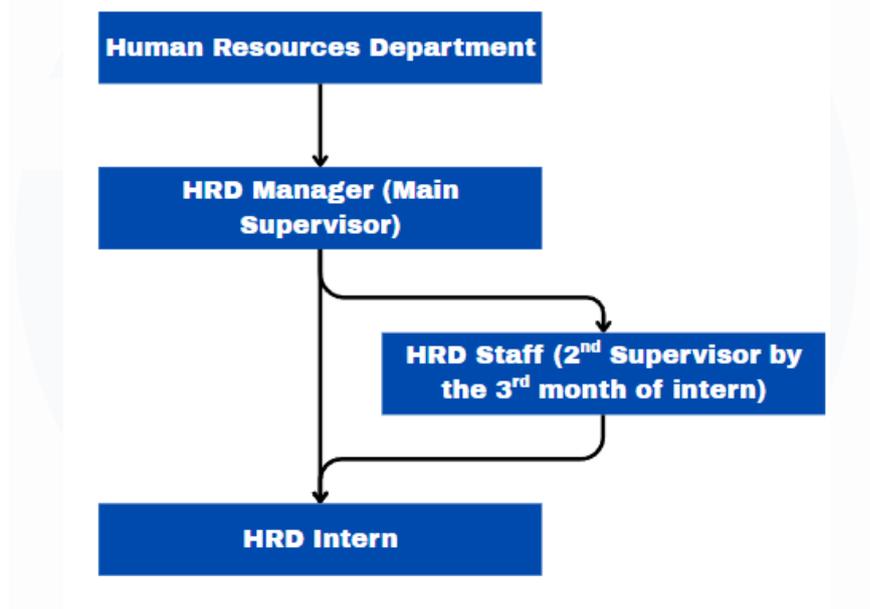


BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi



Gambar 3.1 Kududukan Posisi Kerja

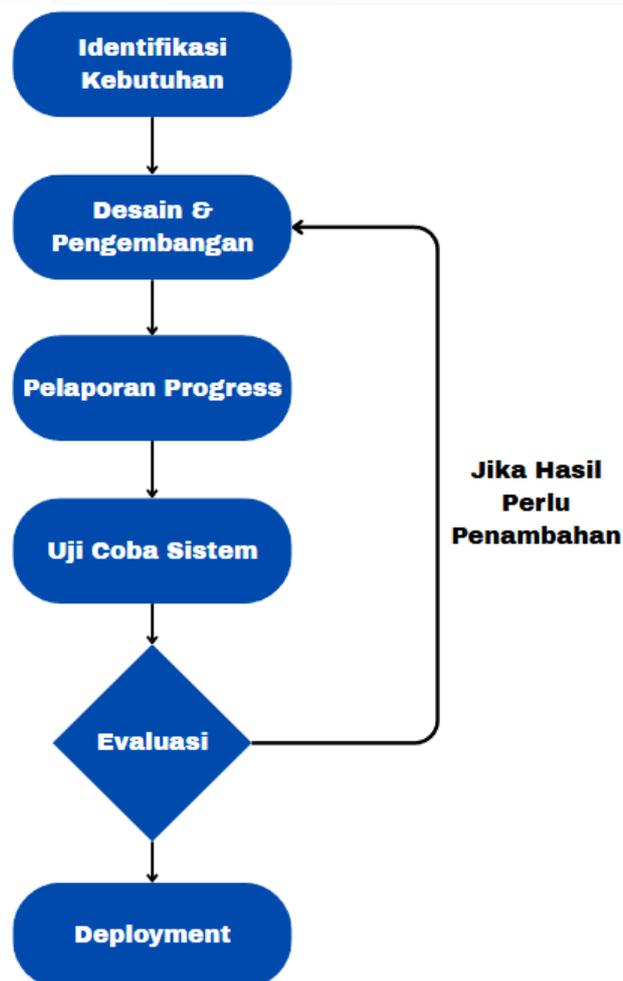
Selama pelaksanaan kerja magang di PT Mahardhika Samudera Sejahtera (MSS), saya ditempatkan di bawah koordinasi Departemen Human Resource Development (HRD). Tugas utama saya adalah merancang dan mengembangkan sistem absensi digital berbasis aplikasi mobile untuk karyawan dan dashboard web untuk admin HRD. Sistem ini dibangun dari awal sesuai dengan kebutuhan internal perusahaan dalam meningkatkan efisiensi pengelolaan data kehadiran karyawan.

Dalam struktur organisasi magang, posisi saya berada pada jalur teknis independen yang berperan sebagai penghubung antara kebutuhan operasional tim HRD dan solusi berbasis teknologi informasi. Meski bekerja secara remote, komunikasi dengan pihak perusahaan tetap terjaga dengan baik melalui pertemuan daring dan kunjungan langsung ke kantor saat diperlukan, terutama saat melakukan presentasi progress proyek, demo fitur, maupun validasi kebutuhan sistem.

Pada bulan-bulan awal magang, saya berkoordinasi langsung dengan supervisor utama dari tim HRD yang memberikan arahan umum mengenai rancangan sistem

dan fitur yang dibutuhkan. Namun, pada bulan Mei 2025, supervisor utama saya mulai tidak dapat mendampingi secara rutin karena keterlibatan dalam proyek lapangan di luar kota. Sebagai gantinya, saya diberikan supervisor kedua yang menggantikan peran pengawasan dan menjadi penghubung selama proses revisi dan finalisasi sistem. Peralihan ini berjalan dengan baik dan tidak mengganggu kelancaran pengembangan, karena semua dokumentasi dan catatan kerja telah ditata sejak awal.

Alur koordinasi kerja adalah sebagai berikut:



Gambar 3.2 Alur kerja

1. Identifikasi Kebutuhan: Tim HRD menyampaikan kebutuhan dan spesifikasi awal sistem absensi yang diinginkan, baik dari sisi fungsionalitas maupun tampilan antarmuka.

2. Desain dan Pengembangan: Saya mulai merancang dan membangun sistem menggunakan teknologi web dan mobile, menyesuaikan dengan kebutuhan perusahaan.
3. Pelaporan Progress: Setiap progress disampaikan secara berkala melalui laporan mingguan atau diskusi daring, baik kepada supervisor utama maupun supervisor kedua (pada bulan Mei).
4. Uji Coba Sistem: Setelah fitur utama selesai, dilakukan uji coba internal terhadap sistem absensi untuk mengevaluasi fungsionalitas, akurasi data, dan kemudahan penggunaan.
5. Evaluasi dan Deployment: Berdasarkan hasil uji coba dan masukan dari pihak HRD, dilakukan perbaikan dan penyempurnaan sistem hingga mencapai versi final yang stabil dan siap di-deploykan.

3.2 Tugas dan Uraian Kerja Magang

Tujuan Kerja Magang

Selama masa kerja magang di PT MSS Plant Maintenance, saya diberi tanggung jawab penuh untuk merancang, mengembangkan, dan mengimplementasikan sistem absensi karyawan berbasis digital. Sistem ini terdiri dari dua bagian utama, yaitu aplikasi mobile untuk karyawan dan dashboard web untuk admin HRD. Tujuan dari proyek ini adalah untuk menggantikan metode absensi manual dengan sistem yang lebih modern, efisien, dan terintegrasi secara real-time antara karyawan dan pihak manajemen HRD.

