**Dokumen Desain Perangkat Lunak (Software**

**Design Document - SDD) Modul Presensi dan Absensi**

**1. Pendahuluan**

**1.1 Tujuan**

Dokumen ini bertujuan untuk menjelaskan desain teknis dari Modul Presensi dan Absensi dalam sistem Perrollingan dan Absensi Karyawan Humas Kominfo Kota Bima. Modul ini merupakan bagian penting dari sistem informasi yang berfungsi untuk mencatat kehadiran karyawan secara real-time dan terintegrasi dengan sistem jadwal peliputan.

Dokumen ini menyediakan gambaran lengkap mengenai arsitektur perangkat lunak, komponen-komponen utama, antarmuka, alur kerja, serta detail teknis lainnya yang akan digunakan oleh tim pengembang dalam proses implementasi modul. Dengan adanya dokumen ini, diharapkan proses pengembangan menjadi lebih terarah, terstruktur, dan meminimalkan kesalahan dalam implementasi sistem.

Modul ini juga akan mendukung efisiensi dalam proses administrasi kehadiran, memungkinkan pencatatan otomatis dan pelaporan kehadiran yang akurat.

**1.2 Ruang Lingkup**

Sistem informasi ini dirancang untuk mengelola data pengguna dan kehadiran karyawan secara efisien dalam lingkungan kerja Humas Kominfo Kota Bima. Modul Presensi dan Absensi merupakan bagian inti dari sistem, yang mendukung proses pencatatan kehadiran berbasis digital serta penyusunan jadwal peliputan secara otomatis dan terstruktur.

Fitur utama dalam ruang lingkup pengembangan modul ini meliputi:

* **Autentikasi dan Otorisasi Pengguna**  
  Menyediakan sistem login yang aman dengan pembagian hak akses berdasarkan peran (admin, karyawan, supervisor, dll).
* **Manajemen Data Berbasis CRUD (Create, Read, Update, Delete)**  
  Memungkinkan pengelolaan data pengguna, data presensi, dan jadwal peliputan secara fleksibel dan efisien.
* **Pembuatan Laporan dan Analitik**  
  Menyediakan fitur laporan kehadiran harian, mingguan, atau bulanan serta visualisasi data statistik untuk mendukung pengambilan keputusan.
* **Integrasi API untuk Komunikasi dengan Sistem Eksternal**  
  Modul mendukung API RESTful yang memungkinkan integrasi dengan sistem lain, seperti sistem informasi kepegawaian atau sistem pengelolaan jadwal kota.
* **Mekanisme PencadanganData**

Sistem menyediakan fitur backup data secara otomatis maupun manual untuk menjaga keberlangsungan layanan dan mencegah kehilangan data penting.

Modul ini akan dirancang agar bersifat skalabel, mudah diintegrasikan, dan dapat dikembangkan lebih lanjut sesuai dengan kebutuhan organisasi di masa depan.

**1.3 Referensi**

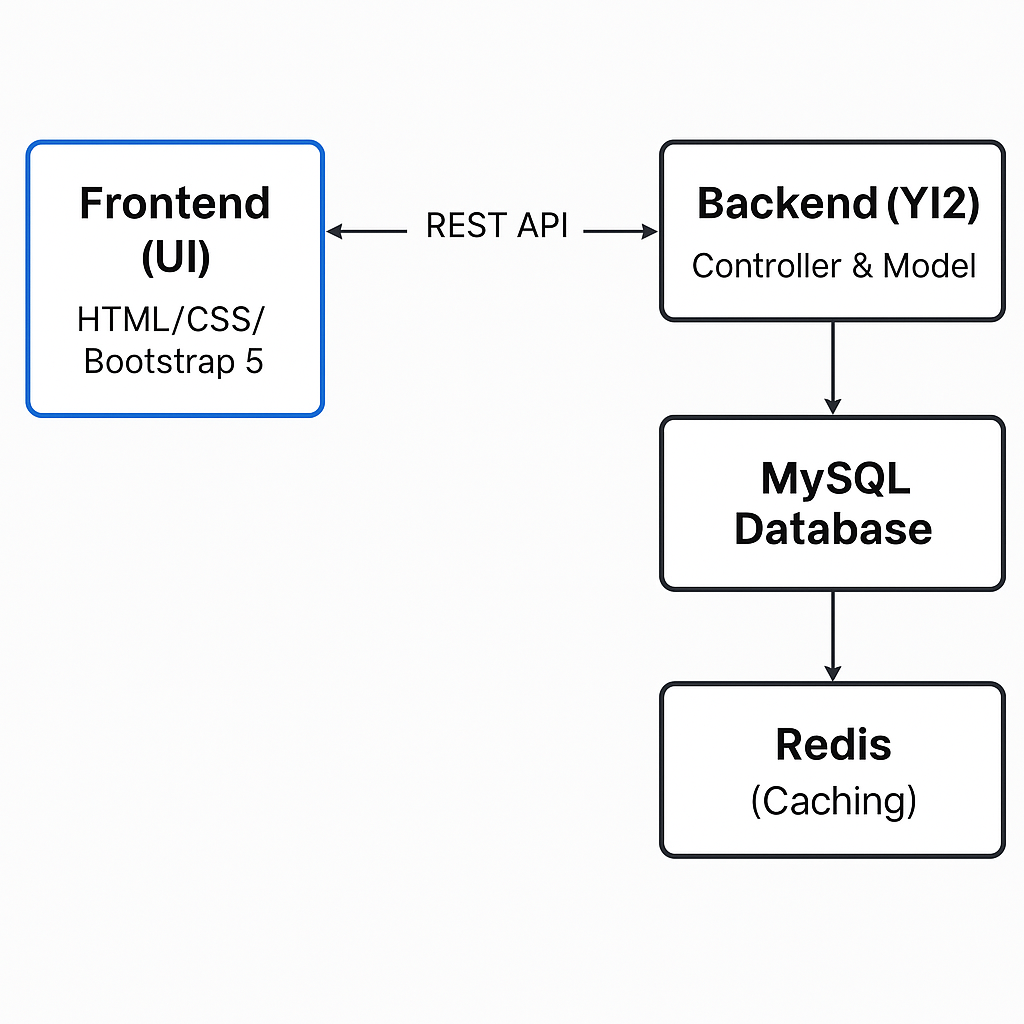
Dokumen ini disusun dengan merujuk pada beberapa sumber dan standar teknis yang relevan untuk memastikan kualitas desain perangkat lunak yang baik, efisien, dan aman. Referensi yang digunakan antara lain:

* **IEEE 1016-2009: Standard for Software Design Description**  
  Merupakan standar internasional yang digunakan sebagai pedoman dalam penyusunan dokumen desain perangkat lunak secara sistematis dan profesional.
* **Dokumentasi Yii2 Framework**  
  Digunakan sebagai acuan utama dalam pengembangan sistem berbasis Yii2, mencakup struktur MVC, penggunaan ActiveRecord, dan implementasi fitur keamanan.
* **Panduan Bootstrap 5 untuk Desain Antarmuka Pengguna**  
  Digunakan dalam perancangan antarmuka pengguna (UI) agar responsif, modern, dan mudah digunakan.
* **Panduan Keamanan OWASP (Open Web Application Security Project)**  
  Digunakan sebagai rujukan dalam menerapkan praktik terbaik keamanan aplikasi web, termasuk pencegahan terhadap serangan umum seperti SQL Injection, XSS, dan CSRF.

**2. Desain Arsitektur**

**2.1 Diagram Arsitektur Sistem**

Sistem Presensi dan Absensi ini menggunakan arsitektur berbasis **MVC (Model-View-Controller)** untuk memisahkan logika bisnis, tampilan antarmuka pengguna, dan pengelolaan data. Pendekatan ini memudahkan pengembangan, pemeliharaan, serta skalabilitas sistem.



**Komponen Teknologi:**

* **Backend**:  
  Menggunakan **PHP Yii2 Framework** sebagai kerangka kerja utama. Yii2 menyediakan struktur MVC yang kuat, sistem routing, ORM (ActiveRecord), dan berbagai fitur keamanan bawaan.
* **Frontend**:  
  Dibangun menggunakan kombinasi **HTML, CSS**, dan **Bootstrap 5**, untuk menciptakan antarmuka yang modern, responsif, dan user-friendly.
* **Database**:  
  Sistem menggunakan **MySQL** sebagai database relasional untuk menyimpan data pengguna, jadwal, kehadiran, dan log sistem.
* **API**:  
  Mendukung **RESTful API** untuk memungkinkan sistem ini berkomunikasi dengan aplikasi atau sistem eksternal seperti sistem kepegawaian atau pelaporan kota.
* **Caching**:  
  Menggunakan **Redis** sebagai cache layer untuk mempercepat proses pemanggilan data yang sering diakses, sehingga meningkatkan performa keseluruhan sistem.

**2.2 Komponen Utama Sistem**

Sistem ini dibangun menggunakan arsitektur **MVC (Model-View-Controller)** yang dilengkapi dengan komponen tambahan untuk mendukung integrasi dan performa sistem. Berikut adalah penjelasan dari masing-masing komponen utama:

1. **Model**  
   Komponen ini bertanggung jawab atas pengelolaan logika data serta interaksi langsung dengan database. Model digunakan untuk mengambil, menyimpan, memperbarui, dan menghapus data dari MySQL dengan memanfaatkan fitur **ActiveRecord** dari Yii2.
2. **View**  
   Merupakan antarmuka pengguna yang dirancang menggunakan HTML, CSS, dan Bootstrap 5. View menyajikan data yang telah diolah oleh controller dalam bentuk tampilan yang interaktif, responsif, dan mudah digunakan oleh pengguna akhir.
3. **Controller**  
   Controller bertugas sebagai penghubung antara Model dan View. Komponen ini mengelola alur data, menangani request dari pengguna, memanggil model untuk pengolahan data, dan meneruskan hasilnya ke View untuk ditampilkan.
4. **Database**  
   Data disimpan secara terstruktur dalam sistem manajemen basis data **MySQL**. Database ini menyimpan informasi penting seperti data pengguna, jadwal presensi, riwayat kehadiran, dan log aktivitas. Struktur tabel dirancang untuk menjaga integritas dan konsistensi data.
5. **API Layer**  
   Sistem ini menyediakan **RESTful API** sebagai jalur komunikasi antar sistem dan untuk mendukung integrasi dengan layanan eksternal, seperti sistem kepegawaian, aplikasi mobile, atau dashboard monitoring.
6. **Caching Layer**  
   Untuk meningkatkan performa, sistem menggunakan **Redis** sebagai caching layer. Redis menyimpan data sementara yang sering diakses (seperti daftar pengguna aktif, jadwal peliputan hari ini, dsb.) agar tidak perlu diambil berulang dari database, sehingga mempercepat proses pengambilan data dan mengurangi beban server.

**3. Desain Modul dan Komponen**

**3.1 Modul Autentikasi Pengguna**

Modul ini berfungsi sebagai gerbang utama untuk memastikan bahwa hanya pengguna yang terverifikasi yang dapat mengakses sistem. Proses autentikasi dilakukan secara aman dengan memanfaatkan teknologi modern seperti OAuth2 dan JWT.

**• Fungsi Utama:**

* **Login:** Memverifikasi kredensial pengguna untuk memberikan akses ke sistem.
* **Logout:** Mengakhiri sesi pengguna dan menghapus token autentikasi.
* **Registrasi:** Mendaftarkan pengguna baru dengan validasi data awal.
* **Pemulihan Kata Sandi:** Menyediakan fitur reset password melalui email atau token verifikasi.

**• Input:**

* **Username / Email**
* **Password**
* **Token Autentikasi (untuk validasi lanjutan / reset password)**

**• Output:**

* **Token Sesi Pengguna (JWT)**
* **Status Autentikasi (berhasil/gagal)**
* **Pesan kesalahan atau keberhasilan**

**• Teknologi Digunakan:**

* **OAuth2:** Untuk manajemen autentikasi yang aman dan terstandarisasi.
* **JWT (JSON Web Token):** Untuk menyimpan informasi sesi pengguna dalam bentuk token yang dapat diverifikasi secara independen.

**• Validasi Keamanan:**

* **Captcha:** Diterapkan saat proses login untuk mencegah serangan brute force dan bot otomatis.
* **Rate Limiting:** Pembatasan jumlah percobaan login dalam rentang waktu tertentu untuk mencegah penyalahgunaan sistem.

**3.2 Modul Manajemen Data**

Modul ini berperan dalam pengelolaan data inti dalam sistem, mencakup proses **CRUD (Create, Read, Update, Delete)** terhadap entitas seperti pengguna, data presensi, transaksi, serta data peliputan.

**• Fungsi Utama:**

* **Create:** Menambahkan data baru ke dalam sistem.
* **Read:** Menampilkan atau mengambil data yang tersimpan.
* **Update:** Memperbarui informasi yang sudah ada.
* **Delete:** Menghapus data dari database jika sudah tidak relevan atau diperlukan.

**• Input:**

* Data pengguna (nama, email, peran, status, dsb.)
* Data transaksi (tanggal, deskripsi, nilai, dsb.)
* Parameter pencarian/filter untuk listing data

**• Output:**

* Data yang tersimpan atau diperbarui dalam **database MySQL**
* Respons API dalam format **JSON**, seperti status keberhasilan, pesan kesalahan, dan data hasil query

**• Teknologi Digunakan:**

* **ORM Yii2 (ActiveRecord):** Untuk mempermudah interaksi dengan database secara OOP dan efisien.
* **Mekanisme Indexing Database:** Digunakan untuk mengoptimalkan performa query terutama saat menangani pencarian data dalam jumlah besar.
* **Validasi Data Otomatis:** Menggunakan rules bawaan dari Yii2 untuk memastikan integritas data yang masuk ke sistem.

**3.3 Modul Laporan dan Analitik**

Modul ini berfungsi untuk menghasilkan laporan berdasarkan data yang tersimpan dalam sistem, serta menyediakan visualisasi dalam bentuk grafik statistik untuk mempermudah analisis dan pengambilan keputusan.

**• Fungsi Utama:**

* **Pembuatan Laporan:** Menyusun laporan otomatis berdasarkan data pengguna, aktivitas presensi, rekap peliputan, dan statistik kinerja.
* **Visualisasi Data:** Menampilkan grafik seperti bar chart, pie chart, dan line chart untuk memperjelas tren data.
* **Ekspor Laporan:** Menyediakan opsi ekspor ke dalam format **PDF** dan **Excel** untuk keperluan distribusi atau dokumentasi offline.

