

LAPORAN PROJECT

GeoProof (Geolocation Camera Attendance System)

MATA KULIAH

PENGEMBANGAN APLIKASI WEB



KELOMPOK 2 KELAS E:

M. Rafly Husen Rafsanjani (20230140205)

Irfansyah Ridho Aninda (20230140223)

Aswin Lutfian Prasetyo (20230140244)

Galih Maulana Syawalqi (20230140248)

PROGRAM STUDI TEKNIK INFORMATIKA

UNIVERSITAS MUHAMMADIYAH YOGYAKARTA

2025

DAFTAR ISI

DAFTAR ISI	2
1. Ringkasan Proyek	5
1.1 Nama Proyek dan Tema	5
1.2 Deskripsi Singkat.....	5
1.3 Target Pengguna	5
2. Latar Belakang dan Masalah	5
2.1 Latar belakang.....	5
2.2 Permasalahan yang ingin diselesaikan.....	6
3. Tujuan dan Manfaat.....	6
3.1 Tujuan Sistem	6
3.2 Manfaat bagi Mahasiswa.....	7
3.3 Manfaat bagi Admin.....	7
4. Ruang Lingkup	7
4.1 Ruang lingkup fungsional	7
4.2 Batasan	8
4.3 Asumsi.....	8
5. Fitur Utama	8
5.1 Fitur untuk Mahasiswa (User).....	8
5.2 Fitur Admin	9
5.3 Aturan Validasi Inti (Backend Rules)	10
5.4 Aktor Sistem.....	11
5.5 Skenario Utama yang Direkomendasikan untuk Diagram.....	11
6. Aturan Geofence dan Validasi Lokasi.....	11
6.1 Definisi Circular Geofence	11
6.2 Validasi di Backend.....	11
6.3 Kebijakan Akurasi Lokasi	11
6.4 Data yang disimpan untuk Bukti dan Audit	12

7. Anti-Spoofing dan Kontrol Keamanan	12
7.1 Defense-in-depth	12
7.2 Bukti foto (selfie) sebagai verifikasi tambahan	13
7.3 Catatan keterbatasan web terhadap fake GPS	14
8. Arsitektur Sistem	14
8.1 3-Tier Architecture.....	14
8.2 Komponen pendukung.....	15
8.3 Alur data ringkas	15
9. Teknologi yang Digunakan	16
9.1 Frontend: React, Browser API (Geolocation, Camera)	16
9.2 Map: Leaflet + OpenStreetMap	17
9.3 Backend: Node.js + Express (REST).....	18
9.4 Security: JWT, bcrypt	19
9.5 Upload: multer	20
9.6 Database: MySQL	21
10. Deployment Plan	22
10.1 Domain/Subdomain.....	22
10.2 Environment	23
10.3 Catatan konfigurasi umum.....	24
11. Rencana Artefak UML	26
11.1 Use Case Diagram.....	26
11.2 Flowchart.....	27
11.3 Activity Diagram.....	28
11.4 Sequence Diagram	29
11.5 Class Diagram	30
11.6 ERD.....	31
11.7 Arsitektur Diagram	32
12. Penutup	32
12.1 Ringkasan keputusan desain	32

12.2 Next step pengerjaan	33
13. Documentation	35
13.1 Tampilan Web per Homepage	35

1. Ringkasan Proyek

1.1 Nama Proyek dan Tema

Nama Proyek: GeoProof

Tema: Geolocation Camera (IoT berbasis smartphone sebagai perangkat sensor)

1.2 Deskripsi Singkat

GeoProof adalah aplikasi web presensi kegiatan/KKN untuk mahasiswa yang memanfaatkan **kamera** dan **lokasi GPS** dari smartphone sebagai bukti kehadiran. Sistem menerapkan **circular geofence** (titik pusat dan radius) sehingga presensi hanya dapat dilakukan apabila mahasiswa berada di area yang ditentukan (misalnya area kampus untuk uji coba). Setiap presensi menghasilkan bukti berupa foto (selfie) serta data lokasi (latitude, longitude, akurasi, dan waktu server) yang tersimpan dan dapat direkap oleh admin.

1.3 Target Pengguna

1. **Mahasiswa (User):** melakukan presensi kegiatan/KKN dengan selfie dan lokasi pada area yang diizinkan.
2. **Admin (Panitia/Dosen Pembimbing/Asisten):** mengelola lokasi presensi (geofence), memantau presensi yang masuk, melakukan verifikasi bila diperlukan, serta mengekspor rekap presensi.

2. Latar Belakang dan Masalah

2.1 Latar belakang

Kegiatan akademik dan non-akademik mahasiswa, termasuk kegiatan organisasi, kepanitiaan, praktikum lapangan, serta program Kuliah Kerja Nyata (KKN), umumnya memerlukan mekanisme presensi untuk memastikan partisipasi dan keterlibatan peserta. Pada praktiknya, presensi sering dilakukan menggunakan metode manual seperti tanda tangan pada kertas, input nama pada daftar hadir, atau konfirmasi melalui pesan singkat. Metode tersebut relatif mudah diterapkan, namun memiliki beberapa kelemahan, seperti proses rekap yang memakan waktu, potensi kesalahan pencatatan, dan keterbatasan bukti kehadiran yang sulit diverifikasi.

Seiring meningkatnya penggunaan smartphone oleh mahasiswa, peluang pemanfaatan perangkat tersebut sebagai sumber data sensor juga semakin relevan. Smartphone secara umum telah memiliki kamera serta layanan lokasi (GPS) yang mampu menangkap data dunia nyata dan mengirimkannya melalui jaringan ke sistem informasi. Dalam konteks tema *Geolocation Camera*, kombinasi kamera dan lokasi dapat dimanfaatkan sebagai bukti presensi yang lebih kuat dibanding sekadar input teks, karena presensi tidak hanya mencatat “siapa” yang hadir, tetapi juga “di mana” presensi dilakukan serta “kapan” presensi dilakukan.