Timeline Mingguan Pengerjaan Magang

Selama masa magang, telah dikerjakan beberapa proyek yang telah ditugaskan. Rincian proyek yang dikerjakan dapat dilihat pada tabel berikut:

Tabel 3.2 Timeline Kerja Magang

Bulan	Minggu	Deskripsi
Maret	1	Perkenalan dengan tim HRD dan pendiskusian proyek
Maret	2	Pembuatan Database dan Backend Core System untuk website admin HRD
Maret	3	Pembuatan Frontend display untuk website admin HRD
Maret	4	Integrasi Backend dengan Frontend secara keseluruhan untuk website
April	1	Pembelajaran mandiri, serta uji-coba dalam pembuatan aplikasi berbasis mobile
April	2	Pembuatan system reverse geocoding untuk mobile dengan pembantuan eksternal dari Google Cloud, Serta pembuatan koneksi backend dari web dengan aplikasi mobile
April	3	Pembelajaran mandiri dalam publish ke Google Play Store
April	4	Pengisian dan pembuatan dokumen-dokumen yang dibutuhkan untuk Google Play
Mei	1	Penungguan validasi dokumen untuk Google Play, serta instruksi berikut oleh Supervisor
Mei	2	Penambahan fitur notifikasi email, dengan pembantuan eksternal <i>Brevo</i>
Mei	3	Penambahan fitur-fitur sisa seperti log <i>Check-in/Check-out</i> , permintaan foto selfie, dan membenaran display di table admin website

Mei	4	Penyelesaian, Bug Fixing, dan deployment/hosting untuk Website dan Mobile
-----	---	---------------------------------------------------------------------------

Landasan Teori

Framework utama yang digunakan dalam pengembangan sistem absensi karyawan ini adalah Laravel untuk backend. Laravel merupakan salah satu framework PHP paling populer yang menerapkan arsitektur Model-View-Controller (MVC). Laravel menyediakan berbagai fitur modern yang sangat membantu dalam membangun backend aplikasi web, seperti sistem routing, middleware, autentikasi pengguna, manajemen database dengan Eloquent ORM, serta dukungan keamanan terhadap serangan umum seperti SQL injection dan CSRF.

Di sisi frontend, sistem dibangun menggunakan React yang dipadukan dengan TypeScript. React adalah pustaka JavaScript yang digunakan untuk membangun *user interface* yang dinamis, modular, dan mudah dipelihara. React bekerja dengan konsep Virtual DOM dan berbasis komponen, sehingga memudahkan pengembang dalam memisahkan logika tampilan menjadi bagian-bagian kecil yang terorganisir. Sementara itu, penggunaan *TypeScript* dari *JavaScript* yang mendukung pengetikan data secara statis—membantu meningkatkan keandalan kode serta mempermudah debugging selama proses pengembangan.

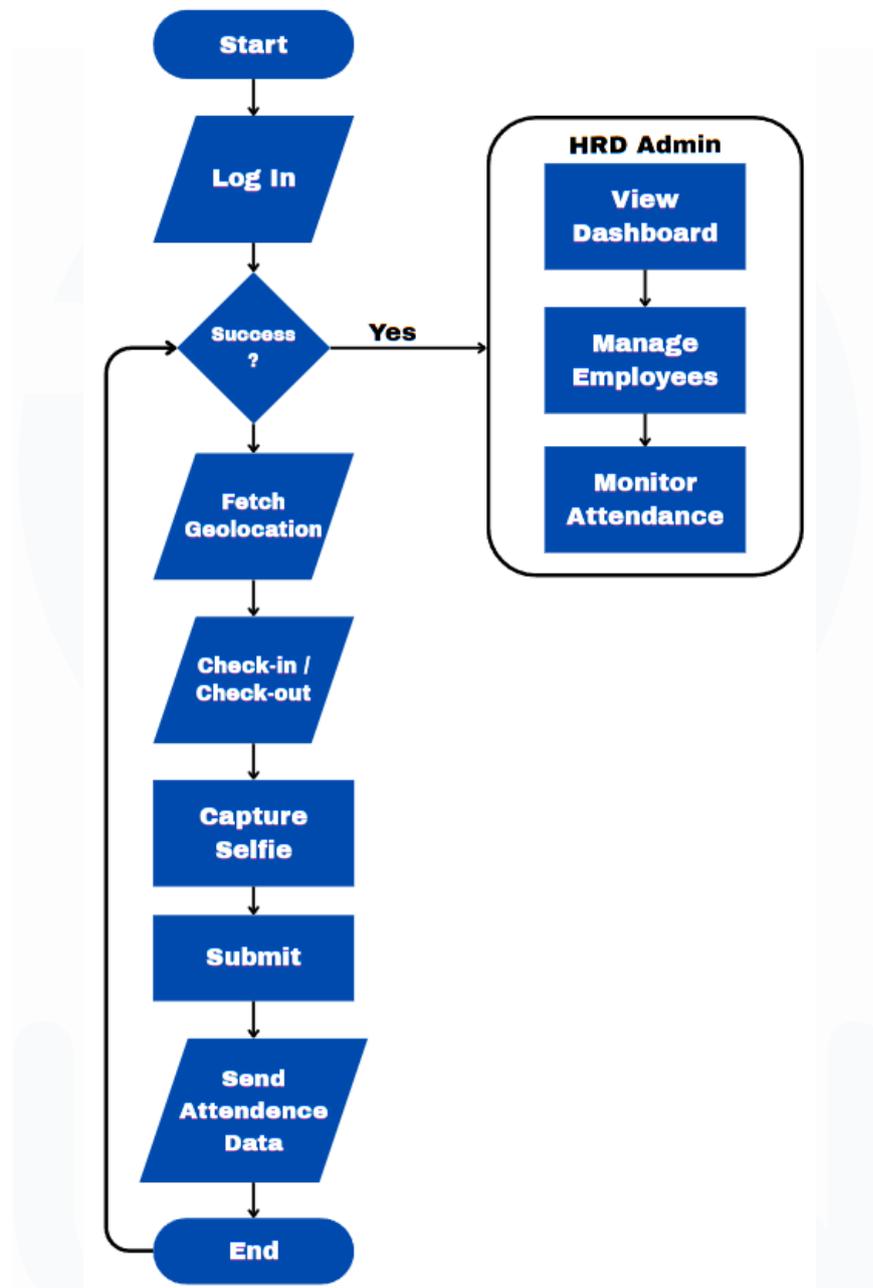
Untuk kebutuhan aplikasi mobile, digunakan *React Native*, yaitu framework yang memungkinkan pengembangan aplikasi Android menggunakan bahasa dan paradigma yang serupa dengan *React*. *React Native* memberikan akses ke fitur-fitur perangkat seperti kamera, GPS, dan penyimpanan lokal, sehingga sangat mendukung implementasi fungsi-fungsi penting seperti pengambilan foto selfie dan pelacakan lokasi untuk keperluan absensi.

Salah satu komponen penting dalam aplikasi absensi ini adalah proses reverse geocoding, yaitu konversi koordinat lokasi (latitude dan longitude) menjadi alamat fisik yang dapat dibaca oleh manusia. Fungsi ini diimplementasikan dengan

menggunakan layanan *Google Maps Geocoding API*, yang menyediakan data lokasi secara real-time dan akurat. Fitur ini tidak hanya menambah keakuratan data absensi, tetapi juga membantu admin dalam memverifikasi kehadiran berdasarkan lokasi aktual pengguna.

Selain itu, sistem juga mengintegrasikan fitur notifikasi email transaksional yang digunakan untuk mengirimkan pesan secara otomatis kepada pengguna. Fitur ini digunakan dalam berbagai skenario, seperti saat karyawan baru ditambahkan ke sistem, data mereka diperbarui oleh admin, atau saat permintaan reset password dilakukan. Untuk mendukung hal ini, sistem menggunakan layanan email pihak ketiga yaitu Brevo (sebelumnya dikenal sebagai Sendinblue). *Brevo* menyediakan API yang memungkinkan pengiriman email secara otomatis dan terintegrasi dengan backend Laravel, sehingga proses notifikasi dapat berjalan secara instan dan efisien.