**• Output:**

* **File PDF** (dengan struktur rapi dan bisa dicetak)
* **File Excel (.xlsx)** (untuk keperluan olah data lebih lanjut)
* **Grafik Statistik Interaktif** (terintegrasi dalam dashboard atau halaman laporan)

**• Teknologi Digunakan:**

* **MPDF:** Untuk menghasilkan file PDF dari halaman HTML yang diformat dengan CSS.
* **PhpSpreadsheet:** Digunakan untuk pembuatan dan ekspor file Excel (.xlsx).
* **Chart.js:** Library JavaScript yang digunakan untuk menampilkan grafik data secara interaktif di tampilan web.

**3.4 Modul Integrasi API**

Modul ini berfungsi sebagai penghubung antara sistem dengan layanan eksternal, memungkinkan pertukaran data dan integrasi fitur tambahan seperti sistem pembayaran, notifikasi, maupun layanan pihak ketiga lainnya.

**• Fungsi Utama:**

* **Integrasi Sistem Pihak Ketiga:** Memungkinkan sistem untuk berkomunikasi dengan layanan eksternal seperti sistem kepegawaian, gateway pembayaran, atau notifikasi (email/SMS).
* **Pengambilan & Pengiriman Data:** Mengirim data tertentu dari sistem ke layanan eksternal, serta menerima respons dan hasil dari proses tersebut.
* **Pemrosesan Respons:** Menangani respons yang diterima dari layanan eksternal untuk ditampilkan atau disimpan sesuai kebutuhan.

**• Input:**

* **Permintaan (Request)** dalam format **JSON**, berisi data yang dibutuhkan oleh layanan eksternal (misalnya: ID transaksi, data pengguna, dll).

**• Output:**

* **Respons API** dari sistem eksternal dalam format JSON, yang dapat berisi data hasil proses, status, atau pesan kesalahan.

**• Teknologi Digunakan:**

* **RESTful API:** Protokol standar yang digunakan untuk komunikasi antar sistem melalui HTTP dengan pendekatan CRUD.
* **Autentikasi API Key:** Setiap permintaan ke atau dari layanan eksternal dilengkapi dengan API key untuk menjaga keamanan dan otorisasi akses.
* **Yii2 REST Controller:** Untuk mendefinisikan endpoint yang akan diakses oleh pihak ketiga secara aman dan terstruktur.

**4. Desain Basis Data**

**4.1 Model Data**

Struktur basis data dirancang untuk mendukung fungsi utama sistem seperti manajemen pengguna, pencatatan aktivitas, transaksi, dan integrasi API. Berikut adalah model data utama yang digunakan:

**1. users**

Digunakan untuk menyimpan informasi pengguna sistem.

| Kolom | Tipe Data | Keterangan |
| --- | --- | --- |
| id | INT (PK) | ID unik pengguna |
| nama | VARCHAR | Nama lengkap pengguna |
| email | VARCHAR | Alamat email (unik) |
| password | VARCHAR | Password yang sudah di-hash |
| role | ENUM | Peran pengguna (admin, staf, viewer) |
| created\_at | DATETIME | Waktu pembuatan akun |
| updated\_at | DATETIME | Waktu terakhir pembaruan data pengguna |

**2. transactions**

Mencatat data transaksi atau aktivitas pelaporan tertentu.

| Kolom | Tipe Data | Keterangan |
| --- | --- | --- |
| id | INT (PK) | ID unik transaksi |
| user\_id | INT (FK) | ID pengguna terkait (relasi ke users) |
| amount | DECIMAL | Nilai transaksi |
| date | DATE | Tanggal transaksi |
| status | ENUM | Status transaksi (pending, selesai) |

**3. logs**

Merekam aktivitas pengguna sebagai log sistem.

