



## **LAPORAN UJI KOMPETENSI**

SKEMA : PEMROGRAMAN MOBILE PRATAMA (JUNIOR MOBILE COMPUTING)

Nama Asesi : Muhammad Heriyanto

NIM : E31221142

Nama Asesor : Hermawan Arief P.

MET : 000.004160.2018

**POLITEKNIK NEGERI JEMBER**  
**KEMENTERIAN PENDIDIKAN TINGGI, SAINS DAN**  
**TEKNOLOGI 2025**

## **Aplikasi EPresensi**

Epresensi adalah aplikasi berbasis mobile yang dibuat dengan tujuan sebagai syarat untuk melakukan Uji Kompetensi pada Skema Pemograman *Mobile Pratama (Junior Mobile Computing)*. Aplikasi dibentuk dalam membantu proses pencatatan kehadiran dari siswa secara *digital* dengan menggunakan *software* atau aplikasi *mobile* yang aman dan terintegrasi.

### **Tools Perangkat Lunak Untuk Sistem Informasi Akademik**

#### **1. Sistem Operasi**

- Windows 11 (sebagai OS utama dalam pembuatan aplikasi)

#### **2. Backend & Database**

- Express js (Node js framework untuk REST API pada backend)
- MySQL (Database untuk penyimpanan data user dan absensi)
- Swagger (Dokumentasi dan pengujian API)

#### **3. Mobile Framework & Development**

- React Native (Expo) – Framework utama untuk pembuatan aplikasi mobile lintas platform

#### **4. Frontend Tools (Mobile UI & Libraries)**

- React Navigation – Untuk navigasi antar halaman
- Axios – Untuk komunikasi API dari frontend ke backend
- Expo Location / Maps – Untuk fitur GPS dan penentuan lokasi absensi
- AsyncStorage / SecureStore – Untuk menyimpan token JWT di perangkat

#### **5. Code Editor / IDE**

- Visual Studio Code – Editor kode utama dengan ekstensi React Native & JavaScript

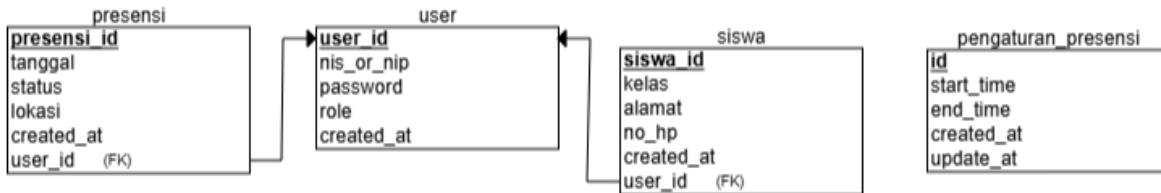
#### **6. Browser & Testing Tools**

- Google Chrome + Swagger UI – Untuk testing API
- Expo Go / Android Emulator – Untuk menjalankan dan menguji aplikasi mobile

#### **7. Authentication & Security**

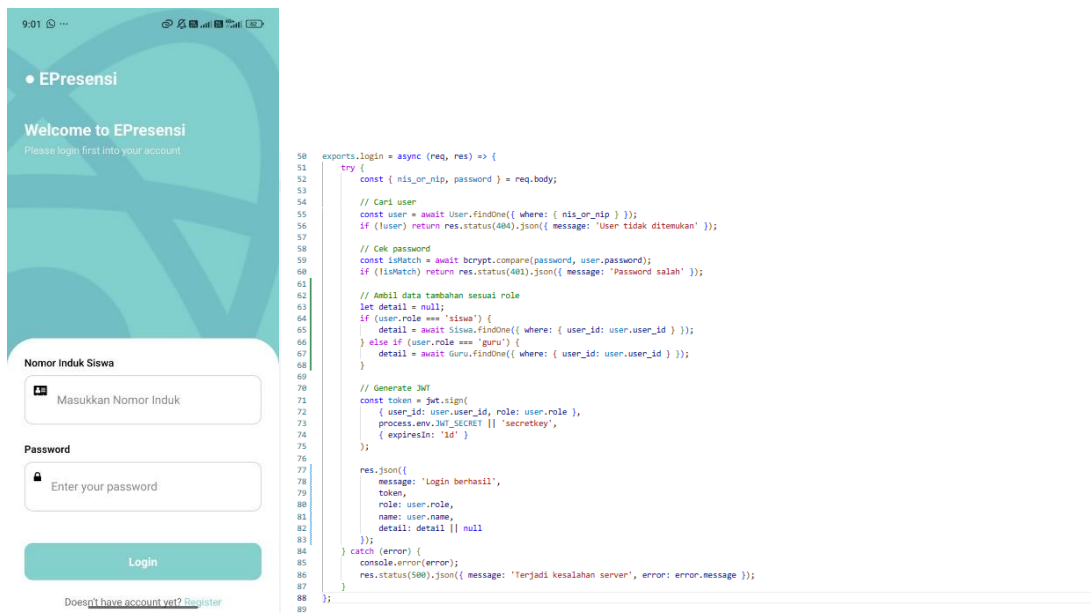
- JWT (JSON Web Token) – Untuk autentikasi dan otorisasi pengguna
- CORS – Untuk mengatur akses antara frontend dan backend
- bcrypt.js – Untuk enkripsi password pengguna di backend