Berdasarkan kebutuhan tersebut, GeoProof dirancang sebagai aplikasi web presensi kegiatan/KKN yang memanfaatkan foto selfie sebagai bukti visual dan koordinat GPS sebagai bukti lokasi. Untuk meningkatkan validitas, GeoProof menerapkan *circular geofence* sehingga presensi hanya dapat dilakukan pada area yang telah ditentukan (misalnya area universitas sebagai lokasi uji coba awal). Dengan pendekatan ini, sistem diharapkan dapat membantu proses presensi menjadi lebih tertib, terdokumentasi, serta mudah direkap.

2.2 Permasalahan yang ingin diselesaikan

Berdasarkan kondisi yang umum terjadi pada presensi kegiatan mahasiswa, permasalahan yang ingin ditangani oleh GeoProof meliputi:

1. **Validitas lokasi presensi sulit dipastikan** pada metode presensi manual atau berbasis input teks, karena tidak ada pembuktian lokasi yang terukur.
2. **Potensi presensi tidak sesuai kondisi lapangan** (misalnya titip absen atau presensi dari luar area kegiatan) lebih sulit terdeteksi tanpa kontrol berbasis lokasi.
3. **Rekap presensi memerlukan waktu dan rentan kesalahan**, terutama jika presensi dilakukan secara manual dan harus dipindahkan kembali ke format digital.
4. **Ketersediaan bukti kehadiran yang lemah**, karena catatan presensi tradisional umumnya tidak menyertakan bukti visual dan metadata waktu yang terstandar.
5. **Kebutuhan sistem yang mudah diakses** oleh mahasiswa tanpa instalasi aplikasi khusus; web menjadi pilihan karena dapat diakses lintas perangkat selama memiliki browser, kamera, dan layanan lokasi.

Untuk menjawab permasalahan tersebut, GeoProof menerapkan pendekatan presensi berbasis bukti (foto) dan validasi lokasi melalui geofence. Validasi geofence dilakukan pada sisi backend untuk menjaga integritas proses, disertai penyimpanan data audit seperti akurasi GPS, jarak terhadap pusat geofence, serta waktu dari server.

3. Tujuan dan Manfaat

3.1 Tujuan Sistem

Tujuan utama dari pengembangan GeoProof adalah:

1. **Menyediakan sistem presensi berbasis web** untuk kegiatan mahasiswa/KKN yang mudah diakses tanpa instalasi aplikasi khusus.
2. **Meningkatkan validitas presensi** dengan memanfaatkan kombinasi bukti foto (selfie) dan bukti lokasi (koordinat GPS) pada saat presensi dilakukan.
3. **Menerapkan pembatasan area presensi (geofence)** sehingga presensi hanya dapat dilakukan pada wilayah yang telah ditentukan (misalnya area universitas sebagai lokasi uji coba awal).

4. **Mendukung proses pencatatan dan rekap presensi yang lebih efisien** melalui penyimpanan data presensi secara terstruktur di database.
5. **Menyediakan jejak audit dasar** (lokasi, akurasi GPS, jarak ke titik geofence, dan waktu server) untuk kebutuhan monitoring dan evaluasi presensi.

3.2 Manfaat bagi Mahasiswa

GeoProof memberikan manfaat bagi mahasiswa, antara lain:

1. **Proses presensi lebih praktis dan cepat** karena dapat dilakukan melalui perangkat yang dimiliki (smartphone) selama memiliki kamera, layanan lokasi, dan internet.
2. **Bukti presensi lebih jelas** karena sistem menyimpan foto dan metadata lokasi serta waktu pada saat presensi dilakukan.
3. **Transparansi riwayat presensi** melalui fitur riwayat/status presensi, sehingga mahasiswa dapat memantau presensi yang sudah tercatat.
4. **Meminimalkan kesalahan input manual**, karena data penting seperti lokasi dan waktu dicatat otomatis oleh sistem.

3.3 Manfaat bagi Admin

GeoProof memberikan manfaat bagi admin/pengelola kegiatan, antara lain:

1. **Kontrol area presensi yang lebih terukur** melalui pengaturan geofence (titik pusat dan radius) untuk setiap lokasi presensi.
2. **Rekap presensi lebih mudah dan terstruktur**, sehingga proses monitoring kehadiran dapat dilakukan lebih cepat dibanding metode manual.
3. **Ketersediaan bukti dan audit trail**, termasuk foto, koordinat, jarak terhadap geofence, akurasi GPS, dan waktu server, sehingga verifikasi dapat dilakukan bila diperlukan.
4. **Mendukung pelaporan kegiatan**, misalnya melalui tampilan daftar presensi dan opsi ekspor rekap (jika diterapkan), untuk keperluan dokumentasi dan evaluasi.

4. Ruang Lingkup

4.1 Ruang lingkup fungsional

Sistem GeoProof mencakup fungsi utama berikut:

1. Autentikasi pengguna (login) untuk mahasiswa dan admin.
2. Presensi mahasiswa dengan:
 - a. pengambilan foto selfie melalui browser,
 - b. pengambilan lokasi GPS melalui browser,
 - c. validasi geofence oleh backend,
 - d. penyimpanan bukti presensi (foto + lokasi + waktu).
3. Riwayat presensi mahasiswa (status, tanggal, dan detail bukti).

4. Manajemen geofence oleh admin (titik pusat dan radius, serta pengaktifan lokasi presensi).
5. Rekap presensi oleh admin (daftar presensi, filter sederhana, dan ekspor data jika diperlukan).