Dalam keseluruhan arsitektur sistem, komunikasi antara frontend dan backend dilakukan menggunakan *Application Programming Interface (API)* berbasis REST. API digunakan untuk menghubungkan aplikasi mobile dan dashboard web dengan backend Laravel, termasuk untuk proses autentikasi, pengambilan data absensi, pengiriman data lokasi, hingga pengelolaan informasi karyawan. Dengan pendekatan ini, sistem menjadi lebih fleksibel, modular, dan siap dikembangkan lebih lanjut secara terpisah antara frontend dan backend.



Gambar 3.3 Flowchart proses keseluruhan sistem.

Flowchart di atas menggambarkan alur proses sistem absensi digital yang dikembangkan dalam proyek magang, yang terdiri dari dua bagian utama: proses dari sisi karyawan (melalui aplikasi mobile berbasis React Native) dan proses dari sisi admin HRD (melalui dashboard web berbasis Laravel dan React).

Proses dari Sisi Karyawan (Aplikasi Mobile)

- **Start:** Proses dimulai ketika karyawan membuka aplikasi absensi pada perangkat mobile mereka.
- **Log In:** Karyawan memasukkan kredensial login (email dan password) yang sudah didaftarkan sebelumnya oleh admin HRD ke dalam sistem. Sistem login memverifikasi kecocokan data terhadap database pada tabel employee.
- **Success?:** Jika login berhasil, pengguna akan diarahkan ke halaman utama aplikasi. Jika tidak, sistem menampilkan pesan kesalahan dan meminta input ulang. Proses login menggunakan auth API yang terhubung ke backend Laravel [5].
- **Fetch Geolocation:** Setelah login, aplikasi mengambil data lokasi perangkat secara otomatis menggunakan sensor GPS dan API geolokasi perangkat. Data berupa latitude dan longitude kemudian dikonversi menjadi nama alamat menggunakan Google Maps Reverse Geocoding API [6].
- **Check-In / Check-Out:** Karyawan memilih aksi absensi: check-in saat datang atau check-out saat pulang. Sistem akan mencatat jenis aksi ini dalam basis data.
- **Capture Selfie:** Sebelum data dikirim, pengguna diwajibkan mengambil foto selfie sebagai validasi kehadiran untuk mencegah absensi titip absen (buddy punching).
- **Submit:** Data berupa nama, waktu, status (check-in/check-out), lokasi (lat-long dan alamat), foto selfie, dan departemen dikompilasi dan disiapkan untuk dikirim.
- **Send Attendance Data:** Semua data dikirim ke server melalui API ke tabel transactions yang kemudian dapat diakses oleh admin HRD secara real-time melalui dashboard web. Proses ini menggunakan metode HTTP POST melalui library Axios di React Native [4].
- **End:** Proses absensi selesai.

Proses dari Sisi Admin HRD (Dashboard Web)

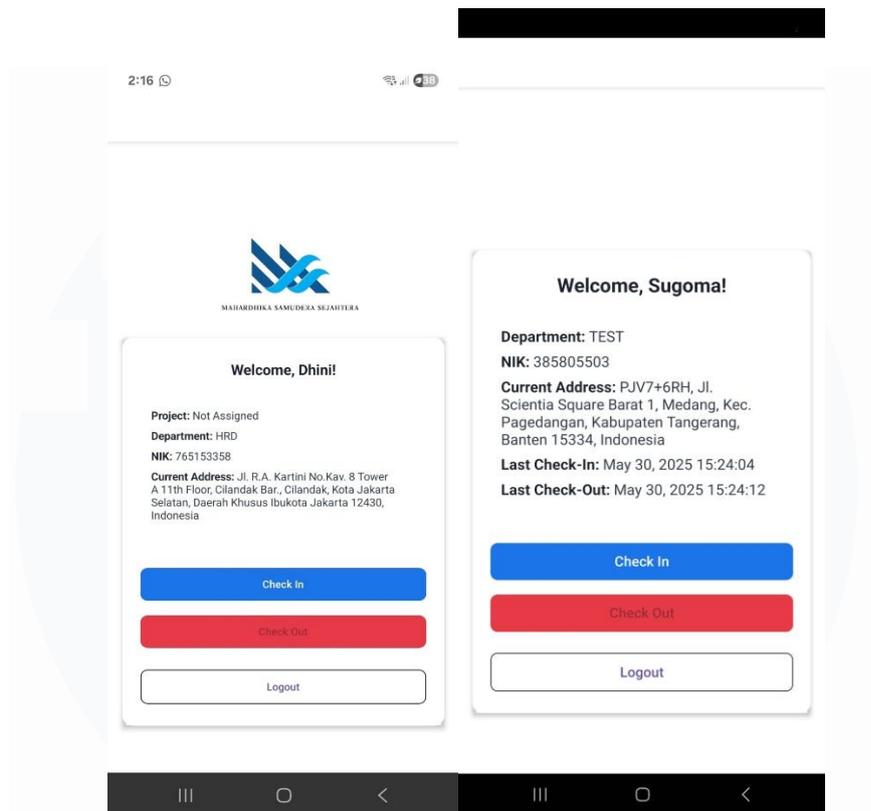
- **View Dashboard:** Admin HRD mengakses dashboard menggunakan akun khusus. Sistem web ini dibangun menggunakan framework Laravel untuk backend dan React + TypeScript untuk frontend [3][5].
- **Manage Employees:** Admin dapat menambahkan, mengedit, dan menghapus data karyawan. Setiap perubahan yang dilakukan akan mengirimkan notifikasi otomatis ke email karyawan menggunakan Brevo Transactional Email API [7]. Sistem ini juga mendukung fitur reset password yang mengirimkan kata sandi baru secara otomatis ke email yang terdaftar.
- **Monitor Attendance:** Admin dapat memantau data kehadiran secara real-time di halaman Transactions Table, lengkap dengan fitur filter berdasarkan nama, departemen, rentang tanggal, dan jam, serta ekspor data ke format Excel untuk kebutuhan rekap bulanan atau penggajian.

Flowchart ini menunjukkan bahwa sistem dirancang untuk meminimalisasi manipulasi data, mempercepat proses absensi, dan mempermudah pengelolaan SDM secara digital. Penggunaan integrasi teknologi seperti React Native untuk aplikasi mobile [4], Laravel dan React untuk dashboard [3][5], serta API dari Google Maps [6] dan Brevo [7] menjadikan sistem ini modern, fleksibel, dan efisien dalam menjawab kebutuhan internal PT MSS Plant Maintenance.

Pengembangan Aplikasi *Mobile* (Android):

Aplikasi mobile dikembangkan menggunakan *React Native*, yang memungkinkan aplikasi berjalan di perangkat Android. Aplikasi ini memiliki beberapa fitur utama sebagai berikut:

1. **Login Sistem:** Karyawan dapat login menggunakan kredensial yang telah dimasukkan oleh admin ke dalam sistem melalui website. Informasi ini mencakup nama, email, nomor HP, password, dan departemen masing-masing.



Gambar 3.4 Mobile Display

Pada Gambar 3.4 menampilkan antarmuka utama aplikasi mobile absensi karyawan setelah berhasil melakukan login. Tampilan ini dirancang menggunakan React Native [4] dan ditujukan untuk memberikan informasi kehadiran yang jelas dan real-time kepada setiap karyawan.