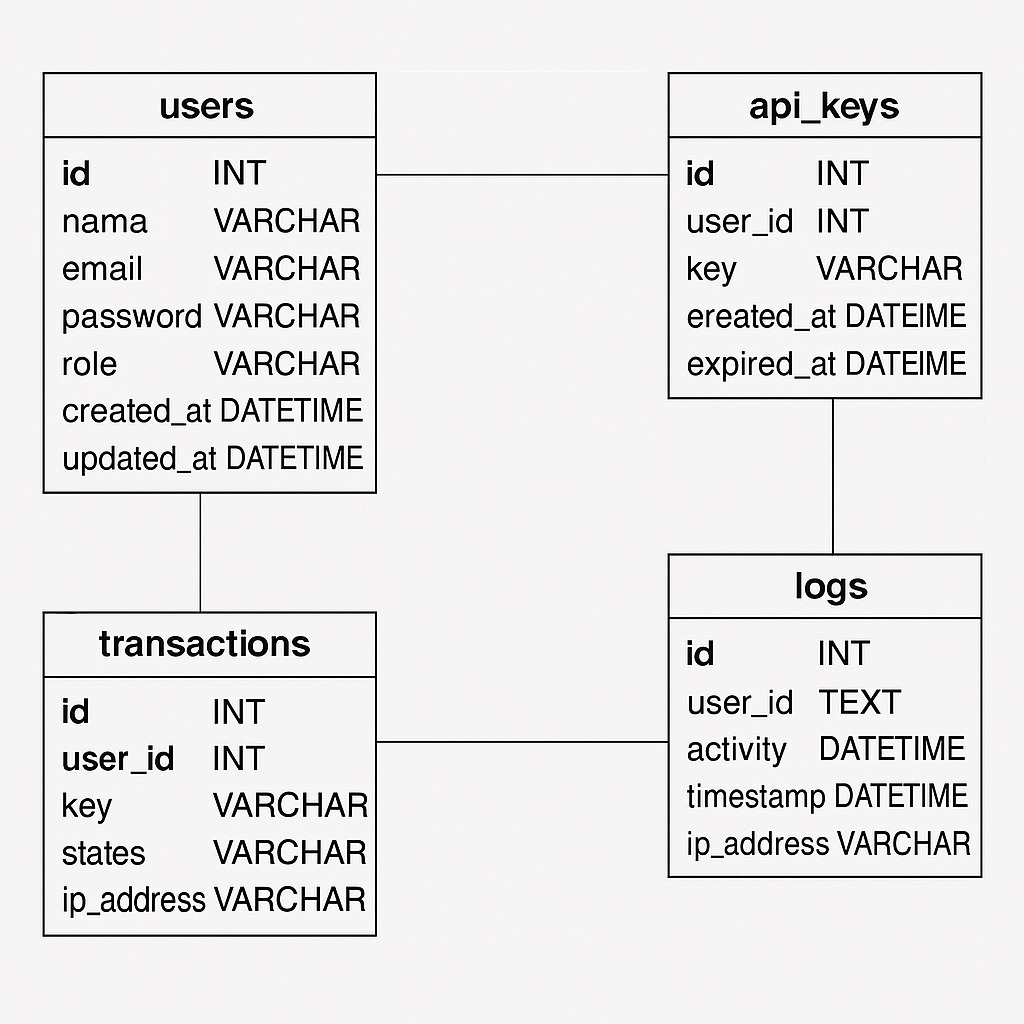
| Kolom | Tipe Data | Keterangan |
| --- | --- | --- |
| id | INT (PK) | ID unik log |
| user\_id | INT (FK) | ID pengguna yang melakukan aktivitas |
| activity | TEXT | Deskripsi aktivitas |
| timestamp | DATETIME | Waktu aktivitas dilakukan |
| ip\_address | VARCHAR | Alamat IP pengguna |

**4. api\_keys**

Digunakan untuk autentikasi akses API pihak ketiga.

| Kolom | Tipe Data | Keterangan |
| --- | --- | --- |
| id | INT (PK) | ID unik API key |
| user\_id | INT (FK) | ID pengguna yang memiliki API key |
| key | VARCHAR | Nilai API key |
| created\_at | DATETIME | Tanggal pembuatan API key |
| expired\_at | DATETIME | Tanggal kedaluwarsa API key |

**4.2 Diagram Entity-Relationship (ERD)**



**4.3 Skema Indexing dan Optimasi**

1. Indexing pada Kolom Pencarian Utama

* Deskripsi: Untuk meningkatkan performa query pencarian pada data presensi dan absensi, index akan diterapkan pada kolom-kolom yang sering digunakan dalam pencarian utama, seperti kolom ID Karyawan, Tanggal, dan Nama Karyawan.
* Tujuan: Indexing akan mempercepat proses pencarian dan penyaringan data, sehingga memperkecil waktu eksekusi query dan mengurangi beban pada server.
* Jenis Index yang Digunakan:
  + B-Tree Index untuk kolom yang mengandung data yang terurut atau sering digunakan dalam klausa WHERE, ORDER BY, atau JOIN.
  + Full-Text Index untuk kolom yang sering mengandung pencarian berbasis kata kunci, seperti Nama Karyawan.

2. Partisi Tabel Transaksi Berdasarkan Periode

* Deskripsi: Tabel transaksi absensi akan dipartisi berdasarkan periode waktu, seperti per bulan atau per tahun, guna meningkatkan performa pengambilan data yang lebih cepat pada rentang waktu tertentu.
* Tujuan: Dengan membagi tabel menjadi beberapa bagian berdasarkan periode, sistem dapat mengakses subset data yang lebih kecil, yang akan mempercepat waktu query dan mengurangi penggunaan sumber daya.
* Implementasi:
  + Partisi dapat dilakukan dengan menggunakan fitur partisi pada DBMS seperti MySQL atau PostgreSQL, yang membagi tabel besar menjadi beberapa bagian lebih kecil (partisi) berdasarkan kolom Tanggal atau Bulan.
  + Misalnya, data untuk setiap bulan akan disimpan dalam partisi terpisah seperti absensi\_2025\_01, absensi\_2025\_02, dan seterusnya.