4.2 Batasan

Agar proyek tetap fokus dan realistis untuk pembelajaran, batasan awal sistem adalah:

1. Sistem belum menggunakan perangkat IoT tambahan (ESP32/GPS module terpisah). Data sensor berasal dari smartphone pengguna.
2. Sistem tidak menjamin pencegahan spoofing lokasi secara sempurna (karena keterbatasan web browser), namun menerapkan kontrol mitigasi yang wajar.
3. Verifikasi biometrik (face recognition) tidak termasuk scope. Foto digunakan sebagai bukti visual, bukan identifikasi otomatis.
4. Mode offline tidak didukung; presensi membutuhkan koneksi internet.

4.3 Asumsi

1. Mahasiswa menggunakan perangkat yang memiliki kamera dan layanan lokasi aktif.
2. Pengguna memberikan izin akses kamera dan lokasi pada browser.
3. Area uji coba awal berada di wilayah universitas (kampus) untuk memudahkan pengujian dan validasi.
4. Server memiliki akses penyimpanan untuk file foto dan database MySQL untuk data presensi.

5. Fitur Utama

5.1 Fitur untuk Mahasiswa (User)

A. Autentikasi dan Akses Akun

1. Login Mahasiswa

Mahasiswa melakukan login untuk mengakses fitur presensi dan riwayat presensi.

2. Logout

Mahasiswa keluar dari sistem untuk mengakhiri sesi akses.

B. Presensi Kegiatan/KKN Berbasis Geolocation Camera

1. Melakukan Presensi (Submit Presensi)

Mahasiswa melakukan presensi dengan alur inti berikut:

- Mengaktifkan kamera dan mengambil **foto selfie** melalui browser.
- Mengambil **lokasi GPS** (latitude, longitude, accuracy) melalui browser.

- Mengirim data presensi ke backend untuk diproses dan divalidasi.

2. **Validasi Form Presensi di Sisi Client (Basic Validation)**

Validasi minimal sebelum submit, misalnya:

- Foto wajib ada.
- Lokasi wajib terbaca (tidak null).
- Koneksi internet tersedia (jika dapat dideteksi).

Catatan: validasi ini bersifat kenyamanan pengguna; keputusan akhir tetap di backend.

C. **Riwayat dan Status Presensi**

1. **Melihat Riwayat Presensi**

Mahasiswa dapat melihat daftar presensi yang pernah dilakukan beserta informasi ringkas (tanggal, kegiatan, status).

2. **Melihat Detail Presensi**

Mahasiswa dapat membuka detail presensi tertentu, misalnya:

- Foto bukti
- Koordinat dan (opsional) marker pada peta
- Waktu presensi (timestamp server)
- Status validasi dan catatan (jika ada)

5.2 **Fitur Admin**

1. **Autentikasi dan Manajemen Akses**

- a. **Login Admin**
- b. **Logout**

2. **Manajemen Lokasi Presensi (Geofence)**

c. **Membuat Lokasi Presensi (Create Geofence Lokasi)**

Admin menentukan area presensi menggunakan model circular:

- Nama lokasi (mis. “Kampus UMY – Gerbang Utama”)
- Titik pusat (latitude, longitude)
- Radius (meter)
- Status aktif/tidak aktif

d. **Memperbarui Lokasi Presensi (Update Geofence Lokasi)**

Admin dapat mengubah titik pusat/radius/nama/status.

e. **Menghapus Lokasi Presensi (Delete Geofence Lokasi)**

Admin menghapus lokasi yang tidak dipakai (opsional, tergantung kebijakan data).

f. **Melihat Daftar Lokasi Presensi (List Geofence Lokasi)**

Admin dapat melihat semua lokasi yang tersedia dan statusnya.

3. **Monitoring Presensi**

g. **Melihat Daftar Presensi Masuk**

Admin dapat melihat presensi yang terkumpul dengan filter sederhana (tanggal, kegiatan/lokasi, mahasiswa).

h. **Melihat Detail Presensi Mahasiswa**

Admin dapat melihat bukti dan metadata:

- Foto
- Koordinat, accuracy
- Jarak terhadap pusat geofence (hasil Haversine)
- Hasil validasi (inside/outside) dan alasan penolakan bila invalid

4. **Verifikasi**

i. **Verifikasi Presensi (Approve/Reject)**

Admin dapat menyetujui atau menolak presensi dengan alasan (mis. foto tidak sesuai, lokasi tidak valid, dsb.).

Catatan: Jika Anda ingin sistem sederhana, modul verifikasi bisa dibuat minimal (misalnya hanya “valid/invalid otomatis”). Namun untuk UML, fitur ini sangat membantu agar alur admin lebih kuat.

5. **Rekap dan Pelaporan**

j. **Rekap Presensi dan Ekspor Data (Opsional)**

Admin dapat mengekspor rekap (mis. CSV) untuk kebutuhan dokumentasi.

5.3 Aturan Validasi Inti (Backend Rules)

Bagian ini penting karena akan menjadi “decision/branch” pada flowchart dan activity diagram, serta menjadi inti sequence diagram.

Saat backend menerima request presensi, sistem melakukan:

1. **Validasi autentikasi** (JWT harus valid).
2. **Validasi data input minimal:**
 - foto ada dan format valid
 - lat/lng terbaca
3. **Perhitungan jarak dengan Haversine**
 - hitung distance_m antara titik user dan pusat geofence aktif
4. **Validasi geofence**
 - presensi diterima jika distance_m \leq radius_m
5. **Validasi kualitas lokasi (accuracy threshold)**

- jika accuracy terlalu buruk, presensi dapat ditolak/dianggap tidak valid (aturan final ditetapkan saat implementasi uji coba)

6. Simpan bukti dan metadata

- simpan foto (storage)
- simpan data presensi (DB) termasuk timestamp_server, distance_m, accuracy_m, dan status validasi

5.4 Aktor Sistem

1. **Mahasiswa (User)**
2. **Admin (Panitia/Dosen Pembimbing/Asisten)**
3. **Sistem (Web App + API Server + Database)** sebagai komponen internal pada sequence/activity.

