

BAB 3

Pelaksanaan Magang

3.1 Kedudukan dan Koordinasi

Pelaksanaan kerja magang dilakukan pada divisi *IT & Multimedia* sebagai *IT Staff* dengan peran utama sebagai *Fullstack Developer*. Selama masa magang, penulis berada di bawah arahan Lukmanul Hakim, S.T., M.Sc. selaku *Manager IT & Multimedia*, dan mendapatkan bimbingan langsung dari Muhammad Nurkhaliq, S.Kom selaku mentor teknis. Tugas yang dikerjakan meliputi pengembangan aplikasi berbasis web secara menyeluruh, termasuk *frontend* dan *backend*, pembuatan dan manajemen basis data, integrasi *API*, serta dokumentasi teknis sistem. Penulis juga turut berkontribusi dalam proses *debugging*, optimalisasi performa, hingga *deployment* aplikasi internal. Kegiatan magang ini dilaksanakan secara *full Work From Office (WFO)* di kantor pusat *Arraya Holding*, sehingga seluruh proses koordinasi dilakukan secara langsung bersama tim *IT & Multimedia*. Tim ini juga secara aktif berkolaborasi dengan divisi-divisi lain seperti *HR* untuk memastikan sistem yang dikembangkan mampu memenuhi kebutuhan operasional lintas departemen di lingkungan perusahaan.

3.2 Tugas yang Dilakukan

Kegiatan magang yang dilakukan oleh penulis difokuskan pada pengembangan aplikasi sistem absensi karyawan berbasis web yang diberi nama *Arraya Attendance*. Penulis berperan sebagai *Fullstack Developer* dalam tim *IT & Multimedia* dan bertanggung jawab pada sisi *frontend* maupun *backend* dari sistem yang dibangun.

Tugas ini dilakukan secara bertahap, yang dijelaskan secara rinci dalam tiga tahap berikut:

1. **Research**

Pada tahap ini, penulis melakukan riset terhadap kebutuhan-kebutuhan aplikasi yang akan dikembangkan. Hal ini meliputi pemahaman terhadap alur absen, identifikasi kebutuhan pengguna, serta studi terhadap teknologi dan arsitektur sistem yang akan digunakan. Penulis juga menganalisis struktur data dan *flow* aplikasi berdasarkan hasil diskusi dengan mentor dan tim.

2. *Development*

Pada tahap ini, penulis mulai melakukan pembangunan aplikasi dari awal berdasarkan hasil riset sebelumnya. Aktivitas ini mencakup pembuatan sisi *frontend* dan *backend*, perancangan serta pengelolaan *database*, dan integrasi dengan *API*. Penulis menggunakan *Golang* untuk membangun *API* dan *Next.js* dengan *Material UI* untuk antarmuka pengguna.

3. *Testing*

Setelah proses pembangunan aplikasi selesai, penulis bersama mentor dan tim internal melakukan proses *testing* untuk memastikan aplikasi berjalan sesuai fungsinya. Pengujian dilakukan guna memastikan tidak terdapat *bug* kritis serta sistem dapat digunakan sesuai kebutuhan pengguna internal.



3.3 Uraian Pelaksanaan Magang

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

Minggu Ke-	Pekerjaan yang Dilakukan
1	Briefing bersama tim, mengenal alur kerja dan pembagian peran dalam proyek pembuatan sistem absensi terpusat untuk multi-perusahaan di Arraya Holding.
2	Melakukan riset dan studi pustaka terkait sistem HRIS dan absensi karyawan. Menyusun ERD (Entity Relationship Diagram) untuk kebutuhan database sistem.
3	Implementasi awal database dan penyusunan struktur API untuk fitur User dan Absensi.
4	Pembuatan halaman login, validasi autentikasi, serta dashboard awal untuk role karyawan.
5	Implementasi fitur check-in dan check-out pada halaman absensi, termasuk aturan keterlambatan dan ketidakhadiran.
6	Penyempurnaan form pengajuan izin (izin, sakit, cuti), validasi data, dan upload dokumen lampiran.
7	Pembuatan halaman riwayat absensi dan perizinan dengan fitur filter berdasarkan bulan dan tampilan detail.
8	Penambahan badge status absensi (hadir, izin, sakit, cuti) dan fitur ekspor data absensi ke format PDF dan Excel.
9	Pembuatan API dashboard absensi untuk data harian, mingguan, dan bulanan. Integrasi dengan halaman dashboard karyawan.
10	Perbaikan bug: timestamp absen tidak tampil, foto absensi tidak muncul di admin, dan validasi lokasi absen.
11	Perancangan awal sistem TAC (Tax Automation and Calculation) untuk pengelolaan aktivitas perpajakan perusahaan.
12	Pembuatan struktur database TAC, penetapan role user (admin, client, finance, tax officer), serta sistem autentikasi awal.
13	Pembuatan antarmuka frontend TAC menggunakan Next.js dan Tailwind CSS untuk halaman login dan dashboard admin.

3.3.1 Arraya Attendance

Pada bagian ini, penulis akan menjelaskan apa saja yang telah dikerjakan selama proses magang dalam pengembangan sistem *Arraya Attendance*. Sistem ini merupakan platform absensi berbasis web yang dirancang untuk menggantikan proses absensi manual yang sebelumnya dilakukan secara terpisah di masing-masing entitas perusahaan di bawah naungan Arraya Holding.

Hingga akhir masa magang, sistem telah berhasil dibangun dan digunakan untuk mencatat kehadiran karyawan, termasuk fitur *check-in*, *check-out*, serta pengajuan izin, sakit, dan cuti. Selain itu, sistem ini juga mendukung pengelolaan struktur organisasi perusahaan seperti cabang (*branch*), divisi (*department*), dan jabatan (*position*). Pada saat penyusunan laporan ini, sistem masih berada dalam tahap *bug testing* dan perbaikan antarmuka pengguna, khususnya pada bagian panel admin.

Secara umum, pengguna perlu masuk ke dalam sistem melalui proses *login*. Setelah berhasil masuk, pengguna dapat melakukan absensi harian, melihat riwayat kehadiran, atau mengajukan izin secara digital. Sistem ini dirancang agar mudah digunakan oleh seluruh karyawan dari berbagai cabang, serta memiliki hak akses berbeda berdasarkan peran pengguna.

Dalam sistem ini, terdapat tiga jenis peran (*role*) utama yang membedakan hak akses pengguna, yaitu **Employee**, **Human Capital (HC)**, dan **Admin**. Role *Employee* merupakan peran dasar yang digunakan oleh seluruh staf atau karyawan, dengan akses untuk melakukan *check-in*, *check-out*, melihat riwayat absensi pribadi, serta mengajukan izin secara digital.

Peran *Human Capital (HC)* diperuntukkan bagi tim HRD yang memiliki wewenang administratif, seperti mengelola data karyawan, melihat seluruh pengajuan izin dari berbagai cabang, serta memberikan *approval* akhir terhadap pengajuan izin karyawan.

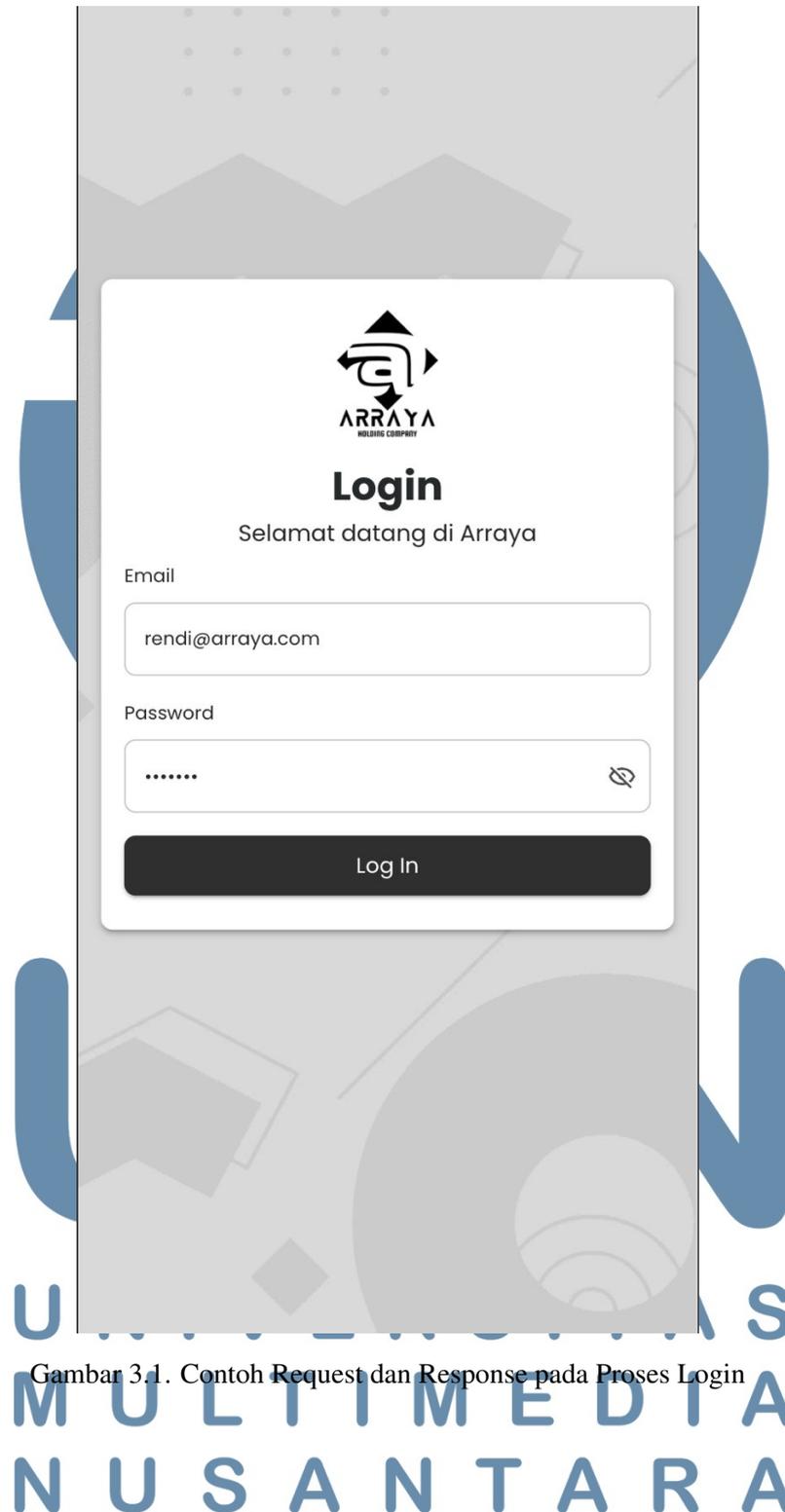
Adapun role *Admin* memiliki akses terhadap konfigurasi sistem secara teknis, termasuk pengelolaan pengguna, struktur organisasi, serta pengaturan aturan absensi. Meskipun demikian, admin tidak memiliki hak untuk menyetujui pengajuan izin; seluruh proses persetujuan hanya dapat dilakukan oleh pengguna dengan peran HC.