Setelah login berhasil, pengguna akan disambut dengan pesan “Welcome, (Nama Karyawan)!” yang bersifat dinamis sesuai dengan identitas pengguna yang terdaftar. Di bawahnya terdapat informasi personal dan status kehadiran terkini, yaitu:

- Department: menunjukkan divisi tempat karyawan bekerja, seperti “HRD”, “Operasional”, atau “Maintenance”.
- NIK: menampilkan Nomor Induk Karyawan yang bersifat unik untuk setiap individu.

- **Current Address:** adalah hasil konversi dari koordinat lokasi (latitude dan longitude) perangkat ke dalam bentuk alamat fisik, menggunakan Google Maps Reverse Geocoding API [6]. Alamat ini diperoleh secara otomatis saat pengguna membuka aplikasi dan memberikan izin akses lokasi.
- **Last Check-in:** menunjukkan waktu dan tanggal terakhir karyawan melakukan check-in dalam format Bulan/Tanggal/Tahun dan Jam, misalnya “07/30/2025 - 08:42”.
- **Last Check-out:** menunjukkan waktu dan tanggal terakhir karyawan melakukan check-out dengan format yang sama.

Di bagian bawah layar terdapat tiga tombol utama yang menjadi fitur inti dari aplikasi:

- **Check-In:** tombol ini akan memulai proses pencatatan kedatangan, di mana pengguna diminta mengaktifkan lokasi dan mengambil foto selfie sebagai bukti kehadiran.
- **Check-Out:** tombol ini digunakan saat karyawan akan menyelesaikan jam kerjanya dan mencatat waktu kepulangan.
- **Logout:** tombol ini memungkinkan pengguna keluar dari akun secara aman.

Desain tampilan dibuat sederhana, intuitif, dan responsif agar dapat diakses dengan nyaman oleh karyawan dari berbagai latar belakang teknis. Informasi yang ditampilkan bersifat real-time, karena seluruh data di-fetch dari database pusat melalui API backend yang dibangun menggunakan Laravel [5].

2. **Fetch Lokasi Real-Time:** Setelah login, aplikasi akan mengambil data lokasi pengguna berupa latitude dan longitude dari perangkat mereka.


```

const handleError = (error: { code: any; message: string | undefined; }) => {
  console.error("Geolocation error:", error);

  switch (error.code) {
    case 3:
      Alert.alert("Timeout Error", "Location request timed out. Try moving to an open area or enabling WiFi.");
      break;
    case 2:
      Alert.alert("Position Unavailable", "GPS signal is weak. Try moving outside or enabling WiFi.");
      break;
    case 1:
      Alert.alert("Permission Denied", "Location access denied. Please allow it in your device settings.");
      break;
    default:
      Alert.alert("Location Error", error.message);
  }
}

Geolocation.getCurrentPosition(
  (position) => handleSuccess(position),
  (fallbackError) => console.warn("Fallback location failed:", fallbackError),
  {
    enableHighAccuracy: false,
    timeout: 10000,
    maximumAge: 60000,
  }
);

Geolocation.getCurrentPosition(
  handleSuccess,
  handleError,
  {
    enableHighAccuracy: true,
    timeout: 20000,
    maximumAge: 0,
  }
);

```

Gambar 3.8 Snippet koding fetching lokasi bagian 4

Gambar 3.5 menunjukkan cuplikan kode yang digunakan dalam aplikasi mobile absensi berbasis React Native untuk melakukan proses pengambilan data lokasi (geolokasi) dan mengonversinya menjadi alamat fisik menggunakan layanan Google Maps Geocoding API [6].

1. Inisialisasi dan Pemanggilan Fungsi Lokasi

Pada bagian awal, terdapat dua useEffect hook:

- Hook pertama berfungsi untuk menginisialisasi library react-native-geocoding menggunakan API Key Google, dan langsung memanggil fungsi getLocation() saat komponen dimuat.
- Hook kedua memanggil loadCheckStatus() apabila status login (isLoggedIn) sudah berhasil, guna mengambil status terakhir dari absensi pengguna.

2. Izin Akses Lokasi

Fungsi `requestLocationPermission()` digunakan untuk meminta izin akses lokasi dari perangkat Android. Jika pengguna tidak memberikan izin, maka proses akan dihentikan, dan akan muncul peringatan menggunakan `Alert`.

3. Pengambilan Lokasi dan Proses Reverse Geocoding

Jika izin sudah diberikan, fungsi `getLocation()` akan memanggil `Geolocation.getCurrentPosition()` untuk mengambil latitude dan longitude dari perangkat pengguna. Data koordinat ini kemudian dikirim ke layanan Google Geocoding API, yang akan mengembalikan alamat lengkap (formatted address). Alamat yang berhasil dikonversi tersebut kemudian disimpan ke dalam state `address`, dan ditampilkan pada antarmuka pengguna.

4. Penanganan Kesalahan (Error Handling)

Jika terjadi kesalahan dalam proses pengambilan lokasi, fungsi `handleError()` akan menangani beberapa kemungkinan error, seperti:

- Timeout
- Sinyal GPS lemah
- Izin ditolak

Setiap kondisi error akan menampilkan alert berbeda untuk membantu pengguna memahami masalah yang sedang terjadi.

5. Fallback Mechanism

Jika proses geolokasi utama gagal, sistem mencoba metode fallback dengan konfigurasi `enable HighAccuracy: false` untuk mendapatkan lokasi dengan cara yang lebih sederhana.

Kode ini menjadi bagian fundamental dalam sistem absensi digital karena menjamin bahwa setiap proses Check-In dan Check-Out dilakukan dengan validasi lokasi yang akurat. Proses ini bertujuan mencegah kecurangan absensi seperti absensi jarak jauh (fake GPS) atau absen atas nama orang lain, sekaligus

memastikan bahwa aktivitas absensi benar-benar terjadi di area kerja sesuai yang ditentukan perusahaan.

```
const handleCheckIn = async () => {
  if (!employeeId || !location) {
    Alert.alert("Receiving Location", "Please wait for location to be ready.");
    return;
  }

  const hasPermission = await requestCameraPermission();
  if (!hasPermission) {
    Alert.alert("Permission Denied", "Camera permission is required to take a selfie.");
    return;
  }

  try {
    const capturedSelfie = await takeSelfie();
    if (!capturedSelfie) return;

    Alert.alert("Checking In", "Please wait...");
    await checkIn(
      employeeId,
      department,
      currentProject,
      location.lat,
      location.long,
      address,
      capturedSelfie,
      token
    );

    setSelfie(capturedSelfie);
    setHasCheckedIn(true);
    saveCheckStatus("checkedIn");
    await loadCheckStatus();
    Alert.alert("Success", "Checked in successfully!");
  } catch (error: any) {
    console.error("Check-in error:", error?.message || error);
    Alert.alert("Error", "Failed to check in. Please try again.");
  }
};
```

Gambar 3.9 Snippet koding absensi Check-in

Pembuatan Dashboard Web (Admin HRD):