5.5 Skenario Utama yang Direkomendasikan untuk Diagram

Agar nanti Anda mudah menggambar di draw.io, gunakan minimal 2 skenario utama berikut:

1. **Skenario 1: Mahasiswa melakukan presensi (valid) di dalam geofence**
(Capture selfie + ambil lokasi + backend validasi + simpan + tampil hasil)
2. **Skenario 2: Mahasiswa melakukan presensi tetapi di luar geofence (invalid)**
(Backend hitung jarak > radius, simpan status invalid / tolak, tampilkan pesan)
3. **Skenario 3: Admin memverifikasi presensi (approve/reject)**

6. Aturan Geofence dan Validasi Lokasi

6.1 Definisi Circular Geofence

Geofence pada GeoProof menggunakan model **circular**, didefinisikan oleh:

1. **Center point:** koordinat pusat (latitude, longitude)
2. **Radius:** jarak dalam meter dari titik pusat
Mahasiswa dianggap valid untuk presensi jika posisinya berada pada radius yang diizinkan dari titik pusat lokasi presensi.

6.2 Validasi di Backend

Validasi geofence dilakukan di **backend** (Node.js + Express) untuk menjaga integritas proses. Backend menerima koordinat dari client, lalu menghitung jarak antara posisi mahasiswa dan titik pusat geofence menggunakan **Haversine formula**. Hasil jarak tersebut dibandingkan dengan radius geofence untuk menentukan status “di dalam area” atau “di luar area”.

6.3 Kebijakan Akurasi Lokasi

Karena GPS memiliki variasi akurasi, sistem menerapkan kebijakan minimal:

1. Backend menyimpan nilai **accuracy** dari browser (dalam meter) sebagai data audit.
2. Jika accuracy berada di atas ambang batas tertentu (misalnya terlalu besar), sistem dapat meminta pengguna mencoba ulang agar presensi lebih valid.

Catatan: nilai ambang batas akan ditentukan pada tahap implementasi dan pengujian.

6.4 Data yang disimpan untuk Bukti dan Audit

Setiap presensi menyimpan informasi inti berikut:

1. Identitas user (mahasiswa)
2. Foto selfie (file/URL)
3. Latitude & longitude
4. Accuracy (meter)
5. Timestamp dari server
6. Hasil perhitungan jarak (distance_m) dan status validasi geofence (inside/outside)

Informasi ini mendukung transparansi dan rekap presensi, serta memudahkan pengecekan bila terjadi perbedaan data.

7. Anti-Spoofing dan Kontrol Keamanan

7.1 Defense-in-depth

GeoProof menerapkan pendekatan **defense-in-depth** dengan beberapa lapisan keamanan untuk meminimalkan risiko manipulasi data presensi:

A. Accuracy Gate (Gerbang Akurasi)

Sistem menerapkan batas minimum akurasi GPS yang diterima:

// Backend validation

`const MAX_ACCURACY_THRESHOLD = 100; // meter`

```
function validateAccuracy(accuracy) {
  if (accuracy > MAX_ACCURACY_THRESHOLD) {
    return {
      valid: false,
      message: 'Akurasi GPS terlalu rendah. Pastikan GPS aktif dan berada di area terbuka.'
    };
  }
  return { valid: true };
}
```

Akurasi (meter)	Status	Aksi
≤ 30	Excellent	Presensi diterima
31-100	Acceptable	Presensi diterima dengan warning

Akurasi (meter)	Status	Aksi
> 100	Poor	Presensi ditolak, minta retry

B. Anomaly Detection (Deteksi Anomali)

Sistem mencatat dan menganalisis pola presensi untuk mendeteksi anomali:

1. **Duplicate Detection:** Mencegah presensi ganda dalam rentang waktu singkat
2. **Distance Anomaly:** Mendeteksi perpindahan lokasi yang tidak wajar (misalnya 10km dalam 5 menit)
3. **Timestamp Validation:** Memastikan waktu presensi menggunakan timestamp server, bukan client

// Contoh deteksi presensi duplikat

```
async function checkDuplicatePresensi(userId, geofenceId) {
  const today = new Date();
  today.setHours(0, 0, 0, 0);
```

```
  const existing = await Presensi.findOne({
    where: {
      user_id: userId,
      geofence_id: geofenceId,
      createdAt: { [Op.gte]: today }
    }
  });
```

```
  return existing !== null;
}
```

C. Rate Limiting

Pembatasan jumlah request untuk mencegah abuse:

Endpoint	Limit	Window
POST /api/presensi	5 requests	per 15 menit
POST /api/auth/login	10 requests	per 15 menit
GET /api/*	100 requests	per 15 menit

7.2 Bukti foto (selfie) sebagai verifikasi tambahan

Foto selfie berfungsi sebagai bukti visual yang dapat diverifikasi oleh admin:

Kebijakan Foto

Aspek	Kebijakan
Format	JPEG/PNG

Aspek	Kebijakan
Ukuran Maksimal	5MB
Kamera	Front-facing (selfie mode)
Penyimpanan	Server storage dengan URL reference di database
Retensi	Sesuai kebijakan institusi

7.3 Catatan keterbatasan web terhadap fake GPS

Keterbatasan yang Diketahui

Aspek	Keterbatasan	Mitigasi
Mock Location	Browser tidak dapat mendeteksi mock location app	Accuracy gate + photo verification
GPS Spoofing	Developer tools dapat memodifikasi koordinat	Server-side validation + audit trail
Timestamp	Client timestamp dapat dimanipulasi	Gunakan server timestamp
VPN/Proxy	Dapat mengubah perceived location	Tidak relevan untuk GPS (GPS \neq IP-based)

Strategi Mitigasi

1. **Multi-factor Verification:** Kombinasi GPS + Photo + Timestamp Server
2. **Admin Review:** Presensi dengan flag suspicious dapat di-review manual
3. **Audit Trail:** Semua data disimpan untuk investigasi jika diperlukan
4. **Kebijakan Non-teknis:** Sanksi akademik untuk pemalsuan presensi

8. Arsitektur Sistem

8.1 3-Tier Architecture

GeoProof menggunakan arsitektur **3-tier** untuk memisahkan tanggung jawab sistem menjadi tiga lapisan utama:

1. Presentation Layer (Client / Frontend)

Aplikasi web yang diakses melalui browser (terutama smartphone) oleh mahasiswa dan admin. Layer ini menangani antarmuka pengguna, pengambilan foto (kamera), pengambilan lokasi (GPS), serta pengiriman data ke server melalui HTTP/HTTPS.