Dalam setiap proses absensi, sistem secara otomatis mencatat lokasi geografis pengguna dalam bentuk *latitude* dan *longitude* yang dikirim dari perangkat pengguna. Informasi ini disimpan ke dalam basis data dan digunakan

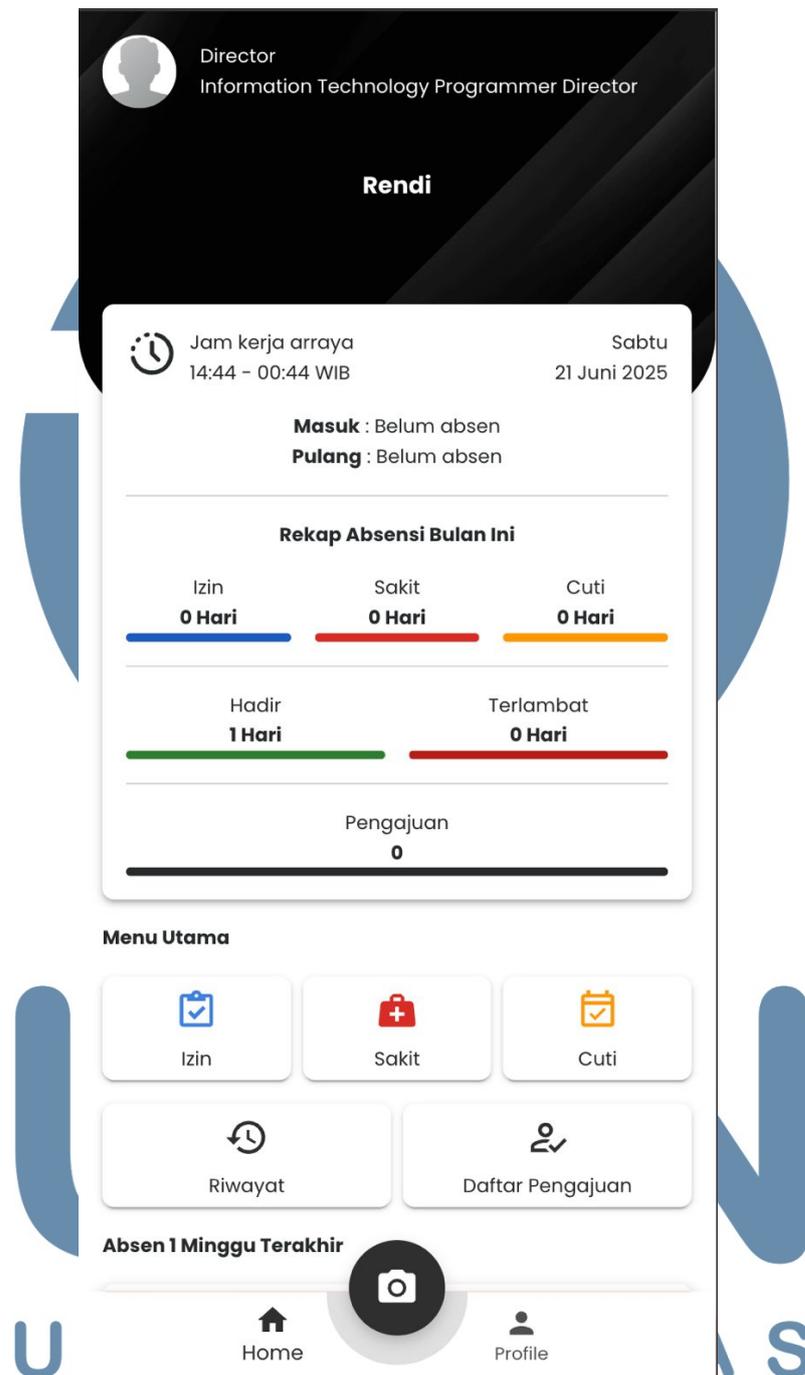
sebagai dasar validasi lokasi absensi.

Namun demikian, demi menjaga kesederhanaan antarmuka serta aspek keamanan data, informasi mentah berupa koordinat *latitude* dan *longitude* tidak ditampilkan kepada pengguna secara langsung. Pada antarmuka admin maupun HC, lokasi absensi ditampilkan dalam bentuk alamat lengkap hasil konversi (geocoding) dari koordinat tersebut. Akses terhadap data koordinat asli hanya tersedia melalui basis data dan bersifat internal, guna mencegah potensi penyalahgunaan atau kerancuan dalam interpretasi data lokasi.





Gambar 3.1. Contoh Request dan Response pada Proses Login



Gambar 3.2. Contoh Request dan Response pada Proses Login

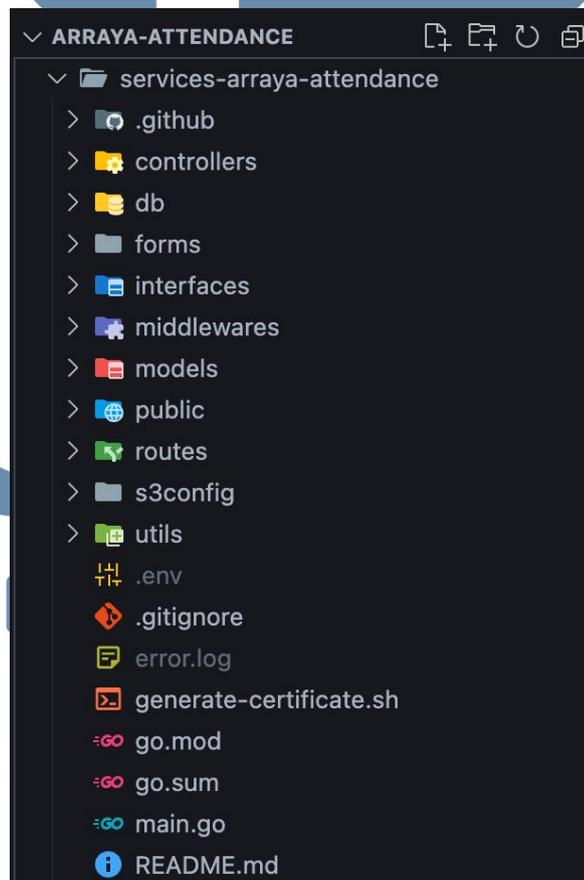
A. Infrastruktur Backend Aplikasi *Arraya Attendance*

Backend aplikasi *Arraya Attendance* dibangun menggunakan bahasa pemrograman Go (*Golang*) dengan framework Gin-Gonic, dan menggunakan PostgreSQL sebagai sistem manajemen basis data relasional. Pemilihan teknologi

ini didasarkan pada pertimbangan performa, skalabilitas, serta kemudahan dalam membangun layanan *RESTful API* yang stabil dan terstruktur.

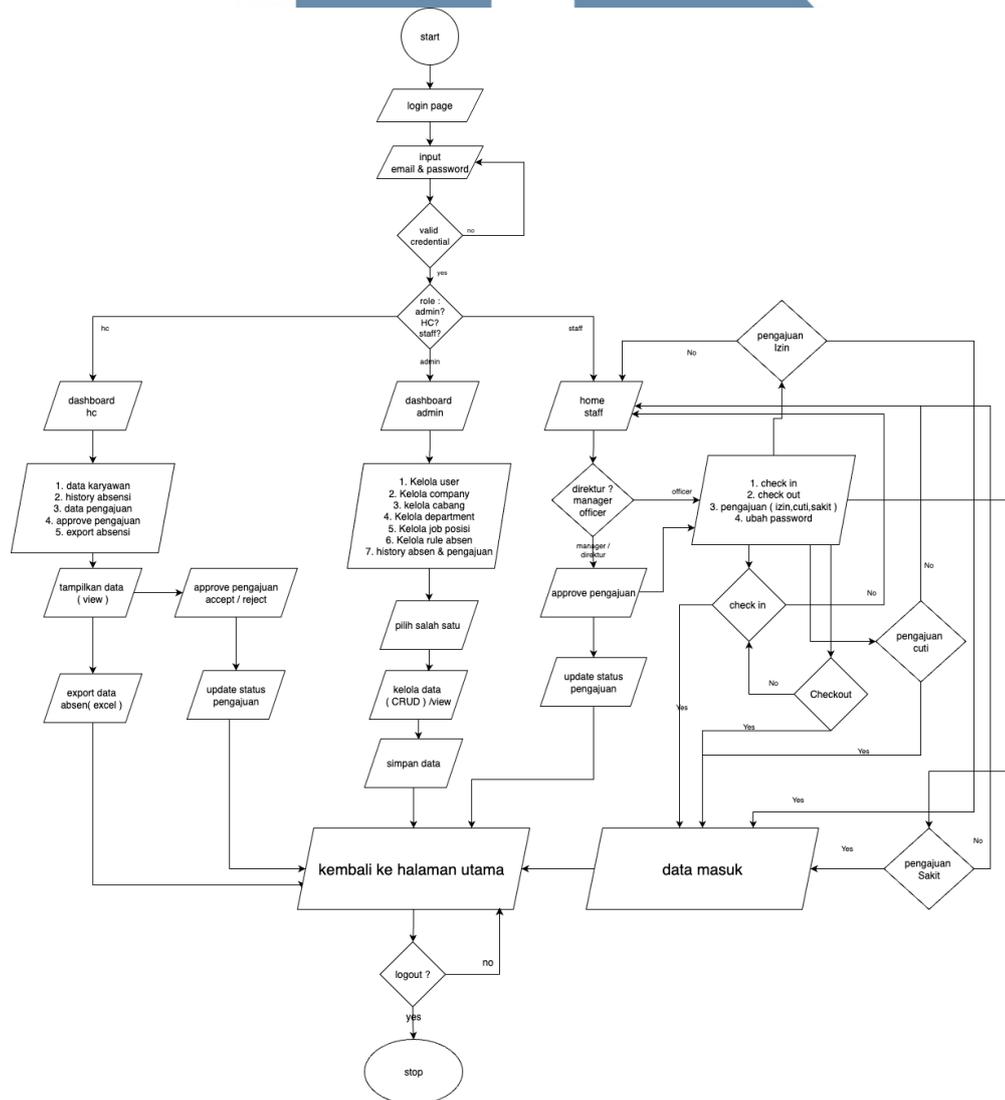
Struktur proyek backend menggunakan pendekatan *layered architecture*, yaitu pemisahan kode berdasarkan tanggung jawabnya. Dalam proyek ini, terdapat lapisan *routes* yang menangani endpoint API, *controllers* yang menjadi jembatan antara permintaan pengguna dan logika bisnis, serta *services* yang menangani proses seperti validasi, pemrosesan absensi, dan pengajuan izin. Untuk komunikasi dengan basis data, digunakan lapisan *models* dan *interfaces* yang membantu menjaga konsistensi data. Selain itu, sistem ini juga dilengkapi *middlewares* untuk autentikasi menggunakan JSON Web Token (JWT), serta folder *forms* untuk memvalidasi input data pengguna secara terpisah dan modular.

Arsitektur ini selaras dengan prinsip *separation of concerns* dalam pengembangan perangkat lunak, yang bertujuan untuk meningkatkan keterbacaan, skalabilitas, serta mempermudah proses debugging dan pemeliharaan sistem. Struktur direktori backend secara umum dapat dilihat pada Gambar 3.3.



Gambar 3.3. Struktur Arsitektur Backend Aplikasi *Arraya Attendance*

Selain struktur direktori backend, sistem ini juga dirancang mengikuti alur penggunaan berdasarkan peran pengguna. Flowchart pada Gambar 3.4 menunjukkan bagaimana pengguna berinteraksi dengan sistem berdasarkan role yang dimilikinya, baik sebagai staff, admin, maupun tim Human Capital.



Gambar 3.4. Flowchart Alur Penggunaan Sistem *Arraya Attendance*

Flowchart tersebut menggambarkan proses dimulai dari login, validasi role pengguna, hingga akses dashboard masing-masing role. Staff memiliki akses untuk melakukan check-in/check-out, mengajukan izin, cuti, dan sakit, serta mengubah password. Sementara itu, direktur dan manajer dapat melakukan verifikasi pengajuan dari bawahannya. Role admin memiliki akses penuh untuk pengelolaan entitas seperti user, cabang, departemen, dan aturan absensi. Tim Human Capital

(HC) memegang peran penting dalam persetujuan akhir serta ekspor data absensi. Seluruh data yang masuk dari proses tersebut akan dicatat dan disimpan ke dalam sistem untuk keperluan dokumentasi, rekap, dan pelaporan.