Dashboard web untuk admin HRD dikembangkan dengan menggunakan Laravel sebagai backend dan React dengan TypeScript sebagai frontend. Komunikasi antara frontend dan backend dilakukan melalui API berbasis REST yang dibangun di Laravel dan di-fetch menggunakan library Axios pada sisi React. Arsitektur ini dipilih karena fleksibilitas, performa, dan kemudahan skalabilitasnya dalam pengembangan aplikasi modern.

```

<?php
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\TransactionController;
use App\Http\Controllers\AuthController;
use App\Http\Controllers\EmployeeController;

Route::post(uri: '/checkin', action: [TransactionController::class, 'checkIn']);
Route::post(uri: '/checkout', action: [TransactionController::class, 'checkOut']);
Route::get(uri: '/transactions', action: [TransactionController::class, 'getTransactions']);
Route::delete(uri: '/transactions/{id}', action: [TransactionController::class, 'destroy']);
Route::get(uri: '/employees', action: [EmployeeController::class, 'index']);
Route::post(uri: '/employees', action: [EmployeeController::class, 'store']);
Route::delete(uri: '/employees/{id}', action: [EmployeeController::class, 'destroy']);
Route::get(uri: '/employees/{id}/last-transaction', action: [TransactionController::class, 'getLastTransaction']);

Route::get(uri: '/user', action: function (Request $request): User {
    return $request->user();
})->middleware(middleware: 'auth:sanctum');

Route::post(uri: '/login', action: [AuthController::class, 'login']);
Route::post(uri: '/reset-password', action: [AuthController::class, 'resetPassword']);
Route::post(uri: '/logout', action: [AuthController::class, 'logout']);
Route::put(uri: '/employees/{id}/update-project', action: [EmployeeController::class, 'updateProject']);
Route::put(uri: '/employees/{id}', action: [EmployeeController::class, 'update']);

```

Gambar 3.10 Snippet kodingan API backend Laravel

Dashboard ini memiliki beberapa fitur utama yang mendukung fungsi administrasi kehadiran dan manajemen data karyawan, di antaranya:

Halaman Transactions Table (Data Absensi) Monitoring

ID	Name	NIK	Latitude	Longitude	Address	Department	Photo	Actions	Status	Time
17	Sugoma	385805503	-6.256890	106.614770	P.J.V7+6RH, Jl. Scientia Square Barat 1, Medang, Kec. Pagedangan, Kabupaten Tangerang, Banten 15334, Indonesia	HRD		Delete	Checked In	5/22/2025, 4:14:24 PM
18	Sugoma	385805503	-6.256890	106.614770	P.J.V7+6RH, Jl. Scientia Square Barat 1, Medang, Kec. Pagedangan, Kabupaten Tangerang, Banten 15334, Indonesia	HRD		Delete	Checked Out	5/22/2025, 4:14:40 PM
19	Sugoma	385805503	-6.256911	106.614772	P.J.V7+6RH, Jl. Scientia Square Barat 1, Medang, Kec. Pagedangan, Kabupaten Tangerang, Banten 15334, Indonesia	HRD		Delete	Checked In	5/22/2025, 4:42:27 PM
20	Sugoma	385805503	-6.256911	106.614772	P.J.V7+6RH, Jl. Scientia Square Barat 1, Medang, Kec. Pagedangan, Kabupaten Tangerang, Banten 15334, Indonesia	HRD		Delete	Checked Out	5/22/2025, 4:42:45 PM
22	Sugoma	385805503	-6.256858	106.614776	P.J.V7+6RH, Jl. Scientia Square Barat 1, Medang, Kec. Pagedangan, Kabupaten Tangerang, Banten 15334, Indonesia	HRD		Delete	Checked Out	5/24/2025, 3:52:20 PM
23	Sugoma	385805503	-6.256760	106.614899	P.J.V7+6RH, Jl. Scientia Square Barat 1, Medang, Kec. Pagedangan, Kabupaten Tangerang, Banten 15334, Indonesia	HRD		Delete	Checked In	5/24/2025, 3:52:37 PM
24	Sugoma	385805503	-6.256760	106.614899	P.J.V7+6RH, Jl. Scientia Square Barat 1, Medang, Kec. Pagedangan, Kabupaten Tangerang, Banten 15334, Indonesia	HRD		Delete	Checked Out	5/24/2025, 3:52:48 PM

48	Dhini	765153358	-6.293381	106.783769	Gg. H. Dun Blok Haji Nemin VI No.28, RT.4/RW.1, Lb. Bulus, Kec. Cilandak, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12440, Indonesia	HRD		Delete	Checked In	6/2/2025, 1:51:40 PM
49	Dhini	765153358	-6.293381	106.783769	Gg. H. Dun Blok Haji Nemin VI No.28, RT.4/RW.1, Lb. Bulus, Kec. Cilandak, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12440, Indonesia	HRD		Delete	Checked Out	6/2/2025, 1:51:50 PM
50	Dhini	765153358	-6.293812	106.784352	Unable to resolve address.	HRD		Delete	Checked In	6/19/2025, 1:12:04 PM
51	Dhini	765153358	-6.293692	106.784520	South Quarter, Jl. R.A. Kartini No.Kav.8 Tower A, 7th Floor, Unit G, Cilandak Bar., Kec. Cilandak, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12430, Indonesia	HRD		Delete	Checked Out	6/20/2025, 11:57:48 AM

Gambar 3.11 Preview website dashboard admin HRD

Gambar 3.11 merupakan pusat informasi kehadiran karyawan secara menyeluruh. Data absensi yang tampil mencakup:

- Nama karyawan
- Departemen
- Waktu absensi
- Lokasi (alamat hasil reverse-geocoding)
- Jenis aksi (Check-in / Check-out)
- Foto selfie yang diambil saat absensi

Fitur unggulan pada halaman ini antara lain:

- Filter berdasarkan nama karyawan
- Filter berdasarkan departemen
- Filter berdasarkan rentang tanggal (awal hingga akhir)
- Filter berdasarkan rentang waktu/jam (contoh: 08.00 – 17.00)
- Ekspor Data ke Excel

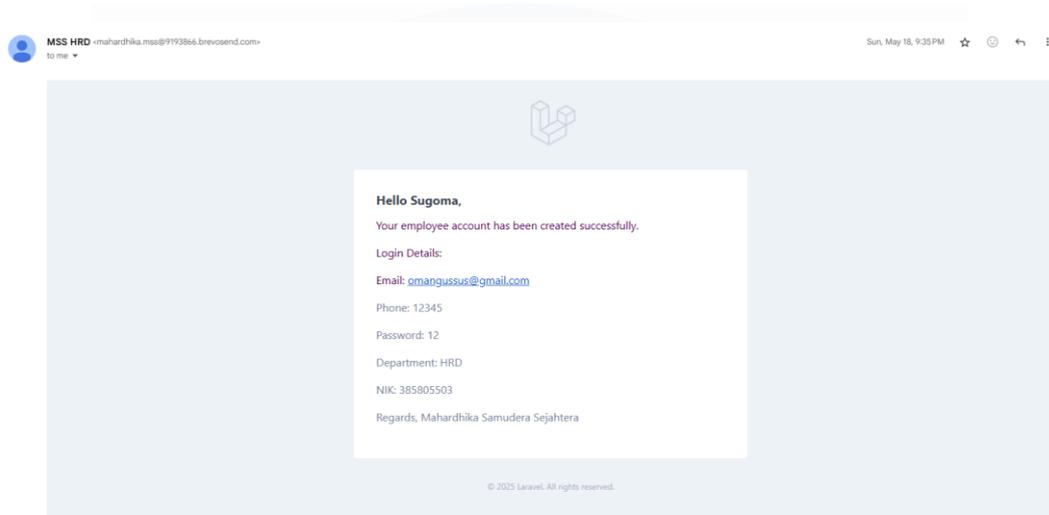
Halaman Employee (Manajemen Data Karyawan)

Employees List [Back to Transactions](#) [Add Employee](#)

Search by Name, Department, or Project

ID	Name	NIK	Email	Phone	Department	Actions
1	Sugoma	388606503	omangusus@gmail.com	12345542	TEST	Exp Delete
2	Dhini	765153358	dhikarize@gmail.com	08775347887	HRD	Exp Delete

Gambar 3.12 Preview halaman karyawan untuk admin HRD



Gambar 3.13 Preview hasil notifikasi email

Gambar 3.12 menampilkan antarmuka halaman Manajemen Data Karyawan yang diakses oleh admin HRD melalui dashboard web. Halaman ini dirancang menggunakan React + TypeScript untuk tampilan frontend dan terhubung ke backend Laravel yang mengelola data karyawan dari database. Antarmuka halaman ini memungkinkan admin untuk melakukan penambahan data karyawan baru, mengedit informasi yang sudah ada, dan menghapus data karyawan apabila diperlukan.