3. Backup Otomatis Setiap 24 Jam

* Deskripsi: Sistem akan melakukan backup otomatis setiap 24 jam untuk memastikan ketersediaan data dan pemulihan sistem jika terjadi kegagalan atau kerusakan data.
* Tujuan: Mencegah kehilangan data yang penting akibat kegagalan sistem atau kesalahan pengguna. Backup reguler memastikan pemulihan data dengan cepat, mengurangi waktu downtime, dan menjaga keberlanjutan operasional sistem.
* Strategi Backup:
  + Incremental Backup: Hanya data yang berubah sejak backup terakhir yang akan dibackup, mengurangi ukuran backup dan waktu yang dibutuhkan.
  + Full Backup: Dilakukan satu kali dalam seminggu untuk menyimpan seluruh data sebagai cadangan utama.
  + Penyimpanan: Backup akan disimpan di server terpisah atau cloud storage untuk memastikan data aman dan dapat dipulihkan jika diperlukan.

**5. Antarmuka Pengguna**

**5.1 Wireframe Desain**

Wireframe adalah representasi visual dari antarmuka pengguna (UI) yang menggambarkan struktur dan elemen-elemen dasar dalam setiap halaman. Desain ini berfokus pada kemudahan navigasi dan pengoperasian bagi pengguna.

**1. Halaman Login**

* **Deskripsi**: Halaman login memungkinkan pengguna untuk mengakses sistem dengan memasukkan username dan password. Untuk meningkatkan keamanan, halaman ini juga akan menyediakan opsi autentikasi dua faktor (2FA).
* **Elemen**:
  + **Form Input**: Kolom untuk memasukkan username dan password.
  + **Tombol Login**: Tombol untuk mengirim data dan memverifikasi kredensial.
  + **Autentikasi Dua Faktor**: Setelah memasukkan username dan password yang benar, pengguna akan diminta untuk memasukkan kode verifikasi yang dikirim melalui email atau aplikasi autentikator.
  + **Tautan Lupa Password**: Untuk membantu pengguna yang lupa kata sandi mereka.
  + **Tautan Daftar Akun Baru**: Untuk pengguna yang belum memiliki akun.
* **Wireframe**:
  + Kolom Username di atas kolom Password.
  + Tombol Login berada di bawah kolom password.
  + Opsi 2FA (misalnya, kolom input kode verifikasi) akan muncul setelah login pertama kali.

**2. Dashboard**

* **Deskripsi**: Halaman dashboard menyediakan tampilan ringkasan data dan laporan terkait presensi dan absensi, dengan kemampuan filter untuk menyesuaikan data yang ditampilkan.
* **Elemen**:
  + **Ringkasan Data**: Menampilkan grafik dan statistik utama, seperti jumlah kehadiran, keterlambatan, atau absensi.
  + **Filter**: Dropdown atau pilihan tanggal untuk memfilter laporan, misalnya berdasarkan periode waktu tertentu (bulanan, tahunan).
  + **Tabel Laporan**: Tabel interaktif yang menampilkan data absensi atau presensi dengan opsi untuk mengurutkan berdasarkan kolom tertentu.
  + **Grafik Interaktif**: Bar chart, pie chart, atau line chart untuk visualisasi data.
  + **Navigasi**: Menu navigasi untuk mengakses halaman lain seperti CRUD atau pengaturan.
* **Wireframe**:
  + Header berisi nama aplikasi dan menu navigasi.
  + Di bawah header, ada area utama untuk menampilkan ringkasan data dengan grafik dan statistik.
  + Samping kanan atau bawah terdapat filter untuk memilih periode data.
  + Tabel dengan data yang dapat dipilih dan disortir.

**3. Halaman CRUD (Create, Read, Update, Delete)**

* **Deskripsi**: Halaman ini digunakan untuk mengelola data presensi dan absensi karyawan, dengan fitur input data dan pengelolaan data yang telah ada.
* **Elemen**:
  + **Form Input**: Kolom untuk memasukkan data baru, seperti ID Karyawan, Tanggal, Jam Masuk, Jam Keluar, Status Kehadiran, dan Keterangan.
  + **Tabel Data**: Menampilkan data absensi yang telah disimpan, dengan fitur pencarian, pengurutan berdasarkan kolom, dan pagination.
  + **Fungsi Edit dan Hapus**: Setiap baris dalam tabel memiliki opsi untuk mengedit atau menghapus data yang ada.
  + **Tombol Simpan**: Tombol untuk menyimpan data yang telah dimasukkan.
  + **Tombol Batal**: Tombol untuk membatalkan perubahan yang dilakukan.
* **Wireframe**:
  + Form input data di bagian atas halaman, dengan kolom input yang jelas dan tombol simpan di bawahnya.
  + Tabel di bawah form dengan kolom untuk ID Karyawan, Tanggal, Status Kehadiran, dan tombol aksi seperti Edit dan Hapus di setiap baris.
  + Pagination di bawah tabel untuk navigasi data lebih lanjut