A.1 API Login & Autentikasi User

Untuk mengakses fitur sistem absensi *Arraya Attendance*, setiap pengguna harus terlebih dahulu melakukan proses *login*. Proses ini bertujuan untuk mengautentikasi identitas pengguna dan memberikan hak akses yang sesuai berdasarkan peran (*role*) dalam sistem, seperti admin, karyawan, atau kepala divisi.

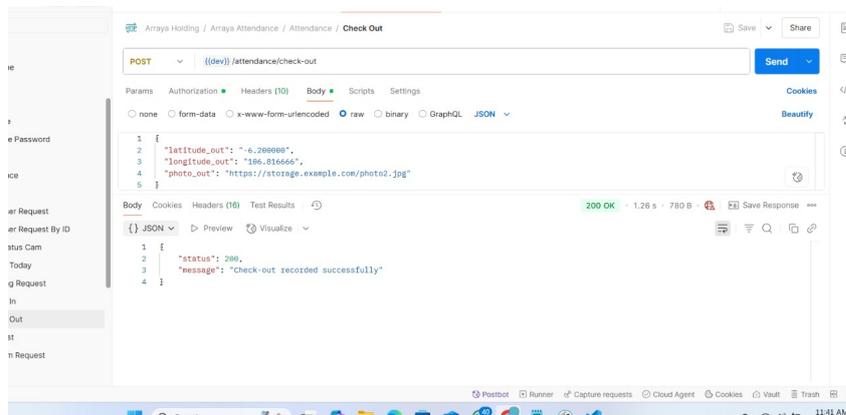
Pengguna melakukan login dengan mengirimkan data berupa email dan password melalui endpoint `POST /api/v1/auth/login`. Data ini kemudian diproses oleh handler yang terdapat dalam file `controllers/auth.go`. Validasi input dilakukan terlebih dahulu menggunakan struktur form `forms/auth.go` dan dieksekusi oleh fungsi validasi dari `forms/validator.go`.

Jika data yang diberikan valid, sistem akan mencocokkannya dengan data yang tersimpan di basis data menggunakan model dari `models/auth.go`. Setelah validasi berhasil, sistem akan membuat *JSON Web Token (JWT)* sebagai tanda autentikasi, yang berisi informasi penting seperti ID pengguna, peran (*role*), dan masa berlaku token. Pembuatan serta pengelolaan token ini dilakukan melalui `middlewares/token.go`.

Token JWT yang telah dibuat akan dikembalikan kepada pengguna sebagai bagian dari respon login yang sukses. Token ini wajib disertakan dalam setiap permintaan berikutnya melalui `Authorization Header` agar dapat mengakses API yang bersifat privat.

Setiap endpoint yang dilindungi akan melakukan verifikasi token melalui `middleware` yang berada di `middlewares/token.go`. Apabila token yang diberikan tidak valid atau telah kedaluwarsa, maka permintaan akan ditolak dan sistem akan mengembalikan respon `401 Unauthorized`.

Dengan penerapan autentikasi berbasis token ini, sistem dapat memastikan bahwa setiap akses hanya dilakukan oleh pengguna yang sah dan memiliki hak akses sesuai perannya. Dokumentasi endpoint, contoh struktur request, serta respon login dapat dilihat pada Gambar 3.5.



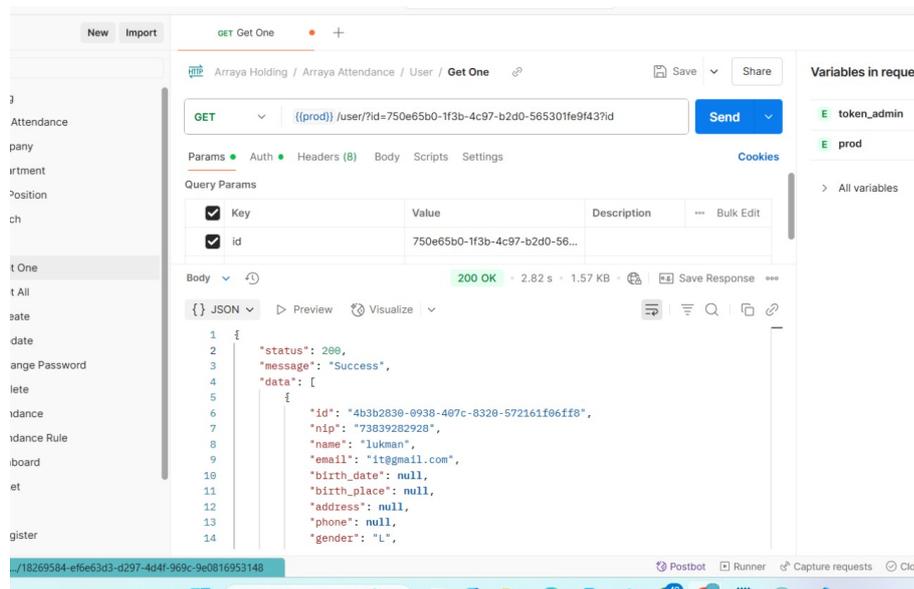
Gambar 3.5. Contoh Request dan Response pada Proses Login

A.2 API User (Get by ID)

Setelah pengguna berhasil melakukan proses login dan menerima token autentikasi, pengguna dapat melihat informasi profilnya melalui endpoint `GET /api/v1/user/profile`. Endpoint ini dirancang untuk menampilkan data pengguna yang sedang login, tanpa memerlukan parameter `user_id` secara eksplisit. Sistem akan mengekstrak informasi identitas pengguna dari token JWT yang dikirimkan melalui header permintaan.

Token tersebut dikirim dengan format `Authorization: Bearer <access_token>` pada tab Header. Validasi token dilakukan pada middleware `middlewares/token.go` untuk memastikan hanya pengguna dengan autentikasi sah yang dapat mengakses data pribadi mereka. Setelah token dinyatakan valid, data pengguna diambil dari database melalui pemanggilan model `models/user.go`, dan hasilnya dikembalikan dalam format JSON. Endpoint ini digunakan pada dashboard aplikasi untuk menampilkan detail informasi karyawan yang sedang login, seperti nama lengkap, jabatan, email, dan informasi lainnya.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



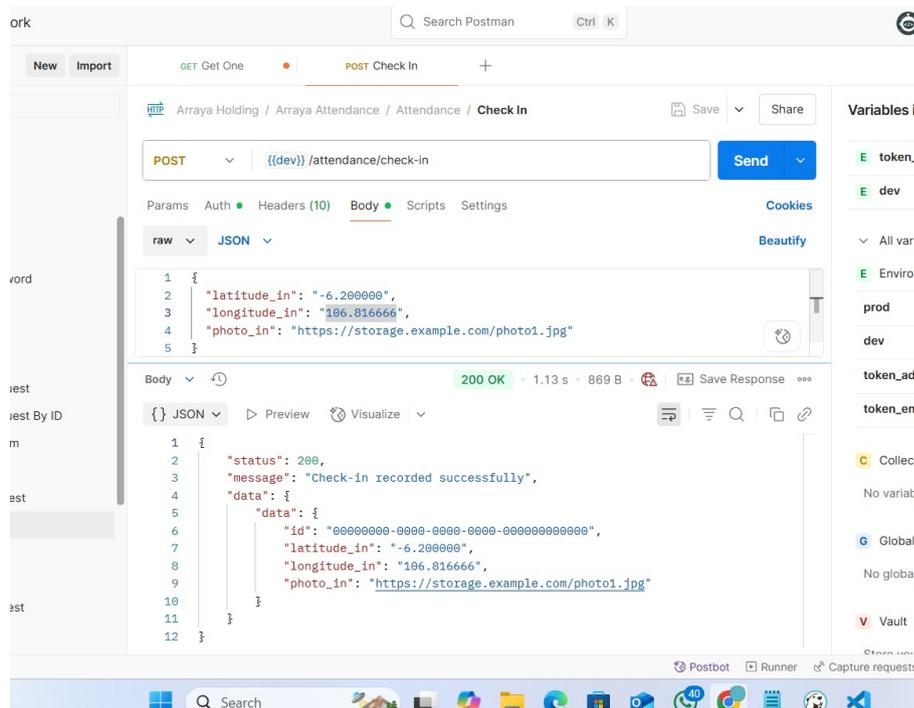
Gambar 3.6. Contoh Permintaan API untuk Menampilkan Data Profil Pengguna (GET /user/profile)

A.3 API Absensi (Check-In)

Untuk mencatat kehadiran, pengguna melakukan proses *check-in* dengan mengirimkan permintaan POST ke endpoint `/api/v1/attendance/check-in`. Proses ini bertujuan untuk mencatat waktu masuk kerja beserta lokasi geografis dan dokumentasi berupa foto. Data yang dikirimkan mencakup tiga informasi utama, yaitu `latitude_in`, `longitude_in`, dan `photo_in` (URL foto).

Setiap permintaan harus disertai dengan token autentikasi pada header sebagai `Authorization: Bearer <access_token>`. Token tersebut akan divalidasi terlebih dahulu melalui `middlewares/token.go`, dan sistem akan memastikan bahwa pengguna memiliki hak akses untuk melakukan *check-in*. Selanjutnya, validasi input dilakukan di `forms/attendance.go` menggunakan `validator.go`.

Jika data yang dikirim valid dan pengguna memenuhi aturan absensi (seperti jarak lokasi dan jam kerja), maka sistem akan menyimpan data kehadiran ke database melalui model `models/attendance.go`. Logika utama proses ini berada pada `controllers/attendance.go`. Endpoint ini merupakan bagian penting dalam mencatat kehadiran harian karyawan.



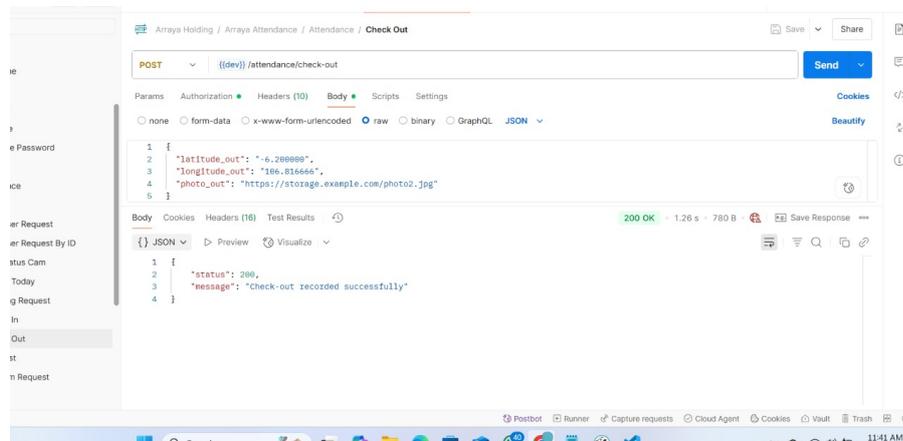
Gambar 3.7. Contoh Permintaan API untuk Proses Check-In (POST /attendance/check-in)

A.4 API Absensi (Check-Out)

Proses *check-out* berfungsi untuk mencatat jam pulang kerja pengguna. Permintaan dilakukan melalui endpoint POST `/api/v1/attendance/check-out`. Sama seperti proses check-in, pengguna harus mengirimkan data lokasi berupa `latitude_out`, `longitude_out`, serta foto sebagai `photo_out`.

Pengguna juga wajib menyertakan token JWT pada header dengan format `Authorization: Bearer <access_token>` untuk proses autentikasi. Sistem kemudian akan memvalidasi token tersebut dan memastikan bahwa pengguna sebelumnya telah melakukan proses check-in sebelum dapat melanjutkan ke check-out. Validasi dan logika bisnis untuk proses ini diatur dalam file `forms/attendance.go`, `middlewares/token.go`, dan `controllers/attendance.go`.

