaraan yang belum dipesan pada waktu yang sama.

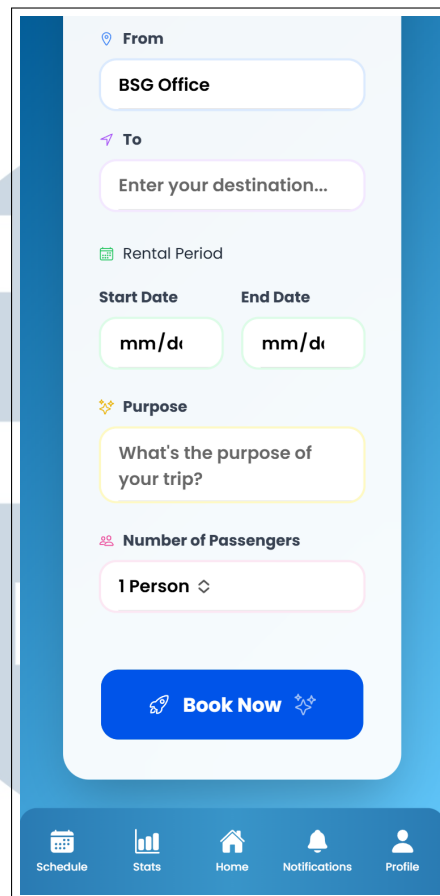
Setelah kendaraan dipilih, barulah pengguna diarahkan ke halaman form peminjaman. Form ini mencakup informasi seperti tanggal dan jam penggunaan, estimasi durasi, tujuan peminjaman, serta keperluan atau deskripsi singkat

perjalanan. Setelah form dilengkapi dan dikirimkan, sistem akan mencatat pengajuan tersebut dan meneruskannya kepada pihak HRD atau bagian yang berwenang untuk direview dan disetujui. Proses ini memastikan bahwa kendaraan operasional digunakan sesuai prosedur dan tidak terjadi tumpang tindih dalam penggunaan kendaraan.

Semua riwayat peminjaman, status persetujuan, serta catatan penggunaan kendaraan akan tersimpan di sistem secara terstruktur, sehingga memudahkan monitoring dan pelaporan. Fitur ini tidak hanya membantu dalam mengelola logistik operasional, tetapi juga mencerminkan sistem kerja yang tertib dan berbasis data. Tampilan antarmuka menu *Vehicle* dibuat intuitif dan responsif agar karyawan dapat melakukan proses peminjaman dengan mudah melalui perangkat apa pun.



Gambar 3.18. Tampilan Fitur Vehicle untuk Peminjaman Kendaraan (1)



Gambar 3.19. Tampilan Fitur Vehicle untuk Peminjaman Kendaraan (2)

3.3.9 Cuti (Leave)

Menu Cuti merupakan salah satu fitur penting dalam sistem manajemen kepegawaian yang berfungsi untuk mengelola permohonan cuti karyawan. Fitur ini memungkinkan karyawan untuk mengajukan cuti secara digital dengan lebih mudah, cepat, dan terstruktur. Proses ini mencakup pemilihan jenis cuti (seperti cuti tahunan, cuti sakit, cuti melahirkan, dan cuti lainnya), pengisian tanggal mulai dan berakhir cuti, serta keterangan tambahan sebagai alasan pengajuan cuti.

Setelah pengajuan dikirimkan, sistem akan mencatat permohonan tersebut dan secara otomatis mengirimkannya ke atasan atau HRD untuk diverifikasi dan disetujui. Fitur ini juga dilengkapi dengan riwayat pengajuan cuti, status permohonan (dalam proses, disetujui, atau ditolak), serta penghitungan sisa cuti tahunan yang tersedia bagi setiap karyawan.

Dengan sistem cuti yang terintegrasi ini, proses administratif menjadi lebih efisien, mengurangi penggunaan dokumen fisik, serta meminimalisir

risiko kesalahan atau kehilangan data cuti karyawan. Selain itu, fitur ini mendukung transparansi antara karyawan dan manajemen mengenai hak cuti dan penggunaannya.