**5.2 Desain Responsif**

Desain responsif bertujuan untuk memastikan aplikasi dapat diakses dan digunakan dengan nyaman di berbagai perangkat, mulai dari desktop, tablet, hingga smartphone. Berikut adalah pendekatan yang akan digunakan untuk mencapai desain responsif:

1. Menggunakan Bootstrap 5

* Deskripsi: Bootstrap 5 adalah framework CSS yang akan digunakan untuk membangun antarmuka pengguna yang responsif. Dengan memanfaatkan kelas dan komponen Bootstrap, aplikasi dapat menyesuaikan diri secara otomatis dengan berbagai ukuran layar dan perangkat.
* Keuntungan:
  + Kompatibilitas Perangkat: Bootstrap 5 menyediakan sistem grid fleksibel dan komponen UI yang responsif yang memastikan aplikasi bekerja dengan baik pada perangkat apapun.
  + Desain Mobile-First: Secara default, Bootstrap 5 mendukung desain mobile-first, yang berarti antarmuka akan dioptimalkan terlebih dahulu untuk perangkat seluler, dan kemudian diatur untuk desktop.
  + Komponen Responsif: Komponen seperti navbar, card, form, dan tabel dapat dengan mudah disesuaikan agar tampil optimal di berbagai perangkat tanpa perlu penyesuaian manual yang rumit.

2. Layout Fleksibel dengan Grid System

* Deskripsi: Bootstrap 5 menggunakan grid system berbasis 12 kolom untuk membuat tata letak yang fleksibel dan dinamis. Grid system memungkinkan kita untuk merancang tampilan yang sesuai dengan perangkat, dengan mengatur lebar dan posisi elemen berdasarkan ukuran layar.
* Keuntungan:
  + Penataan yang Mudah: Dengan menggunakan grid system, elemen-elemen pada halaman (seperti form, tabel, dan dashboard) akan disusun secara otomatis sesuai dengan lebar layar perangkat.
  + Kolom Responsif: Kolom-kolom pada halaman dapat disesuaikan dengan ukuran layar. Misalnya, pada perangkat desktop, tabel dan form akan terlihat dalam satu baris dengan beberapa kolom, tetapi pada perangkat mobile, kolom-kolom tersebut akan disusun vertikal agar lebih mudah dibaca.
  + Custom Breakpoints: Dengan menggunakan kelas seperti col-md-6, col-lg-4, dan col-12, kita dapat menentukan bagaimana elemen-elemen akan menyesuaikan ukuran dan posisi sesuai dengan perangkat (misalnya, lebih kecil pada mobile, lebih besar pada desktop).

3. Penggunaan AJAX untuk Mempercepat Interaksi Pengguna

* Deskripsi: AJAX (Asynchronous JavaScript and XML) akan digunakan untuk memungkinkan interaksi pengguna dengan aplikasi tanpa perlu melakukan refresh atau reload halaman secara keseluruhan. Hal ini akan memberikan pengalaman pengguna yang lebih responsif dan mulus.
* Keuntungan:
  + Tanpa Reload Halaman: Dengan menggunakan AJAX, pengguna dapat melakukan aksi seperti menambahkan, mengedit, atau menghapus data pada halaman tanpa perlu menunggu halaman sepenuhnya dimuat ulang. Hanya bagian tertentu dari halaman yang akan diperbarui.
  + Pengalaman Pengguna yang Lebih Cepat: Interaksi dengan data (misalnya, saat mengubah status absensi atau mencari data) akan lebih cepat karena hanya sebagian data yang diproses dan ditampilkan.
  + Penghematan Bandwidth: Karena hanya data yang dibutuhkan yang dikirim dan diterima, penggunaan bandwidth menjadi lebih efisien, yang juga meningkatkan kinerja aplikasi, terutama pada koneksi internet yang lebih lambat.

**6. Pertimbangan Keamanan**

Keamanan sistem adalah prioritas utama dalam pengembangan aplikasi, terutama ketika berhubungan dengan data sensitif dan interaksi pengguna. Berikut adalah beberapa pertimbangan keamanan yang akan diterapkan untuk memastikan perlindungan terhadap data dan mencegah potensi ancaman keamanan:

1. Enkripsi Data

* Deskripsi: Untuk menjaga kerahasiaan dan integritas data sensitif (seperti kata sandi, informasi pribadi, dan data absensi), data akan dienkripsi sebelum disimpan di database.
* Teknologi:
  + AES-256 (Advanced Encryption Standard) akan digunakan untuk enkripsi data. AES-256 adalah algoritma enkripsi simetris yang sangat kuat dan banyak digunakan dalam industri untuk melindungi data sensitif.
* Penerapan:
  + Data sensitif seperti password, informasi kartu kredit, atau data absensi yang memerlukan perlindungan khusus akan dienkripsi sebelum disimpan di database.
  + Kunci enkripsi akan disimpan dengan aman menggunakan mekanisme penyimpanan yang terpisah, seperti Vault atau Hardware Security Modules (HSM).