Setelah data diverifikasi dan dinyatakan valid, sistem akan mencatat jam pulang beserta lokasi dan dokumentasi ke database melalui `models/attendance.go`. Informasi check-out ini akan digunakan untuk perhitungan jam kerja dan pelaporan absensi karyawan pada sistem admin.



Gambar 3.8. Contoh Permintaan API untuk Proses Check-Out (POST /attendance/check-out)

A.5 API Riwayat Absensi (Attendance History)

API ini digunakan untuk menampilkan daftar riwayat kehadiran seorang pengguna dalam rentang waktu tertentu. Fitur ini memungkinkan pengguna untuk memantau status kehadiran mereka, seperti kehadiran, izin, sakit, cuti, dan keterlambatan, berdasarkan tanggal.

Permintaan dikirimkan menggunakan metode GET ke endpoint:

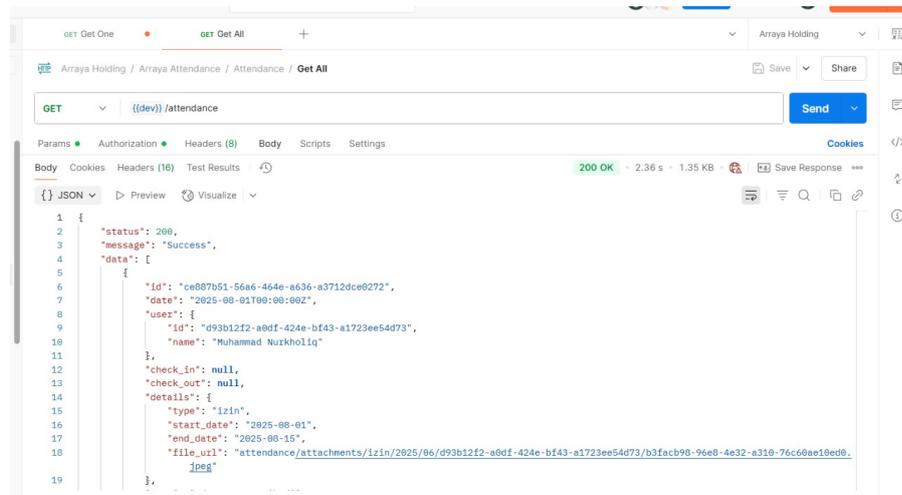
`/api/v1/attendance/history`

Endpoint ini hanya dapat diakses oleh pengguna yang telah login dan memiliki token autentikasi yang valid, yang dikirimkan melalui header `Authorization: Bearer <access_token>`. Sistem akan mengekstrak informasi `user_id` dari token dan menampilkan semua riwayat absensi yang berkaitan dengan pengguna tersebut.

Secara teknis, permintaan ini diproses melalui handler yang terdapat di `controllers/attendance.go`, dan data absensi diambil dari database menggunakan fungsi-fungsi yang didefinisikan dalam `models/attendance.go`. Data yang ditampilkan mencakup tanggal absensi, status, jam check-in dan check-out, keterangan lokasi, serta apakah pengguna terlambat atau tidak.

API ini digunakan dalam antarmuka frontend, khususnya pada dashboard pengguna dan halaman riwayat absensi. Selain itu, data ini juga dapat difilter berdasarkan bulan tertentu atau periode waktu tertentu sesuai kebutuhan pengguna.

Contoh pengujian endpoint ini ditampilkan pada Gambar 3.9.



Gambar 3.9. Contoh Pengujian Endpoint Riwayat Absensi

A.6 API Pengajuan Izin, Sakit, dan Cuti

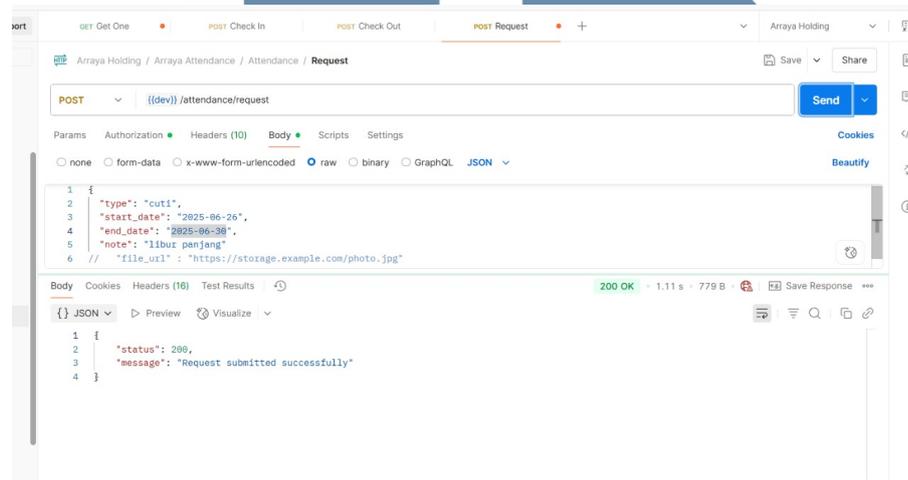
API ini digunakan oleh pengguna untuk mengajukan permohonan izin, baik karena keperluan pribadi (izin), alasan kesehatan (sakit), maupun cuti resmi. Pengajuan dilakukan secara digital melalui endpoint `POST /api/v1/permission` dan hanya dapat diakses oleh pengguna yang telah login. Setiap permintaan dikirimkan dengan menyertakan token autentikasi pada header menggunakan format `Authorization: Bearer <access_token>`.

Data yang dikirimkan pada body permintaan mencakup jenis izin (`status`), tanggal mulai dan berakhirnya izin (`start_date`, `end_date`), keterangan tambahan (`notes`), serta lampiran opsional seperti file bukti pendukung. Setelah permintaan diterima, sistem akan melakukan validasi melalui form yang terdapat di `forms/permission.go`, kemudian diproses oleh controller dan disimpan ke dalam database menggunakan model `models/permission.go`.

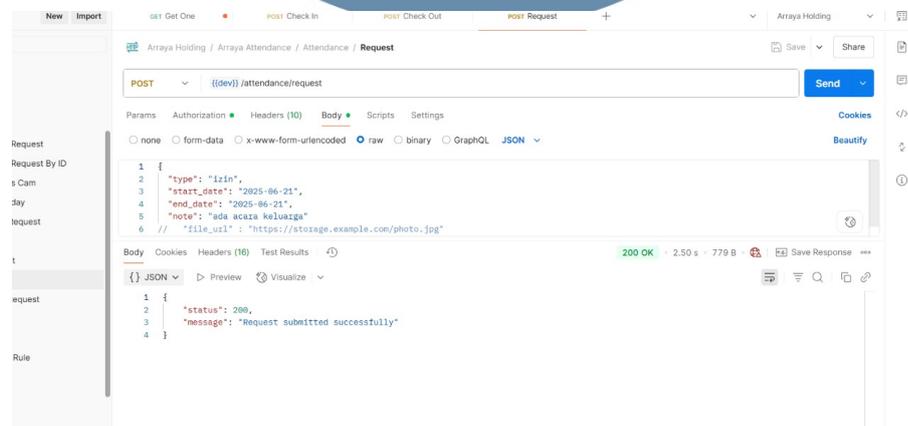
Proses pengajuan ini mengikuti alur persetujuan bertingkat. Untuk pengguna dengan jabatan staf atau karyawan biasa, pengajuan izin harus terlebih dahulu disetujui oleh atasan langsung seperti manajer atau direktur. Selama belum mendapatkan persetujuan dari atasan, permintaan izin tidak akan muncul di dashboard Human Capital (HC). Jika pengajuan telah disetujui oleh atasan, barulah HC akan menerima permintaan tersebut untuk tahap persetujuan akhir. Sementara itu, jika pengaju merupakan pengguna dengan peran manajer atau direktur, maka sistem secara otomatis akan mengirimkan pengajuan langsung ke HC tanpa perlu validasi dari atasan lainnya. Dengan alur seperti ini, sistem memastikan bahwa

proses pengajuan izin dilakukan secara hierarkis dan sesuai struktur organisasi perusahaan.

Contoh pengujian endpoint ini menggunakan Postman ditampilkan pada Gambar ??.

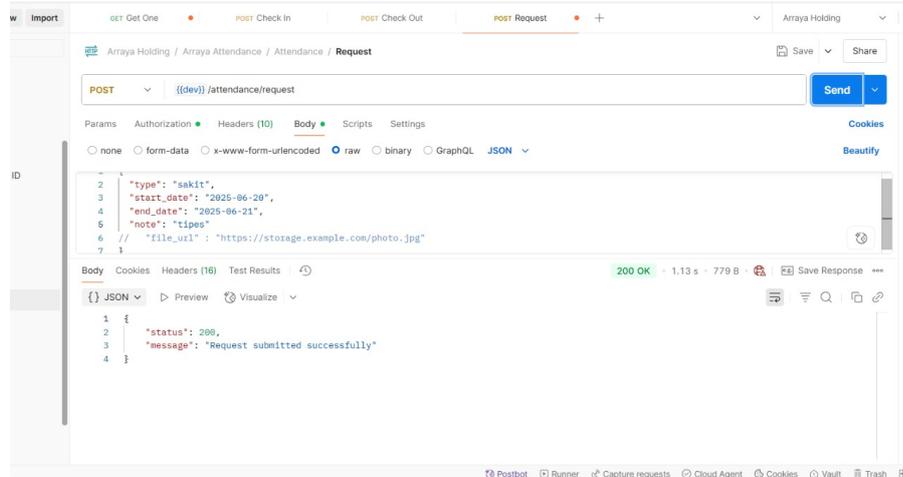


Gambar 3.10. Contoh Pengajuan Cuti melalui Endpoint POST /permission



Gambar 3.11. Contoh Pengajuan Izin melalui Endpoint POST /permission

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.12. Contoh Pengajuan Sakit melalui Endpoint POST /permission