2. Application Layer (Backend / REST API Server)

Server aplikasi yang menyediakan REST API untuk autentikasi, pemrosesan presensi, validasi geofence, manajemen data lokasi (geofence), dan rekap presensi. Layer ini juga menangani keamanan (JWT), validasi input, perhitungan jarak (Haversine), serta kontrol mitigasi seperti pembatasan request (rate limiting) bila diterapkan.

3. **Data Layer (Database & Storage)**

Lapisan penyimpanan yang terdiri dari:

- a. **Database MySQL** untuk data terstruktur (user, geofence, presensi, status).
- b. **Storage file** untuk menyimpan foto bukti presensi (selfie), dengan referensi URL/path disimpan ke database.

Arsitektur ini dipilih karena memudahkan pengembangan, pengujian, dan pemeliharaan, serta membuat komponen sistem lebih modular untuk keperluan dokumentasi UML.

8.2 Komponen pendukung

Selain 3-tier inti, sistem juga menggunakan komponen pendukung berikut:

1. **Browser API**
 - a. **Geolocation API** untuk memperoleh latitude, longitude, dan accuracy.
 - b. **MediaDevices (getUserMedia)** untuk akses kamera dan pengambilan foto selfie.
2. **Map Renderin**
 - c. Peta untuk menampilkan titik presensi/marker serta informasi lokasi secara visual menggunakan Leaflet dengan peta dasar dari OpenStreetMap.
3. **File Upload Handling**
 - d. Mekanisme unggah foto dari client ke backend untuk disimpan pada storage server.
4. **Authentication & Authorization**
 - e. JWT untuk menjaga sesi stateless dan membedakan akses antara mahasiswa dan admin (role-based access).
5. **Logging/Audit**
 - f. Pencatatan informasi penting (timestamp server, jarak, accuracy, status validasi) untuk kebutuhan audit dan troubleshooting.

8.3 Alur data ringkas

Alur data utama pada proses presensi GeoProof adalah sebagai berikut:

1. Mahasiswa login ke sistem melalui frontend.
2. Saat melakukan presensi, frontend meminta izin kamera dan lokasi, lalu mengambil:
 - foto selfie,
 - koordinat (lat/lng) dan accuracy.
3. Frontend mengirim data presensi ke backend melalui REST API.
4. Backend melakukan:
 - validasi autentikasi (JWT),
 - validasi data input,
 - perhitungan jarak (Haversine) terhadap geofence aktif,

- penentuan status valid/invalid berdasarkan radius dan aturan accuracy (jika diterapkan).
5. Backend menyimpan foto ke storage dan menyimpan data presensi ke database (termasuk timestamp server, distance_m, accuracy_m, dan status).
 6. Backend mengembalikan respons ke frontend (berhasil/gagal dan alasan).
 7. Mahasiswa dapat melihat riwayat presensi; admin dapat melihat rekap presensi serta mengelola geofence.

9. Teknologi yang Digunakan

9.1 Frontend: React, Browser API (Geolocation, Camera)

React.js

Aspek	Detail
Versi	18.x
Bundler	Create React App (react-scripts)
State Management	React Hooks (useState, useEffect, useContext)
Routing	React Router DOM v6

// Contoh komponen React dengan Hooks

```
function PresensiPage() {
  const [location, setLocation] = useState(null);
  const [photo, setPhoto] = useState(null);
  const [loading, setLoading] = useState(false);

  useEffect(() => {
    // Get GPS on component mount
    navigator.geolocation.getCurrentPosition(
      (pos) => setLocation(pos.coords),
      (err) => console.error(err),
      { enableHighAccuracy: true }
    );
  }, []);

  return (
    <div className="glass-card p-6">
      {/* UI Components */}
    </div>
  );
}
```

Tailwind CSS

Aspek	Detail
Versi	3.4.x
Mode	JIT (Just-In-Time)
Plugins	Forms, Typography
Custom Theme	Dark mode premium

// *tailwind.config.js*

```
module.exports = {
  content: ['./src/**/*.{js,jsx}'],
  theme: {
    extend: {
      colors: {
        primary: '#10b981',
        secondary: '#22d3ee'
      }
    }
  }
}
```

Browser APIs

API	Kegunaan	Permission
Geolocation API	Mendapatkan koordinat GPS	Location
MediaDevices API	Akses kamera untuk selfie	Camera
Fetch API	HTTP requests ke backend	-

9.2 Map: Leaflet + OpenStreetMap

Komponen	Detail
Library	Leaflet 1.9.x
React Wrapper	react-leaflet 4.x
Tile Provider	OpenStreetMap (gratis, open source)
Plugins	Leaflet.markercluster (opsional)

// *Contoh penggunaan Leaflet di React*

```
import { MapContainer, TileLayer, Marker, Circle } from 'react-leaflet';
```

```
function MapView({ center, radius, userLocation }) {
```

```

return (
  <MapContainer center={center} zoom={15}>
    <TileLayer
      url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
      attribution='&copy; OpenStreetMap contributors'
    />
    { /* Geofence circle */ }
    <Circle center={center} radius={radius} color="green" />
    { /* User marker */ }
    {userLocation && <Marker position={userLocation} />}
  </MapContainer>
);
}