Form Pengajuan Cuti

Nama Lengkap

Jabatan

Divisi

Jenis Cuti

Keperluan/Alasan

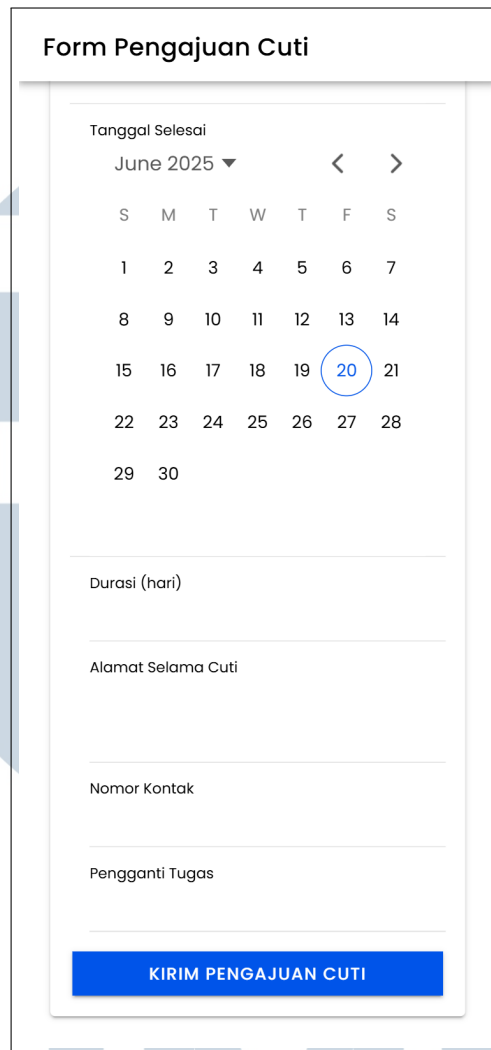
Tanggal Mulai

June 2025

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Gambar 3.20. Tampilan Fitur Cuti dalam Aplikasi HR (1)

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Form Pengajuan Cuti

Tanggal Selesai

June 2025

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Durasi (hari)

Alamat Selama Cuti

Nomor Kontak

Pengganti Tugas

KIRIM PENGAJUAN CUTI

Gambar 3.21. Tampilan Fitur Cuti dalam Aplikasi HR (2)

3.3.10 Profile

Menu **Profile** berfungsi sebagai pusat informasi dan pengaturan akun pengguna dalam sistem HR. Pada halaman ini, pengguna dapat melihat identitas akun seperti nama lengkap dan alamat email yang terdaftar. Desain antarmuka dibuat intuitif dan modern untuk memudahkan navigasi pengguna.

Terdapat beberapa submenu penting dalam halaman ini, di antaranya:

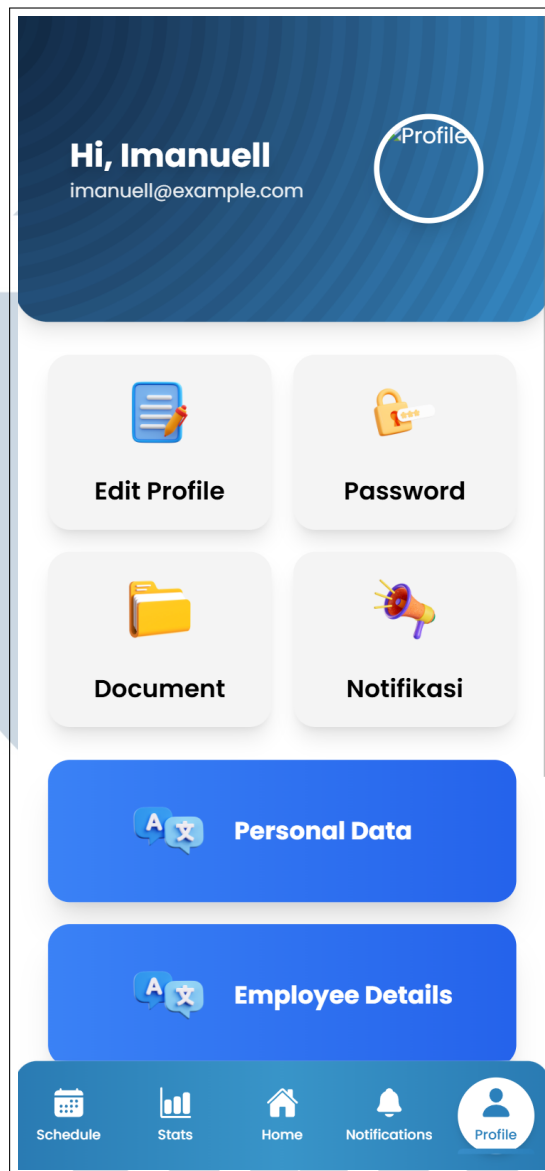
- **Edit Profile** — Fitur ini memungkinkan pengguna untuk memperbarui informasi pribadi mereka, seperti nama, alamat email, nomor telepon, atau foto profil.
- **Password** — Digunakan untuk mengganti kata sandi akun dengan sistem

keamanan tertentu, misalnya validasi password lama dan minimum panjang karakter.

- **Document** — Fitur untuk melihat dan mengelola dokumen yang telah diunggah oleh pengguna atau HR, seperti slip gaji, surat cuti, dan dokumen kepegawaian lainnya.
- **Notifikasi** — Menampilkan semua pemberitahuan penting yang berkaitan dengan status pekerjaan, pengajuan cuti, pinjaman, dan pengumuman perusahaan.
- **Personal Data** — Berisi data pribadi lengkap pengguna yang sebelumnya diinput ke dalam sistem, seperti NIK, tempat/tanggal lahir, status pernikahan, dan alamat tinggal.
- **Employee Details** — Berisi data pekerjaan pengguna di perusahaan, seperti jabatan, departemen, entitas kerja, dan tanggal bergabung.