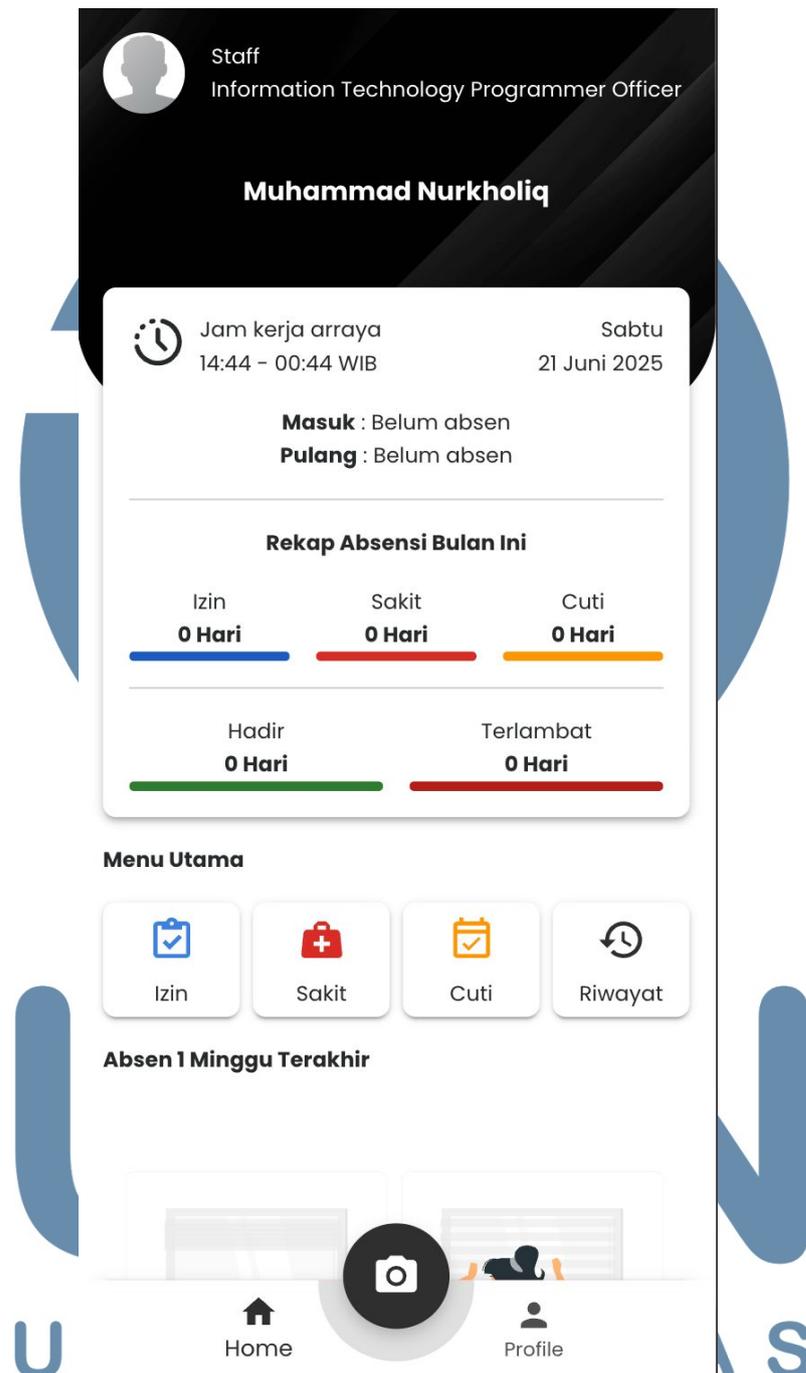
B. Antarmuka Pengguna - Staff

Pengguna dengan peran sebagai *staff* memiliki akses terhadap fitur dasar yang berkaitan dengan kehadiran dan pengajuan izin. Tampilan antarmuka dibuat sederhana dan mudah digunakan, sehingga memudahkan staf untuk melakukan aktivitas harian seperti absensi dan pengecekan riwayat. Seluruh fitur ini dikembangkan menggunakan framework `Next.js` dan pustaka antarmuka `Material UI`, serta terintegrasi dengan backend menggunakan `SWR` untuk pengambilan data secara real-time.

B.1 Dashboard Staff

Halaman dashboard merupakan tampilan awal yang ditampilkan kepada user setelah berhasil login. Pada halaman ini, staff dapat langsung melihat status kehadiran hari ini — apakah sudah melakukan *check-in* atau belum. Di bawahnya terdapat ringkasan statistik absensi bulanan yang mencakup status Hadir, Telat, Izin, Sakit, dan Cuti.

Dashboard juga dilengkapi dengan tombol `Check-In` dan `Check-Out` yang hanya aktif sesuai aturan waktu. Akses fitur ini dibatasi berdasarkan role yang sedang login. Data disesuaikan dengan akun pengguna aktif dan hanya menampilkan informasi milik karyawan tersebut.



Gambar 3.13. Tampilan Dashboard Staff

B.2 Pengajuan Izin

Halaman pengajuan izin memungkinkan staff untuk mengajukan cuti, sakit, atau izin lainnya secara digital. Pada form pengajuan, user harus mengisi beberapa kolom: jenis izin, tanggal mulai dan selesai, alasan, serta mengunggah file lampiran

jika dibutuhkan (misalnya surat keterangan sakit). Setelah pengajuan dikirim, sistem akan meneruskannya secara otomatis ke atasan langsung (manager atau direktur) untuk disetujui terlebih dahulu. Pengajuan baru dapat dilihat oleh Human Capital (HC) apabila telah disetujui oleh atasan. Proses ini mencerminkan hierarki organisasi dan mencegah permintaan yang belum diverifikasi sampai ke pusat.

The image shows a mobile application interface for staff leave requests. The main form is titled "Izin" and includes the following elements:

- Tanggal Mulai** (Start Date): 01 Jun 2025
- Tanggal Akhir** (End Date): 30 Jun 2025
- Terapkan Filter** (Apply Filter) button
- Tambah Pengajuan** (Add Request) button

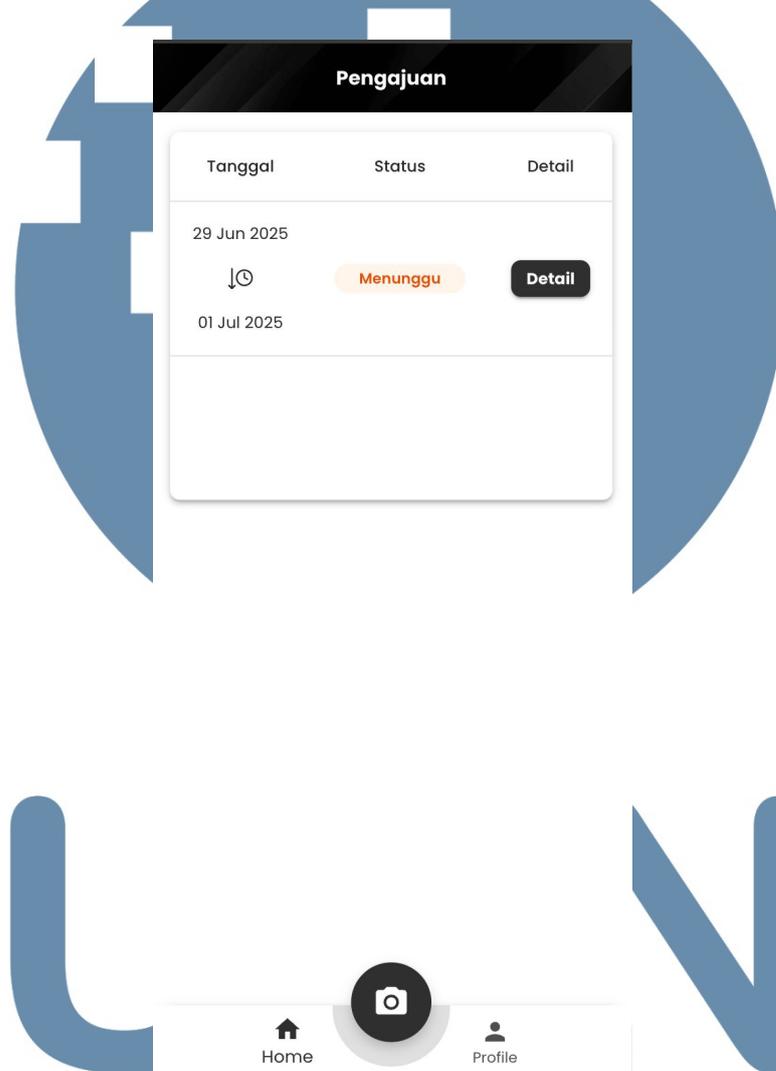
The background of the form features an illustration of a woman standing next to a computer monitor that displays a "NO DATA" error message. At the bottom of the screen, there is a navigation bar with three icons: a home icon, a camera icon, and a profile icon.

Gambar 3.14. Form Pengajuan Izin Staff

B.3 Riwayat Pengajuan Izin

Fitur ini menampilkan daftar semua pengajuan izin yang telah dilakukan oleh staff. Tabel yang ditampilkan mencakup: jenis izin, tanggal, alasan singkat, serta status (Menunggu, Disetujui, atau Ditolak). Dengan fitur ini, pengguna dapat melakukan pengecekan atas proses pengajuan yang sedang berlangsung atau yang

sudah diproses. Selain itu, pengguna juga bisa membuka tampilan detail untuk melihat isi lengkap pengajuan termasuk lampiran bukti (jika ada), dan respon dari atasan atau HC. Fitur ini juga membantu pengguna dalam proses klarifikasi jika terjadi penolakan.

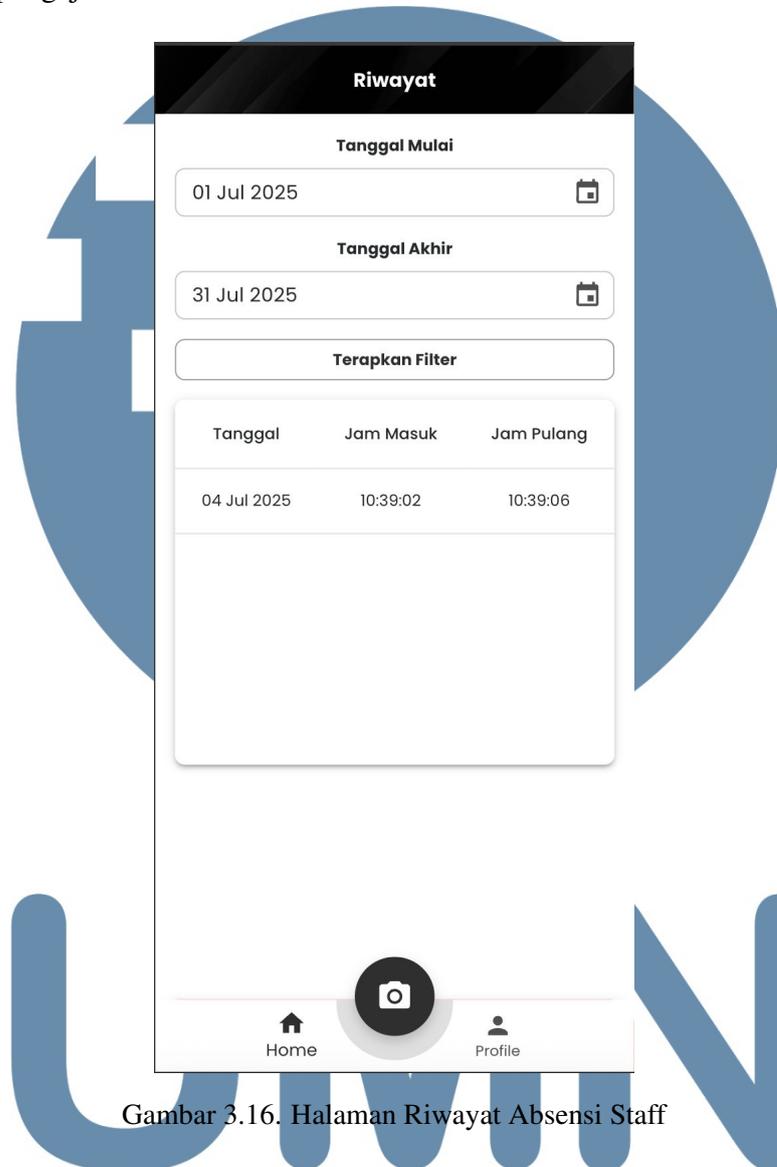


Gambar 3.15. Riwayat Pengajuan Izin Staff

B.4 Riwayat Absensi

Halaman ini memberikan akses bagi karyawan untuk melihat data kehadirannya selama satu bulan. Tampilan berupa tabel dengan entri harian, yang menampilkan tanggal, status kehadiran, jam masuk, jam pulang, serta badge keterangan seperti “Terlambat” atau “Tidak Hadir”. Pengguna dapat memilih bulan tertentu untuk melihat riwayatnya dan melakukan pengecekan terhadap keabsahan

absensi. Fitur ini sangat penting untuk keperluan internal karyawan seperti bukti kerja atau pengajuan administratif lain.