```

9.3 Backend: Node.js + Express (REST)

Komponen	Versi	Kegunaan
Node.js	18.x+	JavaScript runtime
Express.js	4.x	Web framework
Sequelize	6.x	ORM untuk MySQL
cors	2.x	Cross-Origin Resource Sharing
dotenv	16.x	Environment variables
morgan	1.x	HTTP request logging

// server.js structure

```

const express = require('express');
const cors = require('cors');
const morgan = require('morgan');

```

```

const app = express();

```

// Middleware

```

app.use(cors({ origin: process.env.FRONTEND_URL }));
app.use(express.json());
app.use(morgan('combined'));

```

// Routes

```

app.use('/api/auth', require('./routes/auth'));

```

```

app.use('/api/presensi', require('./routes/presensi'));
app.use('/api/geofences', require('./routes/geofences'));

// Error handling
app.use((err, req, res, next) => {
  console.error(err.stack);
  res.status(500).json({ message: 'Internal Server Error' });
});

app.listen(process.env.PORT, () => {
  console.log(`Server running on port ${process.env.PORT}`);
});

```

9.4 Security: JWT, bcrypt

JWT (JSON Web Token)

Aspek	Detail
Library	jsonwebtoken
Algorithm	HS256
Expiration	24 jam
Payload	{ userId, email, role }

```

// JWT implementation
const jwt = require('jsonwebtoken');

function generateToken(user) {
  return jwt.sign(
    { userId: user.id, email: user.email, role: user.role },
    process.env.JWT_SECRET,
    { expiresIn: process.env.JWT_EXPIRES_IN }
  );
}

function verifyToken(token) {
  return jwt.verify(token, process.env.JWT_SECRET);
}

bcrypt

```

Aspek	Detail
Library	bcryptjs
Salt Rounds	10
Kegunaan	Password hashing

// Password hashing

```
const bcrypt = require('bcryptjs');
```

```
async function hashPassword(password) {
  const salt = await bcrypt.genSalt(10);
  return bcrypt.hash(password, salt);
}
```

```
async function comparePassword(password, hash) {
  return bcrypt.compare(password, hash);
}
```

9.5 Upload: multer

Aspek	Detail
Library	multer
Storage	Disk storage
File Filter	Images only (JPEG, PNG)
Size Limit	5MB
Naming	userId-timestamp-randomstring.jpg

// Multer configuration

```
const multer = require('multer');
```

```
const path = require('path');
```

```
const storage = multer.diskStorage({
  destination: './uploads',
  filename: (req, file, cb) => {
    const uniqueSuffix = Date.now() + '-' + Math.random().toString(36).substr(2, 6);
    cb(null, req.user.id + '-' + uniqueSuffix + path.extname(file.originalname));
  }
});
```

```

const upload = multer({
  storage,
  limits: { fileSize: 5 * 1024 * 1024 }, // 5MB
  fileFilter: (req, file, cb) => {
    const allowedTypes = /jpeg|jpg|png/;
    const extname = allowedTypes.test(path.extname(file.originalname).toLowerCase());
    const mimetype = allowedTypes.test(file.mimetype);

    if (extname && mimetype) {
      cb(null, true);
    } else {
      cb(new Error('Only images allowed'));
    }
  }
});

```

9.6 Database: MySQL

Aspek	Detail
DBMS	MySQL 8.0+
ORM	Sequelize 6.x
Migrations	Sequelize CLI
Connection Pool	Default (5 connections)

Database Schema

-- Users Table

```

CREATE TABLE `users` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL,
  `email` VARCHAR(255) NOT NULL UNIQUE,
  `password` VARCHAR(255) NOT NULL,
  `nim` VARCHAR(20) UNIQUE,
  `role` ENUM('mahasiswa', 'admin') DEFAULT 'mahasiswa',
  `createdAt` DATETIME NOT NULL,
  `updatedAt` DATETIME NOT NULL,
  PRIMARY KEY (`id`)
);

```

-- Geofences Table

```
CREATE TABLE `geofences` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL,
  `latitude` DECIMAL(10, 8) NOT NULL,
  `longitude` DECIMAL(11, 8) NOT NULL,
  `radius` INT NOT NULL,
  `is_active` TINYINT(1) DEFAULT 1,
  `createdAt` DATETIME NOT NULL,
  `updatedAt` DATETIME NOT NULL,
  PRIMARY KEY (`id`)
);
```

-- Presensis Table

```
CREATE TABLE `presensis` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `user_id` INT NOT NULL,
  `geofence_id` INT,
  `latitude` DECIMAL(10, 8) NOT NULL,
  `longitude` DECIMAL(11, 8) NOT NULL,
  `accuracy` FLOAT,
  `distance_to_center` FLOAT,
  `photo_url` VARCHAR(255),
  `status` ENUM('valid', 'invalid', 'pending') DEFAULT 'pending',
  `verification_status` ENUM('pending', 'approved', 'rejected') DEFAULT 'pending',
  `verification_notes` TEXT,
  `verified_by` INT,
  `verified_at` DATETIME,
  `createdAt` DATETIME NOT NULL,
  `updatedAt` DATETIME NOT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`user_id`) REFERENCES `users`(`id`),
  FOREIGN KEY (`geofence_id`) REFERENCES `geofences`(`id`),
  FOREIGN KEY (`verified_by`) REFERENCES `users`(`id`)
);
```

10. Deployment Plan

10.1 Domain/Subdomain

Opsi Konfigurasi Domain

Konfigurasi	Frontend	Backend API
Opsi A (Subdomain)	geoproof.domain.com	api.geoproof.domain.com
Opsi B (Path-based)	domain.com	domain.com/api
Opsi C (Port-based)	domain.com	domain.com:3001

Rekomendasi: Opsi B (Path-based) untuk kemudahan CORS dan SSL management.