## Entity Schema Diagram



## Fitur Sistem Pendukung

### 1. LOGIN PAGE

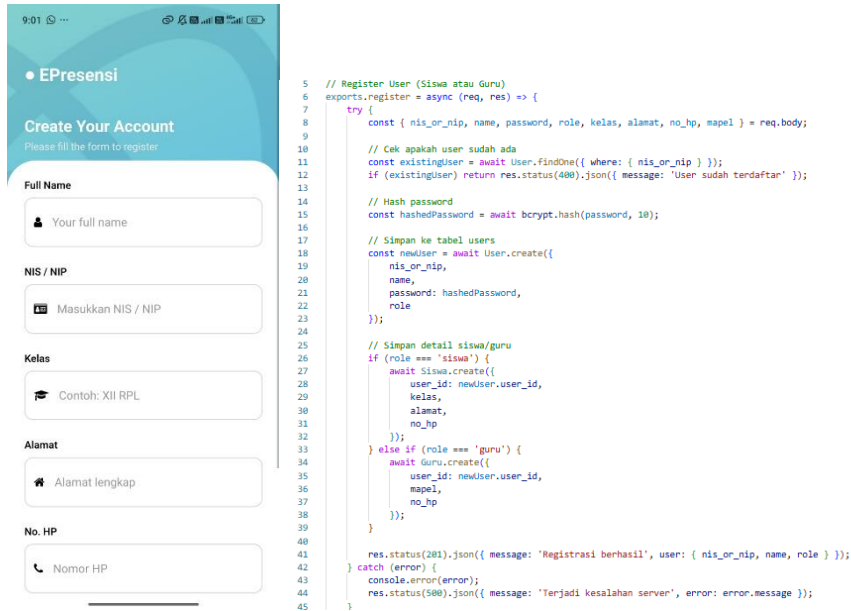


Pada halaman login tersebut dijalankan melalui *framework* atau tools Bernama expo go yang dimana terdiri dari beberapa komponen *react js* yang diantaranya meliputi view , text , TextInput untuk struktur dasar dari *User Interface* nya serta *Keyboarding Avoiding View* dan *Touchable Without Feedback* untuk menangani interaksi keyboard. Terdapat juga *ImageBackground* untuk latar belakang visual dari halaman login.

Sistem navigasi menggunakan Expo Router untuk berpindah halaman , sedangkan *AsyncStorage* digunakan untuk menyimpan data pengguna secara *local* setelah login berhasil. Untuk integrasi dengan API pada *backend* menggunakan *Axios* sebagai HTTP client untuk mengirim request login ke API.