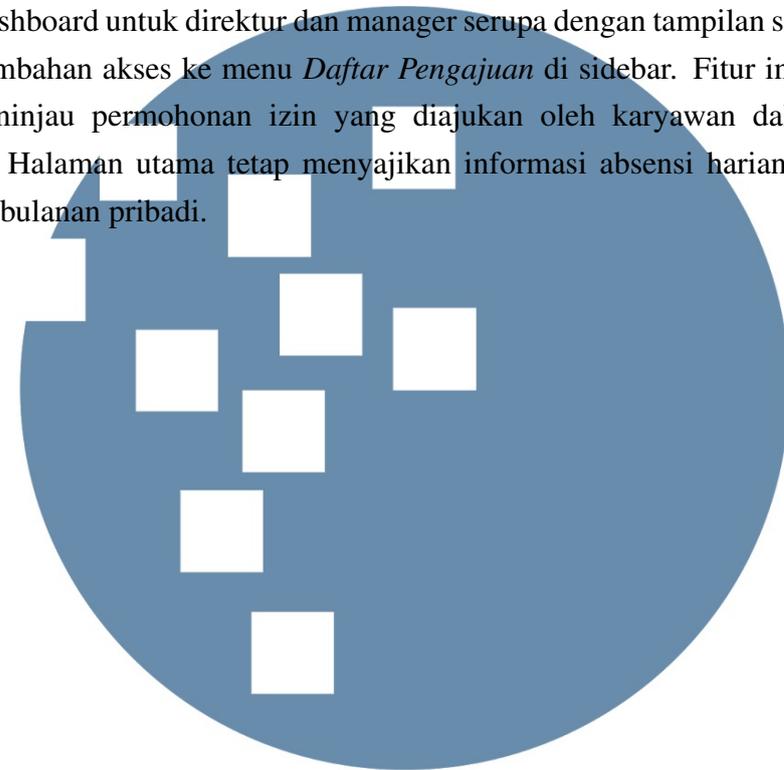
Gambar 3.16. Halaman Riwayat Absensi Staff

C. Antarmuka Pengguna – Direktur / Manager

Pengguna dengan peran sebagai *direktur* atau *manager* dalam sistem *Arraya Attendance* memiliki hak akses yang lebih tinggi dibandingkan staff biasa. Selain dapat melakukan check-in, check-out, serta pengajuan izin pribadi, mereka juga memiliki tanggung jawab untuk memverifikasi pengajuan izin dari bawahan secara langsung. Oleh karena itu, antarmuka untuk direktur dan manager dilengkapi dengan fitur tambahan seperti daftar pengajuan dari bawahan dan halaman persetujuan izin.

C.1 Dashboard Direktur / Manager

Dashboard untuk direktur dan manager serupa dengan tampilan staff, namun dengan tambahan akses ke menu *Daftar Pengajuan* di sidebar. Fitur ini berfungsi untuk meninjau permohonan izin yang diajukan oleh karyawan dalam satuan kerjanya. Halaman utama tetap menyajikan informasi absensi harian dan rekap kehadiran bulanan pribadi.



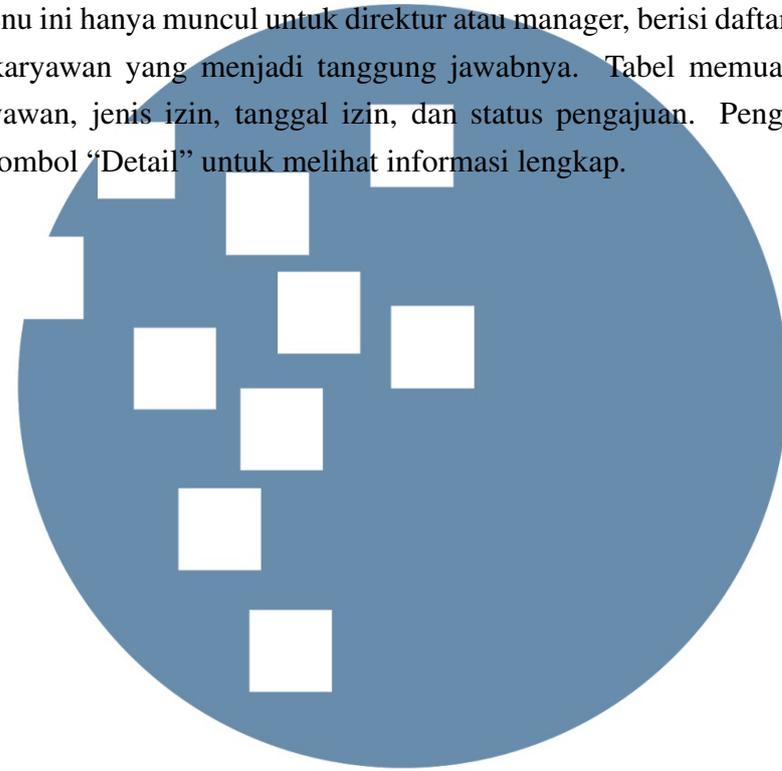
UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.17. Dashboard Direktur / Manager dengan Akses Pengajuan

C.2 Daftar Pengajuan Bawahan

Menu ini hanya muncul untuk direktur atau manager, berisi daftar pengajuan izin dari karyawan yang menjadi tanggung jawabnya. Tabel memuat informasi nama karyawan, jenis izin, tanggal izin, dan status pengajuan. Pengguna dapat menekan tombol “Detail” untuk melihat informasi lengkap.



UMMN

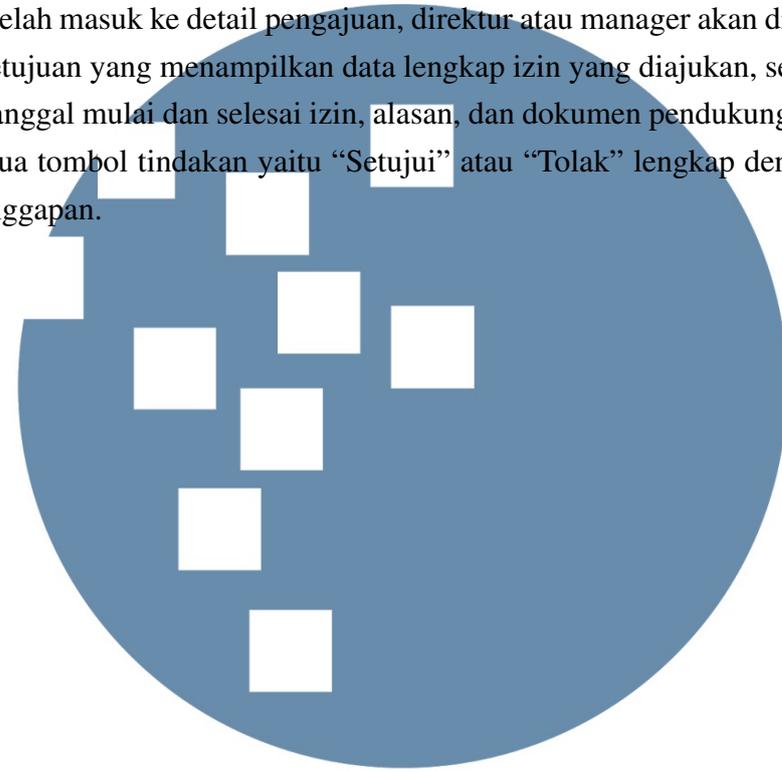
UNIVERSITAS
MULTIMEDIA
NUSANTARA



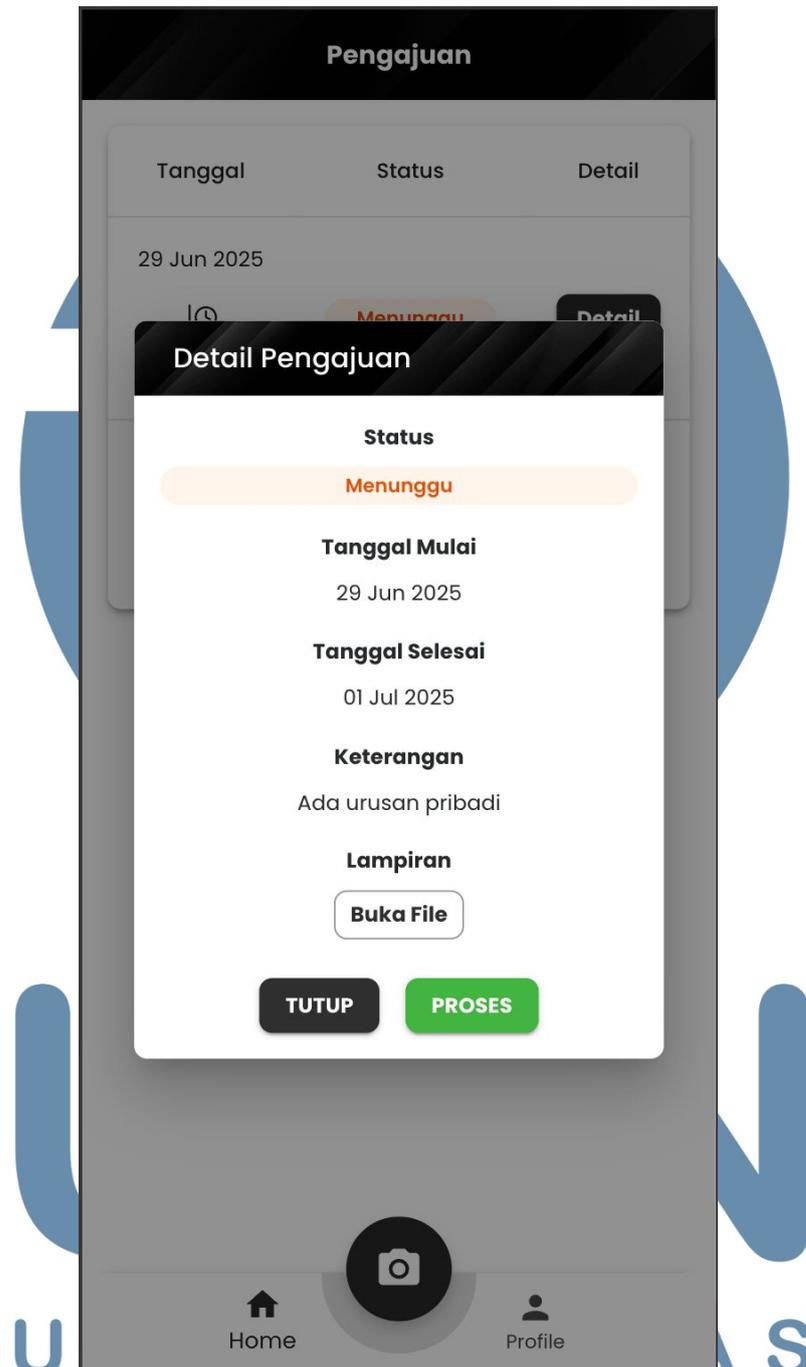
Gambar 3.18. Halaman Daftar Pengajuan Bawahan

C.3 Form Persetujuan Izin

Setelah masuk ke detail pengajuan, direktur atau manager akan diarahkan ke form persetujuan yang menampilkan data lengkap izin yang diajukan, seperti nama pengaju, tanggal mulai dan selesai izin, alasan, dan dokumen pendukung (jika ada). Tersedia dua tombol tindakan yaitu “Setujui” atau “Tolak” lengkap dengan kolom catatan tanggapan.



UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.19. Form Detail Persetujuan Pengajuan Izin

U
MULTIMEDIA
NUSANTARA



Gambar 3.20. Tampilan Form Approval oleh Direktur atau Manager

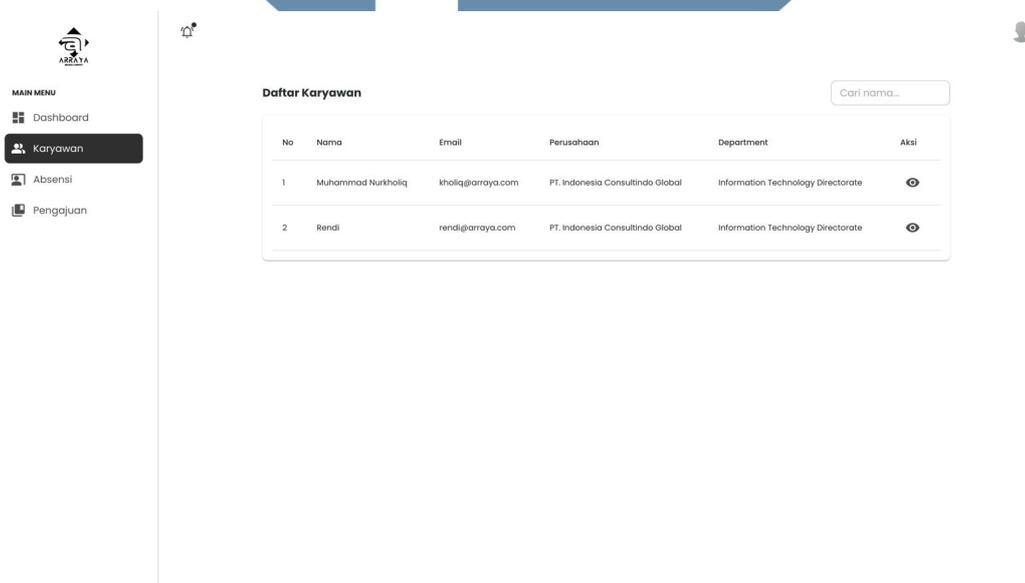
Proses persetujuan ini penting karena pengajuan dari karyawan biasa tidak akan diteruskan ke tim Human Capital (HC) tanpa terlebih dahulu disetujui oleh atasan. Namun, jika pengajuan dilakukan oleh direktur atau manager sendiri, maka permintaan tersebut akan langsung dikirim ke HC tanpa perlu persetujuan lanjutan.

D. Antarmuka Pengguna – Human Capital (HC)

Pengguna dengan peran *Human Capital (HC)* memiliki wewenang administratif tertinggi dalam sistem *Arraya Attendance*. Peran ini melibatkan pengelolaan seluruh data karyawan, pengawasan absensi, serta validasi akhir terhadap pengajuan izin, sakit, dan cuti. Oleh karena itu, antarmuka pengguna untuk HC dilengkapi dengan berbagai fitur manajemen dan akses penuh terhadap data operasional.

D.1 Halaman Data Karyawan

HC dapat melihat daftar seluruh karyawan dalam organisasi yang ditampilkan dalam bentuk tabel. Informasi yang disajikan meliputi nama, email, posisi, cabang, dan status akun. Tersedia tombol “Detail” pada setiap entri yang memungkinkan HC untuk meninjau profil lengkap dan riwayat kehadiran karyawan tersebut.

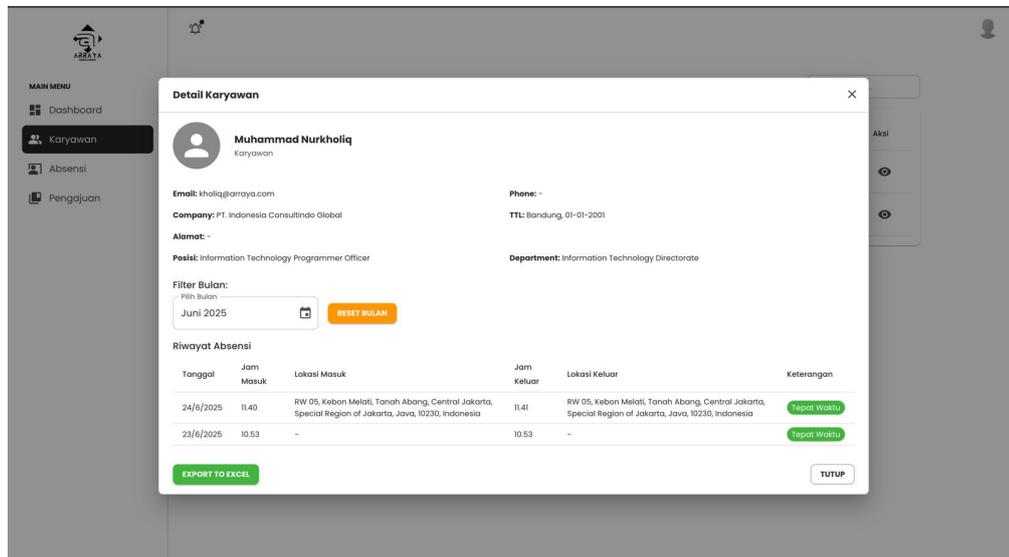


Gambar 3.21. Halaman Daftar Karyawan untuk Human Capital

D.2 Halaman Detail Karyawan

Setelah memilih salah satu karyawan dari daftar, HC dapat melihat detail profil yang mencakup informasi pribadi, jabatan, divisi, dan cabang. Di bagian bawah halaman, terdapat riwayat absensi lengkap dan semua pengajuan izin yang

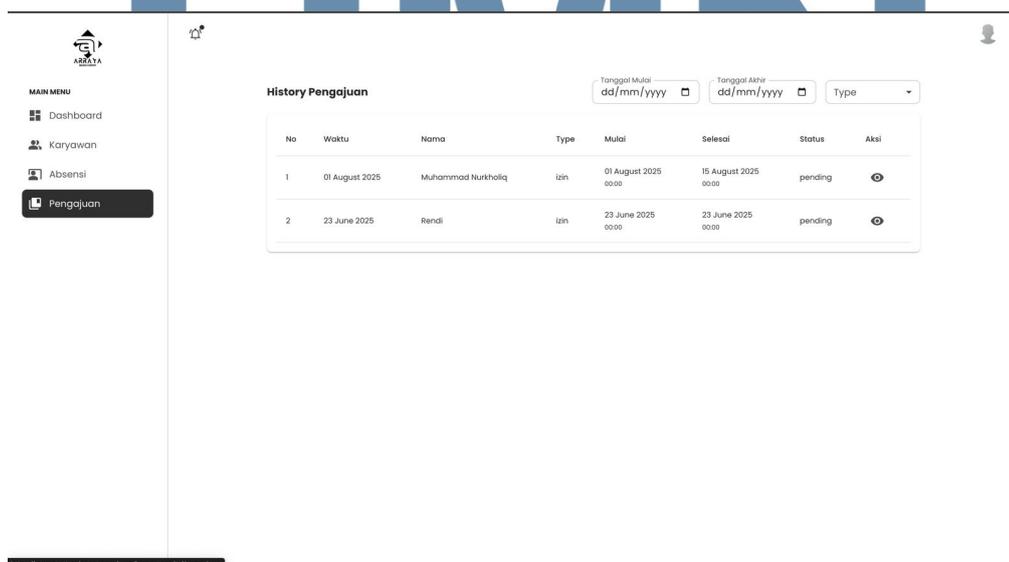
pernah dilakukan oleh karyawan tersebut.



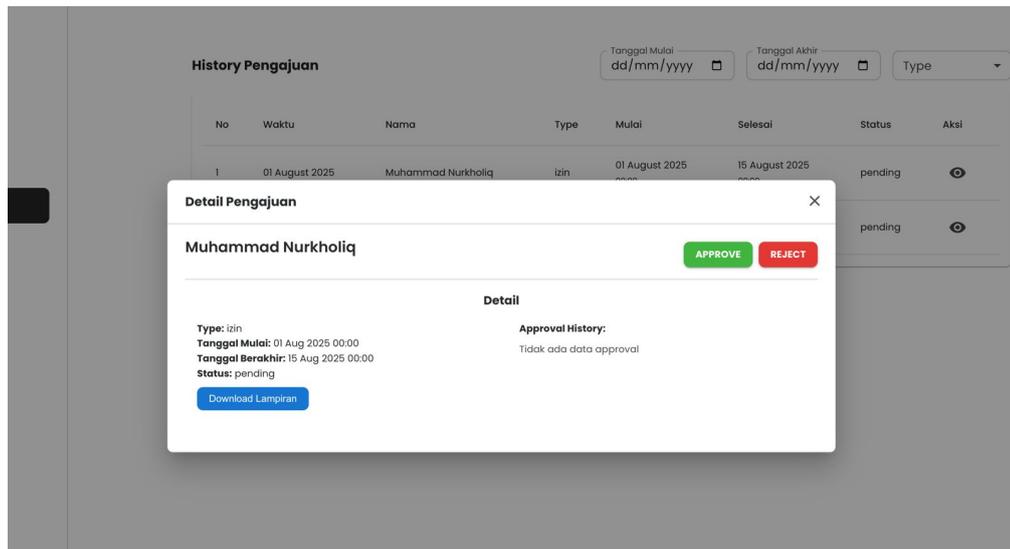
Gambar 3.22. Tampilan Detail Karyawan dan Riwayat Kehadiran

D.3 Halaman Riwayat Pengajuan

HC juga memiliki akses ke seluruh riwayat pengajuan izin dari semua karyawan yang telah disetujui oleh atasan. Di halaman ini, pengajuan ditampilkan lengkap dengan status, jenis izin, tanggal, dan lampiran. HC dapat menyetujui atau menolak pengajuan jika belum diproses.



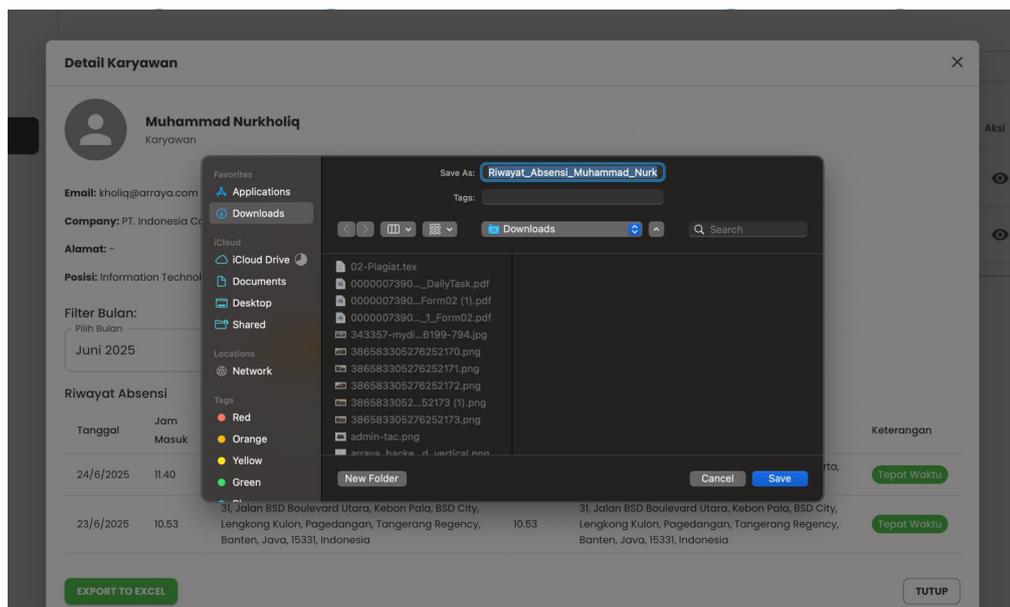
Gambar 3.23. Halaman Riwayat Pengajuan Izin Seluruh Karyawan



Gambar 3.24. Tampilan Detail Pengajuan untuk Diproses oleh HC

D.4 Fitur Ekspor dan Filter

Pada halaman data karyawan maupun absensi, HC diberikan fitur tambahan berupa ekspor data ke format Excel dan filter berdasarkan bulan atau status. Fitur ini memudahkan dalam menyusun laporan bulanan atau kebutuhan administratif lainnya.



Gambar 3.25. Fitur Filter dan Ekspor Data Absensi

	A	B	C	D	E	F	G
1	Tanggal	Status	Jam Masuk	Lokasi Masuk	Jam Keluar	Lokasi Keluar	Keterangan
2	24/6/2025	Tepat Waktu	11.40	RW 05, Kebo	11.41	RW 05, Kebo	Tepat Waktu
3	23/6/2025	Tepat Waktu	10.53	31, Jalan BSD	10.53	31, Jalan BSD	Tepat Waktu
4							
5							

Gambar 3.26. Fitur Filter dan Ekspor Data Absensi

Dengan akses penuh terhadap data operasional, peran HC dalam sistem ini sangat krusial dalam memastikan kelancaran administrasi kehadiran dan validitas izin karyawan.