2. Kontrol Akses (RBAC - Role-Based Access Control)

* Deskripsi: Implementasi Role-Based Access Control (RBAC) bertujuan untuk membatasi akses ke fitur atau data tertentu hanya kepada pengguna dengan peran yang sesuai. Setiap pengguna akan diberikan peran tertentu yang menentukan tingkat akses yang mereka miliki.
* Tujuan: Memastikan bahwa hanya pengguna yang memiliki otorisasi yang sesuai yang dapat mengakses informasi atau melakukan tindakan tertentu.
* Penerapan:
  + Pengguna akan memiliki peran seperti Admin, Manager, Karyawan, dan lainnya.
  + Admin memiliki akses penuh ke semua data dan fitur aplikasi.
  + Manager hanya memiliki akses terbatas pada data tertentu, seperti laporan absensi, dan tidak dapat mengelola akun pengguna.
  + Karyawan hanya dapat melihat dan memperbarui data absensi mereka sendiri.

3. Proteksi API

* Deskripsi: Aplikasi akan menggunakan API untuk komunikasi antara frontend dan backend. Untuk mencegah akses tidak sah dan potensi penyalahgunaan, API akan dilindungi menggunakan API Key dan Token Autentikasi.
* Tujuan: Melindungi API dari akses yang tidak sah dan memastikan bahwa hanya pengguna yang terautentikasi yang dapat mengakses endpoint API.
* Penerapan:
  + API Key: Setiap aplikasi atau pengguna yang mengakses API akan diberikan API Key yang digunakan untuk mengidentifikasi dan memverifikasi aplikasi tersebut.
  + Token Autentikasi: Pengguna akan diminta untuk menggunakan JWT (JSON Web Tokens) setelah login untuk mengakses API. Token ini akan memiliki waktu kedaluwarsa dan hanya dapat digunakan dalam jangka waktu yang ditentukan.
  + Token akan dienkripsi dan disimpan di sisi klien (misalnya, dalam cookies yang aman atau local storage).

4. Audit Log

* Deskripsi: Semua aktivitas pengguna, baik itu login, pembuatan, pembaruan, atau penghapusan data, akan dicatat dalam Audit Log untuk tujuan keamanan dan kepatuhan.
* Tujuan: Meningkatkan transparansi dan memungkinkan pelacakan aktivitas mencurigakan atau kesalahan yang terjadi di dalam sistem.
* Penerapan:
  + Setiap aksi pengguna akan dicatat bersama dengan informasi terkait seperti ID pengguna, timestamp, dan jenis aksi yang dilakukan.
  + Audit log akan disimpan dalam format yang aman dan hanya dapat diakses oleh admin atau pihak yang berwenang.
  + Log dapat digunakan untuk melacak kejadian keamanan dan melakukan analisis jika ada pelanggaran atau insiden.

5. Firewall Aplikasi Web (WAF)

* Deskripsi: Web Application Firewall (WAF) akan digunakan untuk melindungi aplikasi dari berbagai jenis serangan web, termasuk SQL Injection dan Cross-Site Scripting (XSS).
* Tujuan: Menghalangi serangan yang berpotensi merusak atau mengekspos data, serta memblokir akses tidak sah ke aplikasi.
* Penerapan:
  + SQL Injection: WAF akan memblokir permintaan yang mencurigakan, seperti input yang mengandung karakter atau pola yang dapat digunakan untuk menyusupkan kode SQL berbahaya.
  + XSS (Cross-Site Scripting): WAF juga akan memeriksa input pengguna untuk memastikan bahwa skrip jahat tidak dapat dijalankan di sisi klien, melindungi aplikasi dari serangan XSS.
  + Penerapan Keamanan Lainnya: WAF dapat diterapkan bersama dengan mekanisme keamanan lainnya, seperti Rate Limiting untuk membatasi jumlah permintaan dari satu IP dalam jangka waktu tertentu, serta IP Whitelisting untuk membatasi akses hanya kepada IP yang terotorisasi.

Rangkuman Keamanan:

1. Enkripsi Data: AES-256 untuk melindungi data sensitif.
2. Kontrol Akses (RBAC): Menggunakan peran untuk membatasi hak akses pengguna.
3. Proteksi API: API Key dan token autentikasi untuk melindungi akses ke API.
4. Audit Log: Mencatat semua aktivitas pengguna untuk meningkatkan transparansi dan keamanan.
5. WAF: Mencegah serangan web seperti SQL Injection dan XSS dengan menggunakan Web Application Firewall.