Formulir yang tersedia mencakup input seperti:

- Nama lengkap karyawan
- Alamat email aktif
- Nomor HP
- Password awal (dihash otomatis saat disimpan ke database)
- Departemen tempat karyawan bertugas
- NIK (Nomor Induk Karyawan).

Untuk memudahkan pengelolaan data dalam jumlah besar, halaman ini juga dilengkapi dengan fitur pencarian (search) dan filter berdasarkan departemen.

Desain antarmuka ditujukan agar mudah digunakan oleh admin non-teknis dan responsif terhadap berbagai ukuran layar. Setiap penambahan atau perubahan data karyawan akan secara otomatis mengirimkan notifikasi email kepada alamat email yang terdaftar. Fitur notifikasi ini dibangun menggunakan integrasi ke layanan email pihak ketiga bernama *Brevo* (sebelumnya dikenal sebagai *Sendinblue*), yang mendukung pengiriman email transaksional secara cepat dan aman.

Gambar 3.13 menunjukkan contoh hasil notifikasi email otomatis yang dikirimkan kepada karyawan setiap kali data mereka ditambahkan atau diperbarui oleh admin. Email ini dikirim melalui layanan pihak ketiga *Brevo* (sebelumnya *Sendinblue*), yang telah diintegrasikan ke dalam sistem backend menggunakan API transactional email [7].

Pesan email dapat berupa pemberitahuan akun baru, pembaruan informasi karyawan, atau pengiriman ulang password dalam kasus reset. Notifikasi ini tidak hanya membantu meningkatkan transparansi informasi, tetapi juga memastikan bahwa karyawan mengetahui jika terdapat perubahan data yang bersifat penting atau sensitif. Desain email dirancang sederhana dan profesional, serta mendukung format HTML agar dapat ditampilkan dengan baik di berbagai klien email.

```

<?php
namespace App\Notifications;

use Illuminate\Bus\Queueable;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Notifications\Messages\MailMessage;
use Illuminate\Notifications\Notification;
use App\Models\Employee;

2 references | 0 implementations
class NewEmployeeNotification extends Notification
{
    use Queueable;

    10 references
    private Employee $employee;
    2 references
    private string $password;
    2 references
    private bool $isUpdate;
    /**
     * Create a new notification instance.
     */
    1 reference | 0 overrides
    public function __construct(Employee $employee, $password = null, bool $isUpdate = false)
    {
        $this->employee = $employee;
        $this->password = $password;
        $this->isUpdate = $isUpdate;
    }

    /**
     * Get the notification's delivery channels.
     *
     * @return array<int, string>
     */
    0 references | 0 overrides
    public function via(object $notifiable): array
    {
        return ['mail'];
    }
}

```

Gambar 3.14 Snippet kodingan system notifikasi email bagian 1

Gambar 3.14 menampilkan bagian awal dari implementasi kode notifikasi email pada sistem absensi digital, yang dibuat menggunakan fitur Notification dari Laravel [5]. Kelas NewEmployeeNotification berfungsi untuk mengirim email secara otomatis kepada karyawan ketika data mereka ditambahkan atau diperbarui oleh admin melalui halaman manajemen karyawan di dashboard HRD.

Pada bagian ini, dapat dilihat bahwa kelas tersebut:

- Meng-extend class Notification milik Laravel dan menggunakan trait Queueable agar proses pengiriman email dapat dijalankan secara asinkron (dengan queue).
- Menerima dua parameter penting pada konstruktor:
 - Employee \$employee → objek data karyawan yang akan digunakan untuk pengisian isi email.
 - string \$password → password awal untuk akun karyawan (hanya digunakan saat akun baru dibuat).
 - bool \$isUpdate → penanda apakah notifikasi dikirim saat data baru ditambahkan (false) atau hanya diperbarui (true).
- Menentukan metode via() yang berfungsi untuk memilih saluran pengiriman notifikasi, dalam hal ini menggunakan 'mail' sebagai media pengiriman.

Implementasi ini menjadi pondasi logika notifikasi yang mendukung transparansi data, serta memastikan karyawan langsung mendapatkan informasi saat akun dibuat atau diubah.



```

/**
 * Get the mail representation of the notification.
 */
0 references | 0 overrides
public function toMail($notifiable): MailMessage
{
    if ($this->isUpdate) {
        return (new MailMessage)
            ->subject(subject: 'Your Employee Data Was Updated')
            ->line(line: 'Your employee information has been updated.')
            ->line(line: 'Name: ' . $this->employee->user_name)
            ->line(line: 'Department: ' . $this->employee->department)
            ->line(line: 'Current Project: ' . $this->employee->current_project)
            ->line(line: 'Phone: ' . $this->employee->phone)
            ->salutation(salutation: 'Regards, Mahardhika Samudera Sejahtera');
    }

    return (new MailMessage)
        ->subject(subject: 'Welcome! Your Employee Account is Created')
        ->greeting(greeting: 'Hello ' . $this->employee->user_name . ',')
        ->line(line: 'Your employee account has been created successfully.')
        ->line(line: 'Login Details:')
        ->line(line: 'Email: ' . $this->employee->email)
        ->line(line: 'Phone: ' . $this->employee->phone)
        ->line(line: 'Password: ' . $this->password)
        ->line(line: 'Department: ' . $this->employee->department)
        ->line(line: 'NIK: ' . $this->employee->nik)
        ->salutation(salutation: 'Regards, Mahardhika Samudera Sejahtera');
}

```

Gambar 3.15 Snippet kodngan system notifikasi email bagian 2

Gambar 3.15 melanjutkan kode dari kelas NewEmployeeNotification yang menampilkan logika utama dalam menyusun isi pesan email yang akan dikirim. Melalui fungsi toMail(), sistem akan:

- Mengecek apakah status isUpdate bernilai true:
 - Jika ya, maka sistem mengirim email dengan subjek “Your Employee Data Was Updated” yang berisi ringkasan data terbaru milik karyawan, seperti nama, departemen, proyek yang sedang dikerjakan, dan nomor telepon.
 - Jika tidak, maka sistem mengirim email sambutan “Welcome! Your Employee Account is Created”, yang juga menyertakan kredensial login awal seperti email, nomor HP, password, departemen, dan NIK.

- Email tersebut dikonstruksi menggunakan class MailMessage dari Laravel yang menyediakan template HTML otomatis dan standar tata letak yang baik.

Penggunaan email notifikasi ini memanfaatkan layanan pihak ketiga Brevo (Sendinblue) yang terintegrasi melalui API. Dengan demikian, pengiriman email menjadi lebih handal, cepat, dan aman dalam menyampaikan informasi akun yang bersifat personal atau sensitif [7].

```
MAIL_MAILER=smtp
MAIL_HOST=smtp-relay.brevo.com
MAIL_PORT=587
MAIL_USERNAME=8c498a001@smtp-brevo.com
MAIL_PASSWORD=
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS="mahardhika.mss@gmail.com"
MAIL_FROM_NAME="MSS HRD"
```

Gambar 3.16 Input Mailing System yang diberikan oleh Brevo

Gambar 3.16 menampilkan konfigurasi variabel lingkungan (environment variables) pada file .env di proyek Laravel yang digunakan untuk mengatur sistem pengiriman email melalui layanan pihak ketiga Brevo (sebelumnya Sendinblue) [7]. Konfigurasi ini menjadi inti integrasi antara backend Laravel dengan SMTP server eksternal milik Brevo, yang bertugas menangani pengiriman notifikasi email kepada karyawan.