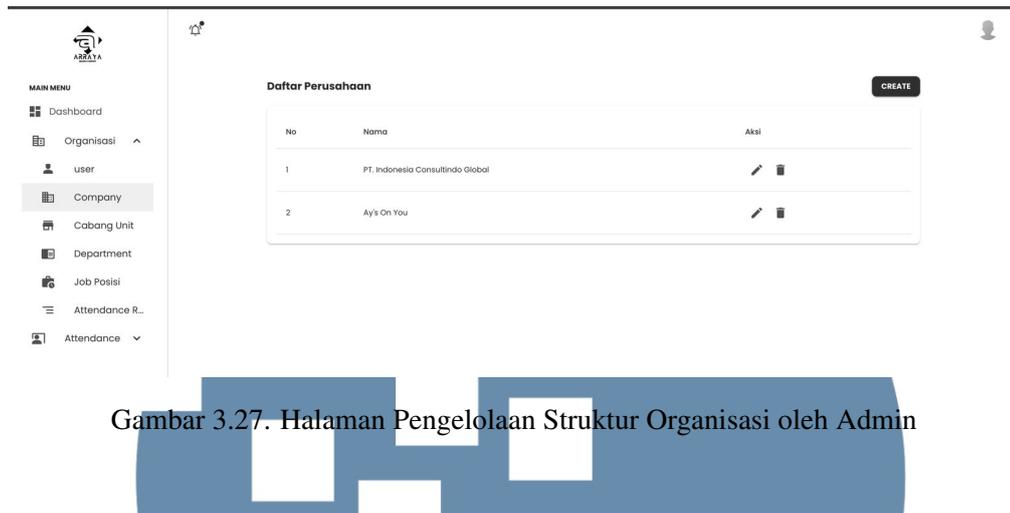
E. Antarmuka Pengguna – Admin

Role *admin* dalam aplikasi *Arraya Attendance* memiliki akses sistem secara menyeluruh dan berperan dalam konfigurasi teknis, struktur organisasi, serta monitoring fitur-fitur utama. Meskipun tidak terlibat langsung dalam pengelolaan absensi harian seperti peran HC, role ini bertanggung jawab terhadap pengaturan awal sistem dan manajemen struktur organisasi digital.

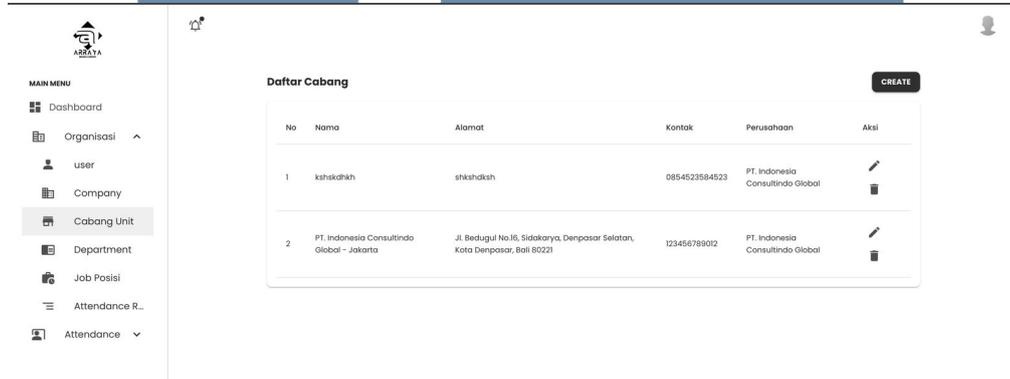
E.1 Halaman Pengelolaan Cabang, Departemen, dan Posisi

Admin memiliki akses penuh terhadap halaman konfigurasi struktur organisasi, yaitu pengelolaan *branch* (cabang), *department* (departemen), dan *position* (jabatan). Fitur ini penting untuk memastikan setiap karyawan terhubung dengan struktur organisasi yang tepat.

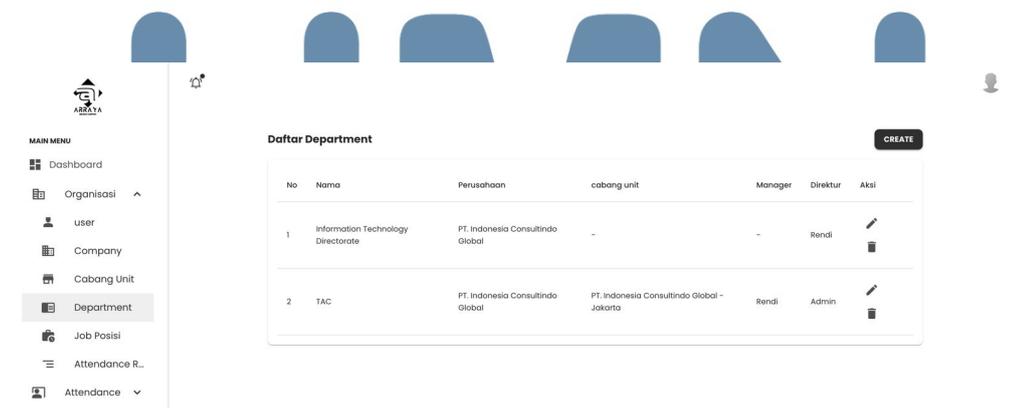
Setiap entitas dapat ditambahkan, diperbarui, atau dihapus melalui form yang disediakan. Data yang ditambahkan akan terhubung ke sistem pengguna untuk menentukan afiliasi dan hierarki karyawan.



Gambar 3.27. Halaman Pengelolaan Struktur Organisasi oleh Admin

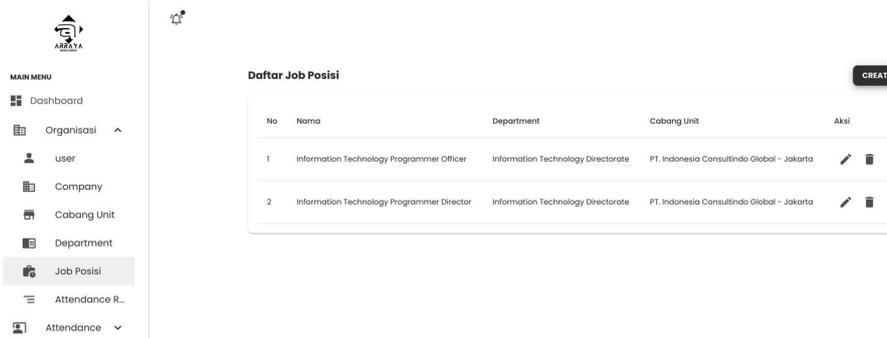


Gambar 3.28. Halaman Pengelolaan Struktur Organisasi oleh Admin



Gambar 3.29. Halaman Pengelolaan Struktur Organisasi oleh Admin

MULTIMEDIA
NUSANTARA



Gambar 3.30. Halaman Pengelolaan Struktur Organisasi oleh Admin

E.2 Pengelolaan Role dan Akses

Pada tahap awal pengembangan sistem, admin berperan dalam menetapkan dan mengelola peran (role) pengguna, termasuk penetapan role karyawan baru seperti staff, manager, direktur, maupun HC. Setiap peran memiliki hak akses berbeda, sehingga penetapan ini memengaruhi fungsionalitas UI yang ditampilkan kepada user.

E.3 Pemantauan Akses dan Uji Fitur

Admin juga dapat melakukan uji coba terhadap akses dan fitur melalui akun-akun simulasi, serta memastikan bahwa sistem dapat berfungsi sesuai spesifikasi. Aktivitas ini biasanya dilakukan saat proses bug testing sebelum sistem dioperasikan penuh.

Catatan: Hingga akhir masa magang, fitur dashboard utama admin masih dalam tahap pengembangan dan belum ditampilkan dalam laporan ini.

3.4 Kendala dan Solusi yang Ditemukan

Selama proses magang dan pengembangan aplikasi *Arraya Attendance*, penulis menghadapi beberapa kendala teknis dan non-teknis yang memengaruhi alur kerja. Adapun beberapa kendala utama yang ditemukan antara lain:

1. **Penggunaan Teknologi Baru:** Pada awal proyek, penulis belum memiliki pengalaman mendalam dalam menggunakan Golang sebagai bahasa backend dan PostgreSQL sebagai database utama. Hal ini menyebabkan penulis

perlu mengalokasikan waktu tambahan untuk mempelajari dasar-dasar pemrograman Golang dan query PostgreSQL yang efisien.

2. **Struktur API dan Validasi yang Ketat:** Penulis harus mengikuti standar struktur API yang telah ditetapkan oleh tim, termasuk sistem validasi form yang modular dan penggunaan Gin Binding. Penyesuaian terhadap pola kerja ini memerlukan ketelitian tinggi untuk menjaga konsistensi dan menghindari error saat testing.
3. **Manajemen Token dan Role-based Access:** Salah satu tantangan adalah memahami sistem otorisasi berbasis JWT Token serta alur login yang mengarahkan pengguna ke tampilan sesuai peran (staff, direktur, HC). Terkadang, terjadi kesalahan akses karena token kadaluarsa atau salah penggunaan role.
4. **Integrasi Gambar dan Lokasi:** Saat mengembangkan fitur absensi dengan foto dan geolokasi, penulis mengalami kesulitan dalam proses upload gambar menggunakan Cloudinary melalui API Next.js, terutama saat berpindah dari environment lokal ke production.
5. **Waktu Pengerjaan yang Terbatas:** Karena proyek dikerjakan dalam waktu magang yang relatif singkat, penulis perlu membagi waktu secara efisien antara pengembangan fitur, debugging, dokumentasi, dan koordinasi dengan tim.

Untuk mengatasi kendala-kendala yang telah dijelaskan pada subbab sebelumnya, penulis dan tim menerapkan berbagai solusi dan pendekatan sebagai berikut:

1. **Pembelajaran Mandiri dan Kolaboratif:** Untuk mengatasi keterbatasan pengetahuan tentang bahasa Golang dan PostgreSQL, penulis mengikuti dokumentasi resmi, tutorial daring, dan berkonsultasi langsung dengan Technical Lead. Hal ini membantu mempercepat proses pemahaman dan implementasi.
2. **Penerapan Standar Validasi Modular:** Penulis menerapkan sistem validasi form yang terstruktur dan reusable dengan mengikuti standar project yang telah ditentukan (menggunakan tag form, json, binding, serta fungsi resolve). Pendekatan ini membantu mempercepat debugging dan meningkatkan konsistensi antar fitur.

3. **Penggunaan Tools Otomatisasi dan Testing:** Untuk memastikan keakuratan alur autentikasi dan akses role-based, dilakukan pengujian menggunakan Postman dengan token-token simulasi dari berbagai peran. Environment variabel juga digunakan untuk membedakan antara endpoint lokal dan production.
4. **Debugging Upload Gambar dan GeoLokasi:** Permasalahan dalam upload gambar ke Cloudinary diatasi dengan mengevaluasi implementasi middleware formidable dan menyesuaikannya dengan arsitektur App Router milik Next.js 14. Selain itu, error lokasi juga ditangani dengan memastikan data latitude dan longitude sesuai format dan izin lokasi pada browser diaktifkan.
5. **Manajemen Waktu dan Prioritas Fitur:** Mengingat waktu magang yang terbatas, fitur-fitur dikembangkan berdasarkan skala prioritas. Fitur inti seperti login, check-in/out, dan pengajuan izin diselesaikan terlebih dahulu, sementara fitur pendukung seperti rekap detail harian disiapkan untuk fase penyempurnaan.