DNS Configuration

Type	Name	Value	TTL
A	@	[Server IP]	3600
A	api	[Server IP]	3600
CNAME	www	@	3600

10.2 Environment

Development Environment

.env (development)

NODE_ENV=development

PORT=3001

Database

DB_HOST=localhost

DB_PORT=3306

DB_NAME=geoproof_dev

DB_USER=root

DB_PASSWORD=password

JWT

JWT_SECRET=dev_secret_key_not_for_production

JWT_EXPIRES_IN=24h

Frontend

REACT_APP_API_URL=http://localhost:3001

Production Environment

.env (production)

NODE_ENV=production

PORT=3001

Database

DB_HOST=localhost

DB_PORT=3306

```

DB_NAME=geoproof_prod
DB_USER=geoproof_user
DB_PASSWORD=[SECURE_PASSWORD]

# JWT (generate with: openssl rand -hex 32)
JWT_SECRET=[64_CHARACTER_RANDOM_STRING]
JWT_EXPIRES_IN=24h

# CORS
FRONTEND_URL=https://geoproof.yourdomain.com

# Frontend
REACT_APP_API_URL=https://geoproof.yourdomain.com/api

```

10.3 Catatan konfigurasi umum

A. SSL/TLS Certificate

Aspek	Requirement	Notes
Certificate	Wajib untuk production	Required untuk Camera & Geolocation API
Provider	Let's Encrypt (gratis)	Auto-renewal setiap 90 hari
Protocol	TLS 1.2+	Disable TLS 1.0 dan 1.1

B. CORS Configuration

```

// Production CORS setup
const corsOptions = {
  origin: process.env.FRONTEND_URL,
  credentials: true,
  methods: ['GET', 'POST', 'PUT', 'DELETE', 'OPTIONS'],
  allowedHeaders: ['Content-Type', 'Authorization']
};

```

```
app.use(cors(corsOptions));
```

C. File Upload Configuration

```

// Serve static files
app.use('/uploads', express.static('uploads'));

// Ensure uploads directory exists
const fs = require('fs');
if (!fs.existsSync('./uploads')) {
  fs.mkdirSync('./uploads', { recursive: true });
}

```

}

D. Database Backup

Daily backup script (cron job)

```
mysqldump -u geoproof_user -p geoproof_db > backup_$(date +%Y%m%d).sql
```

Crontab entry (daily at 2 AM)

```
0 2 * * * /path/to/backup_script.sh
```

E. Deployment Checklist

Pre-Deployment:

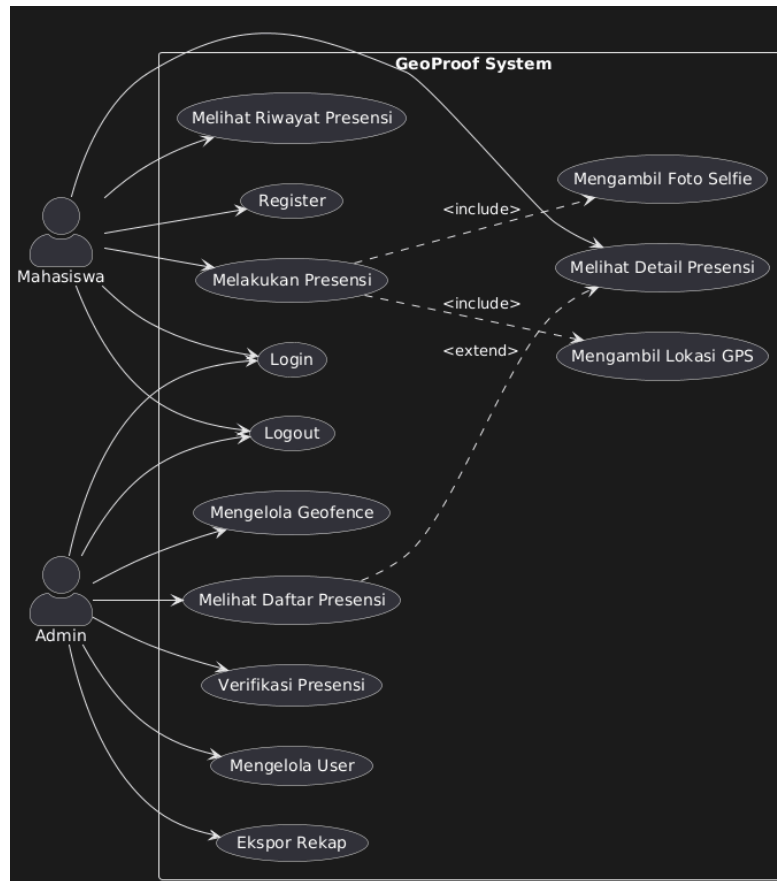
- ☐ Environment variables configured
- ☐ Database migrations executed
- ☐ SSL certificate installed
- ☐ CORS configured
- ☐ File permissions set (uploads folder)

Post-Deployment:

- ☐ Health check endpoint responding
- ☐ Login/Register working
- ☐ Presensi submission working
- ☐ Photo upload working
- ☐ Admin features accessible
- ☐ Mobile responsiveness verified