Berikut penjelasan masing-masing baris konfigurasinya:

- MAIL_MAILER=smtp: Menentukan metode pengiriman email menggunakan SMTP protocol, standar umum untuk komunikasi email antarsistem.
- MAIL_HOST=smtp-relay.brevo.com: Alamat SMTP server milik Brevo yang menjadi endpoint utama pengiriman email.
- MAIL_PORT=587: Port 587 digunakan sebagai standar port untuk koneksi SMTP yang mendukung enkripsi TLS, memastikan keamanan data saat email dikirim.

- `MAIL_USERNAME=...`: Alamat email unik yang diberikan oleh Brevo untuk digunakan sebagai akun pengirim email (auth account).
- `MAIL_PASSWORD=...`: Merupakan API key atau token SMTP password yang dihasilkan dari dasbor akun Brevo. Data ini bersifat rahasia dan digunakan untuk otentikasi sistem agar dapat mengakses layanan email Brevo.
- `MAIL_ENCRYPTION=tls`: Mengaktifkan enkripsi TLS (Transport Layer Security) yang menjaga komunikasi antara server Laravel dan Brevo tetap aman.
- `MAIL_FROM_ADDRESS=mahardhika.mss@gmail.com`: Alamat email yang akan tampil sebagai pengirim (sender) pada kotak masuk penerima. Meskipun pengiriman secara teknis dilakukan lewat Brevo, tampilan di email penerima akan tetap menampilkan nama dan email perusahaan.
- `MAIL_FROM_NAME="MSS HRD"`: Nama pengirim yang akan tampil di bagian judul email (misalnya "MSS HRD"), sehingga terlihat lebih resmi dan profesional.

Dengan konfigurasi ini, sistem Laravel dapat mengirim email secara otomatis, baik untuk notifikasi pembuatan akun karyawan, perubahan data, hingga pengiriman ulang password. Proses ini sepenuhnya berlangsung di backend dan tidak memerlukan intervensi manual dari admin HRD.

Keamanan dan Validasi:

Dalam pengembangan sistem berbasis web dan mobile, aspek keamanan data menjadi salah satu prioritas utama, khususnya dalam menangani data sensitif pengguna seperti informasi akun dan kredensial login. Untuk itu, sistem absensi yang dikembangkan telah dilengkapi dengan berbagai mekanisme validasi input dan pengamanan data, terutama pada saat pembuatan akun karyawan baru oleh admin HRD.

id	user_name	email	phone	password	department	current_project	nik	created_at	updated_at
1	Sugoma	omangussus@gmail.com	12345542	\$2y\$12\$QH1scnLbsD9pxG8lcszOvQ2h43DxtqA5e3kmmgdb	TEST	Not Assigned	385805503	2025-05-18 14:35:47	2025-05-28 12:25:55
2	Dhini	dhiniarizqi@gmail.com	087775347887	\$2y\$12\$H81WbGK0lnuFPE7zwt1vgbsSzLpU/owFbZpJK	HRD	Not Assigned	765153358	2025-05-20 07:08:02	2025-05-26 06:09:02

Gambar 3.17 Preview Hashing password karyawan untuk keamanan

Gambar ini menampilkan representasi proses pengamanan password menggunakan metode hashing, yaitu teknik yang mengubah password asli menjadi bentuk terenkripsi sebelum disimpan di dalam database. Teknik ini diterapkan menggunakan fungsi Hash::make() milik Laravel, yang memanfaatkan algoritma hashing modern seperti bcrypt [5].

Dengan menggunakan hashing, sistem tidak pernah menyimpan password asli (plaintext) secara langsung, sehingga jika database mengalami kebocoran, password tidak bisa dibaca dengan mudah oleh pihak yang tidak berwenang. Praktik ini merupakan standar industri dalam pengamanan kredensial pengguna dan mencegah risiko pencurian identitas (identity theft).

```

1 reference | 0 overrides
public function store(Request $request): JsonResponse|mixed
{
    $request->validate(rules: [
        'user_name' => 'required|unique:employees',
        'email' => 'required|email|unique:employees',
        'phone' => 'required|string|',
        'password' => 'required',
        'department' => 'required'
    ]);

    $plainPassword = $request->password;

    $employee = Employee::create(attributes: [
        'user_name' => $request->user_name,
        'email' => $request->email,
        'phone' => $request->phone,
        'password' => Hash::make(value: $plainPassword),
        'department' => $request->department
    ]);

    $employee->notify(instance: new NewEmployeeNotification(employee: $employee, password: $plainPassword, isUpdate: false));

    return response()->json(data: $employee, status: 201);
}

```

Gambar 3.18 Snippet kodingan pembuatan karyawan baru

Gambar 3.18 menunjukkan cuplikan kode fungsi `store()` dalam controller Laravel yang digunakan untuk menyimpan data karyawan baru ke dalam database.

Langkah-langkah Proses:

1. Validasi Input

Sistem memverifikasi setiap data yang dikirim melalui form input agar tidak ada data kosong, format email salah, atau duplikasi username/email. Proses validasi ini dilakukan menggunakan perintah `validate()`

2. Hashing Password

Password yang diinput oleh admin akan terlebih dahulu di-hash menggunakan fungsi `Hash::make()` sebelum disimpan

3. Penyimpanan ke Database

Semua data karyawan disimpan ke dalam tabel `employees` melalui fungsi `Employee::create()`.

4. Pengiriman Notifikasi Email

Setelah data berhasil disimpan, sistem akan mengirimkan notifikasi email ke karyawan menggunakan kelas `NewEmployeeNotification`, yang berisi informasi akun seperti username, password awal, dan departemen.

5. Respon JSON

Sistem mengembalikan respon dalam format JSON dengan kode status 201 (Created) yang digunakan oleh frontend untuk memberi tahu bahwa proses berhasil.

Hosting dan Deployment Aplikasi

Dalam rangka memastikan aplikasi absensi dapat diakses secara online oleh pihak perusahaan dan karyawan, sistem ini di-hosting menggunakan layanan dari Hostinger, sebuah penyedia layanan web hosting yang populer karena kemudahan

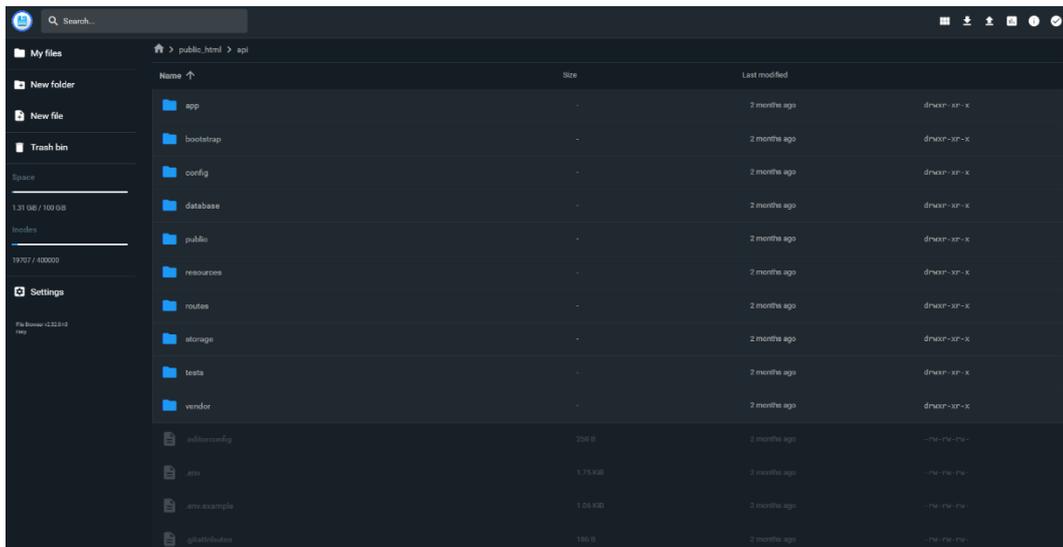
penggunaannya dan dukungan teknis yang baik untuk pengembangan aplikasi modern.