Tampilan antarmuka dari menu *Profile* disajikan dengan gaya yang bersih dan responsif, memudahkan pengguna untuk mengakses informasi penting kapan saja melalui perangkat mobile maupun desktop.





Gambar 3.22. Tampilan Halaman Profile pada Aplikasi HR

3.4 Kendala dan Solusi yang Ditemukan

Selama proses pengembangan aplikasi HR, berbagai kendala teknis maupun non-teknis muncul. Untuk memberikan gambaran yang lebih sistematis, bagian ini dibagi menjadi dua subbagian: **Kendala** dan **Solusi**.

3.4.1 Kendala dalam Pengembangan Sistem

Berikut adalah kendala-kendala utama yang dihadapi selama pengembangan sistem:

1. **Integrasi Frontend dan Backend** Proses penghubungan frontend berbasis Ionic React dengan backend Laravel mengalami hambatan, terutama pada autentikasi yang membutuhkan token CSRF dan penanganan CORS yang sesuai.
2. **Kompleksitas Struktur Data** Jumlah tabel dan relasi data yang kompleks, seperti data pribadi, pekerjaan, keluarga, cuti, dan pinjaman, menyebabkan pengelolaan database menjadi rumit.
3. **Kesulitan Deployment di CPanel** Penyesuaian struktur Laravel pada environment shared hosting (CPanel) menyebabkan error seperti path yang tidak sesuai, masalah izin file, dan file `.env` yang tidak terbaca.
4. **Ketidakkonsistenan UI di Berbagai Perangkat** Komponen UI tidak ditampilkan secara konsisten pada resolusi layar tertentu, terutama pada perangkat Android dengan resolusi rendah.
5. **Validasi dan Keamanan Data yang Kurang Optimal** Formulir seperti cuti dan pinjaman rentan terhadap input tidak valid karena kurangnya validasi yang memadai di sisi frontend dan backend.
6. **Kesulitan Pengujian pada Perangkat Fisik** Fitur seperti upload gambar atau akses kamera menunjukkan perilaku berbeda antara browser dan perangkat Android fisik selama pengujian.
7. **Tantangan Manajemen Waktu** Penumpukan tugas antara pengembangan frontend, backend, dokumentasi, dan pengujian menyebabkan alokasi waktu yang tidak seimbang.
8. **Ketergantungan pada Koneksi Internet** Proses instalasi library atau pembaruan sistem sangat bergantung pada koneksi internet yang stabil.
9. **Inkoherensi Desain UI/UX** Tidak adanya design system di awal menyebabkan ketidaksesuaian gaya antar halaman aplikasi.

10. **Error Tidak Terduga Saat Pengujian** Error acak muncul saat pengguna mengakses sistem secara bersamaan, seperti respons API yang terhenti atau logout otomatis.

3.4.2 Solusi untuk Mengatasi Kendala

1. **Integrasi Frontend dan Backend** Konfigurasi `cors.php` disesuaikan, endpoint `/sanctum/csrf-cookie` dipanggil, dan proxy digunakan untuk pengembangan lokal agar cookie dan token berfungsi dengan baik.
2. **Kompleksitas Struktur Data** Struktur database didesain ulang menggunakan relasi Eloquent seperti `hasMany`, `belongsTo`, dan `hasOne`, serta menerapkan eager loading untuk meningkatkan efisiensi query.
3. **Kesulitan Deployment di CPanel** Folder `public_html` diarahkan ke folder `public` Laravel, file `index.php` disesuaikan, dan izin file diatur secara manual melalui File Manager CPanel.
4. **Ketidakkonsistenan UI di Berbagai Perangkat** Komponen UI disesuaikan dengan Tailwind CSS menggunakan unit responsif, diuji pada emulator Android Studio, dan divalidasi pada perangkat fisik.
5. **Validasi dan Keamanan Data yang Kurang Optimal** Laravel FormRequest diterapkan untuk validasi backend, ditambah validasi real-time di frontend dengan pesan error yang informatif.
6. **Kesulitan Pengujian pada Perangkat Fisik** Capacitor dan plugin Ionic Native digunakan untuk akses kamera dan file, dengan pembuatan APK untuk pengujian langsung di perangkat Android.
7. **Tantangan Manajemen Waktu** Metode Waterfall diterapkan dengan jadwal kerja mingguan yang terstruktur, didukung oleh task board seperti Jira atau Trello.
8. **Ketergantungan pada Koneksi Internet** Pengembangan dijadwalkan pada waktu dengan koneksi stabil, dan cache dependency disimpan untuk mendukung kerja offline.

9. **Inkoherensi Desain UI/UX** Design system dibuat di Figma, dan komponen reusable dikembangkan menggunakan React dan Tailwind CSS untuk menjaga konsistensi antarmuka.
10. **Error Tidak Terduga Saat Pengujian** Laravel Log dan Telescope diaktifkan untuk debugging, serta try-catch ditambahkan di frontend untuk menampilkan pesan error yang jelas kepada pengguna.