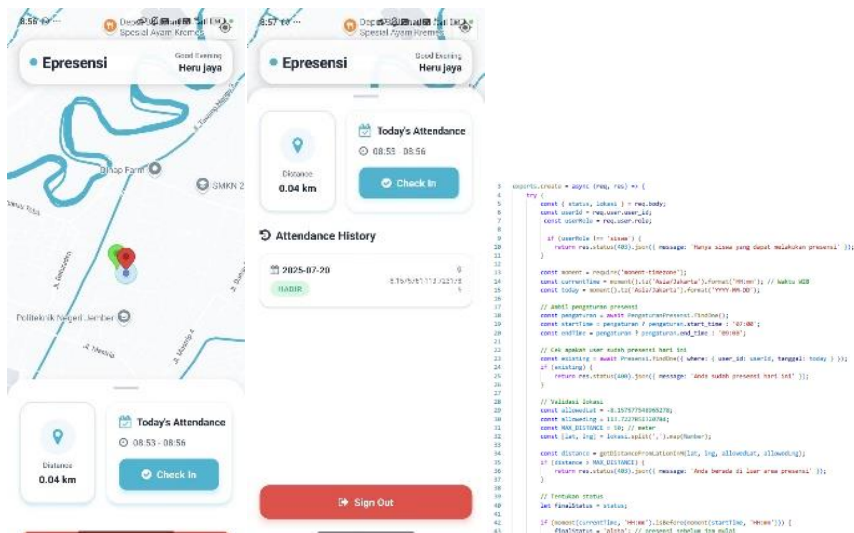
Fungsi utama meliputi *state management* dilengkapi dengan *use state* untuk mengelola inputan email/password dan status keyboard yang memproses autentikasi , validasi role, penyimpanan token , dan navigasi ke halaman selanjutnya.

## 2. REGISTER PAGE



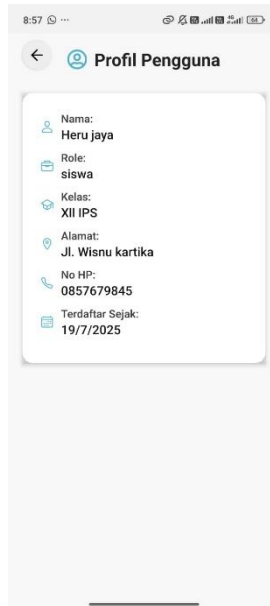
Pada halaman register ini menggunakan *React Native* untuk komponen dasar UI nya yang dijalankan melalui expo go. Komponen dasar yang ada pada halaman ini meliputi React Native Elements untuk button styling, Vector Icons untuk ikon input, Expo Linear Gradient untuk background gradien, Expo Router untuk navigasi antar halaman, dan Axios untuk mengirim data registrasi ke server API.

## 3. BERANDA



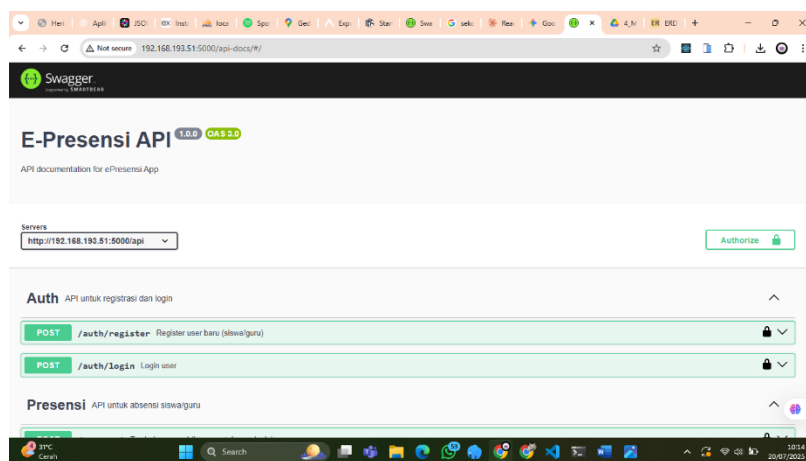
Pada halaman beranda ini menggunakan React-Native maps yang dimana library ini untuk menampilkan peta pada halaman beranda yang dilengkapi dengan circle radius untuk area absensi. Expo location digunakan untuk tracking Lokasi real-time dari user untuk fetch data presensi , kirim presensi dan ambil pengaturan waktu.

#### 4. PROFILE



Pada halaman profile ini untuk menampilkan data profile user yang telah melakukan login yang dimana pada pembuatan UI nya menggunakan beberapa komponen dari *React-Native* yakni *AsyncStorage* untuk mengambil data user yang telah login.

#### 5. API DOKUMENTASI



Pada gambar tersebut menampilkan Dokumentasi API yang diakses dengan menggunakan swagger. Pada swagger tidak hanya untuk melihat dari API yang telah dibuat tapi juga digunakan untuk melakukan terlebih dahulu sebelum di integrasikan dengan frontend.