Alasan Pemilihan Hostinger:

- Dukungan PHP dan Laravel Framework
- Panel kontrol (hPanel) yang ramah pengguna
- Database MySQL yang dapat diakses langsung melalui phpMyAdmin
- SSL untuk keamanan koneksi
- Ketersediaan FTP dan File Manager untuk pengelolaan file aplikasi

Proses Deployment Backend (Laravel):

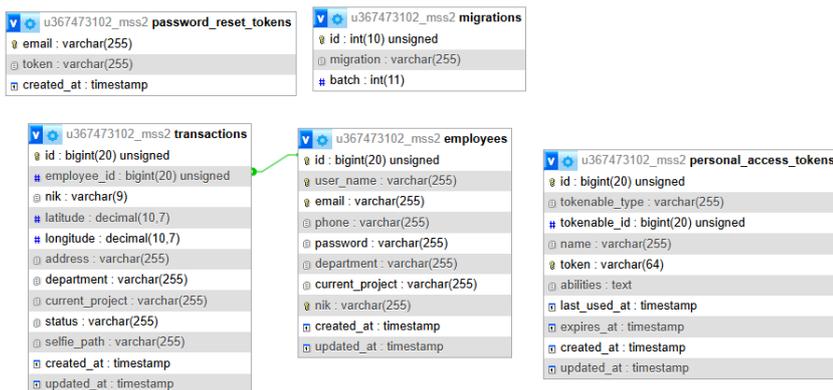
- Kode backend Laravel di-upload ke server Hostinger melalui File Manager.
- File konfigurasi .env disesuaikan agar mengarah ke database MySQL Hostinger.
- Perintah artisan seperti `php artisan migrate` dan `php artisan config:cache` dijalankan melalui terminal SSH atau fitur terminal bawaan Hostinger.
- Domain dikaitkan dengan folder public milik Laravel sebagai root directory.
-



Gambar 3.19 Preview file-file backend di file manager Hostinger

Database Online

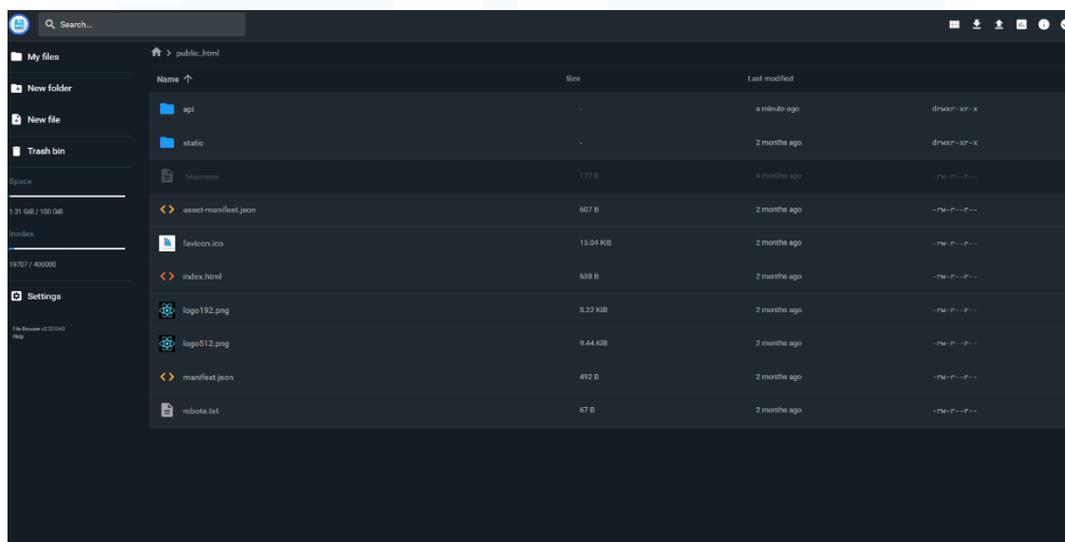
- Seluruh data karyawan dan log transaksi absensi disimpan pada database MySQL yang dikelola langsung melalui phpMyAdmin Hostinger.
- Tabel-tabel penting seperti employees dan transactions dapat diakses dan dikelola melalui koneksi Laravel ke database tersebut.



Gambar 3.20 Tabel ERD

Pengelolaan Versi Frontend:

- Aplikasi frontend React (admin dashboard) dikompilasi menggunakan npm run build dan hasil build dimasukkan ke dalam folder khusus di Hostinger (misalnya dashboard).
- File hasil build akan diakses langsung melalui domain atau subfolder yang ditentukan.



Gambar 3.21 Preview file-file frontend di file manager Hostinger

Keamanan dan Performa:

- Sertifikat SSL dari Hostinger digunakan untuk memastikan koneksi antara klien dan server tetap aman (HTTPS).
- Backup data dapat dijadwalkan melalui fitur otomatis di dashboard Hostinger.
- Penggunaan .htaccess untuk redirect dan pengamanan endpoint tertentu juga diterapkan.

Pengujian dan Pemeliharaan Sistem:

- Sebelum sistem digunakan secara penuh, saya melakukan uji coba internal untuk memastikan seluruh fungsi berjalan sesuai dengan skenario yang diinginkan, baik dari sisi aplikasi mobile maupun web.
- Setelah mendapat masukan dari pihak HRD, saya melakukan penyesuaian terhadap tampilan UI, alur kerja, serta optimasi performa aplikasi agar lebih stabil dan mudah digunakan.
- Perawatan dan debugging dilakukan secara berkala untuk memastikan tidak terjadi error pada sistem terutama saat proses upload foto, pengambilan lokasi, dan integrasi dengan API eksternal seperti Google Cloud.

3.3 Kendala yang Ditemukan

Beberapa kendala yang saya hadapi selama proses kerja magang antara lain:

- Pengembangan Sistem Secara Mandiri: Seluruh sistem dikembangkan secara individual, sehingga memerlukan waktu dan tenaga ekstra, serta seringkali harus mempelajari metode baru secara mandiri.
- Kurangnya Referensi Internal: Tidak adanya dokumentasi teknis sebelumnya membuat proses pengembangan harus dimulai dari nol tanpa bantuan yang jelas.

3.4 Solusi atas Kendala yang Ditemukan

Untuk mengatasi berbagai kendala tersebut, saya menerapkan beberapa strategi berikut:

- Pembelajaran Mandiri: Saya secara aktif mencari referensi dari dokumentasi resmi, forum, dan tutorial online untuk mempercepat pemahaman terhadap teknologi yang digunakan.
- Perencanaan dan Manajemen Waktu: Saya membuat daftar prioritas dan timeline kerja pribadi agar seluruh fitur inti dapat diselesaikan tepat waktu.

- Koordinasi Berkala: Saya menjadwalkan pertemuan rutin secara online atau langsung ke kantor jika diperlukan agar komunikasi dengan tim HRD tetap terjaga dengan baik.