11. Rencana Artefak UML

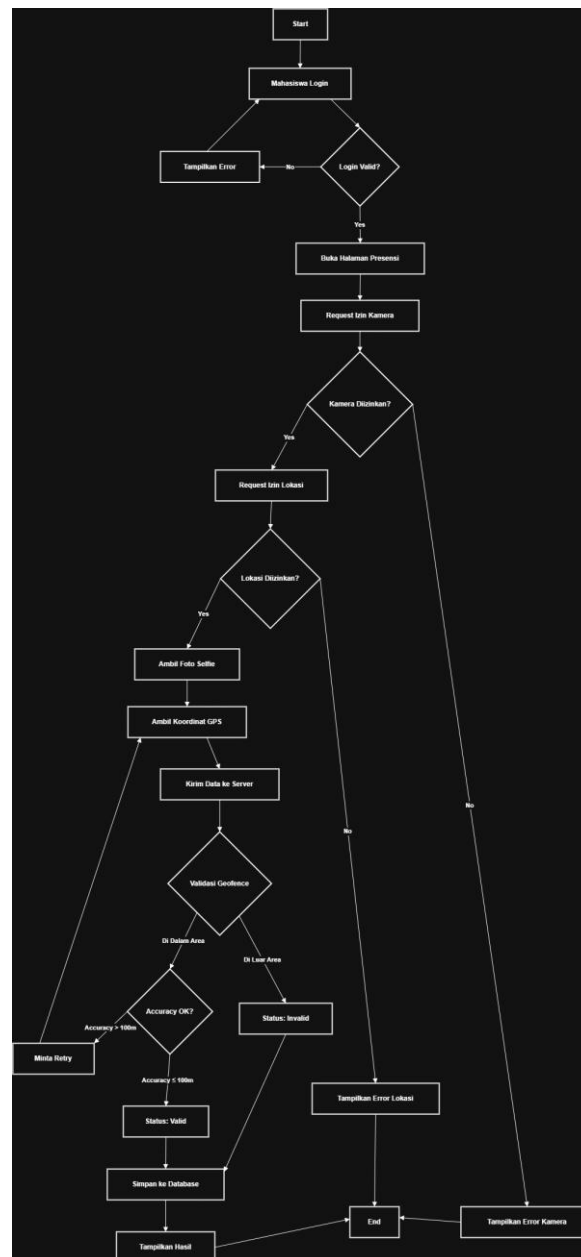
11.1 Use Case Diagram



Deskripsi Use Case Diagram:

Actor	Use Case	Deskripsi
Mahasiswa	Login/Register	Autentikasi untuk mengakses sistem
Mahasiswa	Melakukan Presensi	Submit presensi dengan foto dan lokasi
Mahasiswa	Melihat Riwayat	Melihat daftar presensi yang sudah dilakukan
Admin	Mengelola Geofence	CRUD area presensi
Admin	Verifikasi Presensi	Approve/reject presensi mahasiswa
Admin	Mengelola User	CRUD data user

11.2 Flowchart



11.3 Activity Diagram

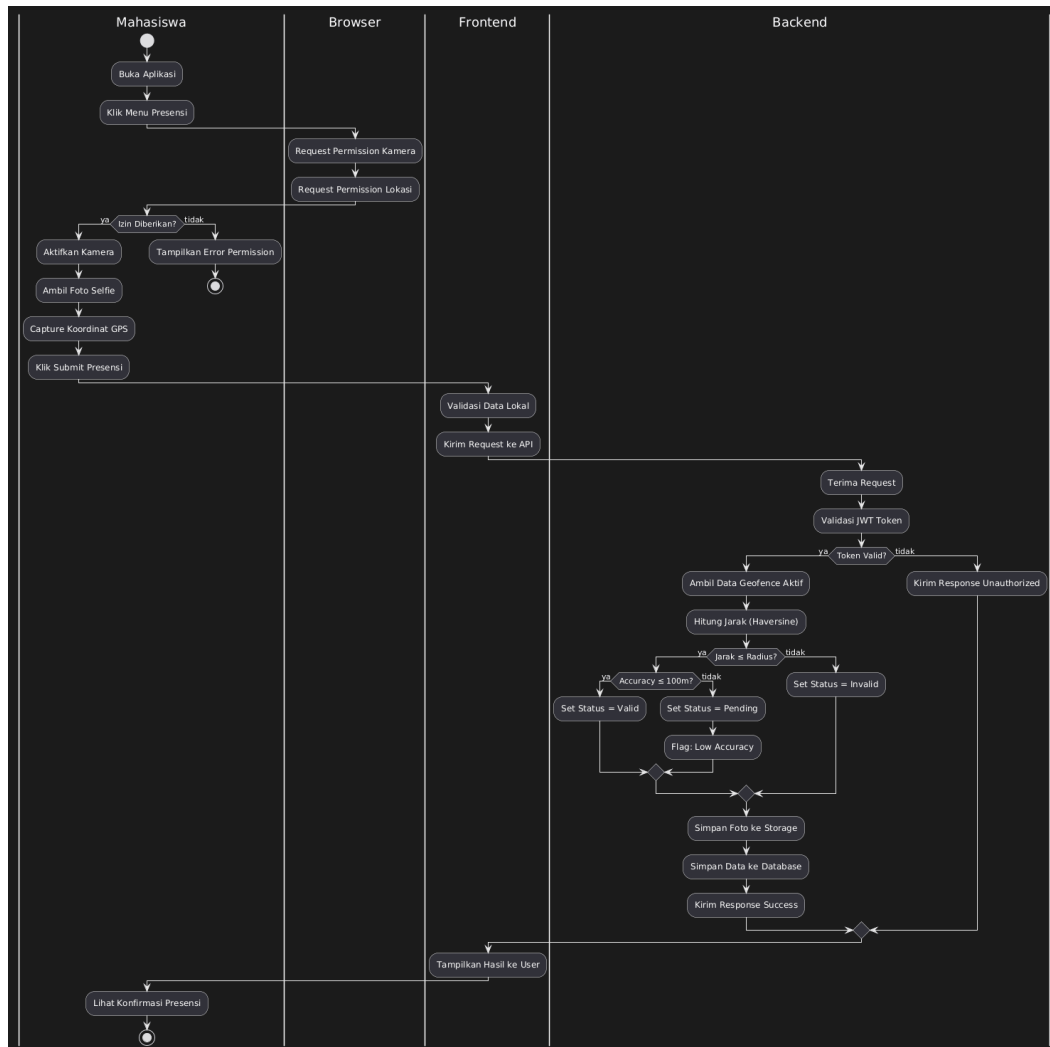


Figure 1. Activity Diagram - Submit Presensi

11.4 Sequence Diagram

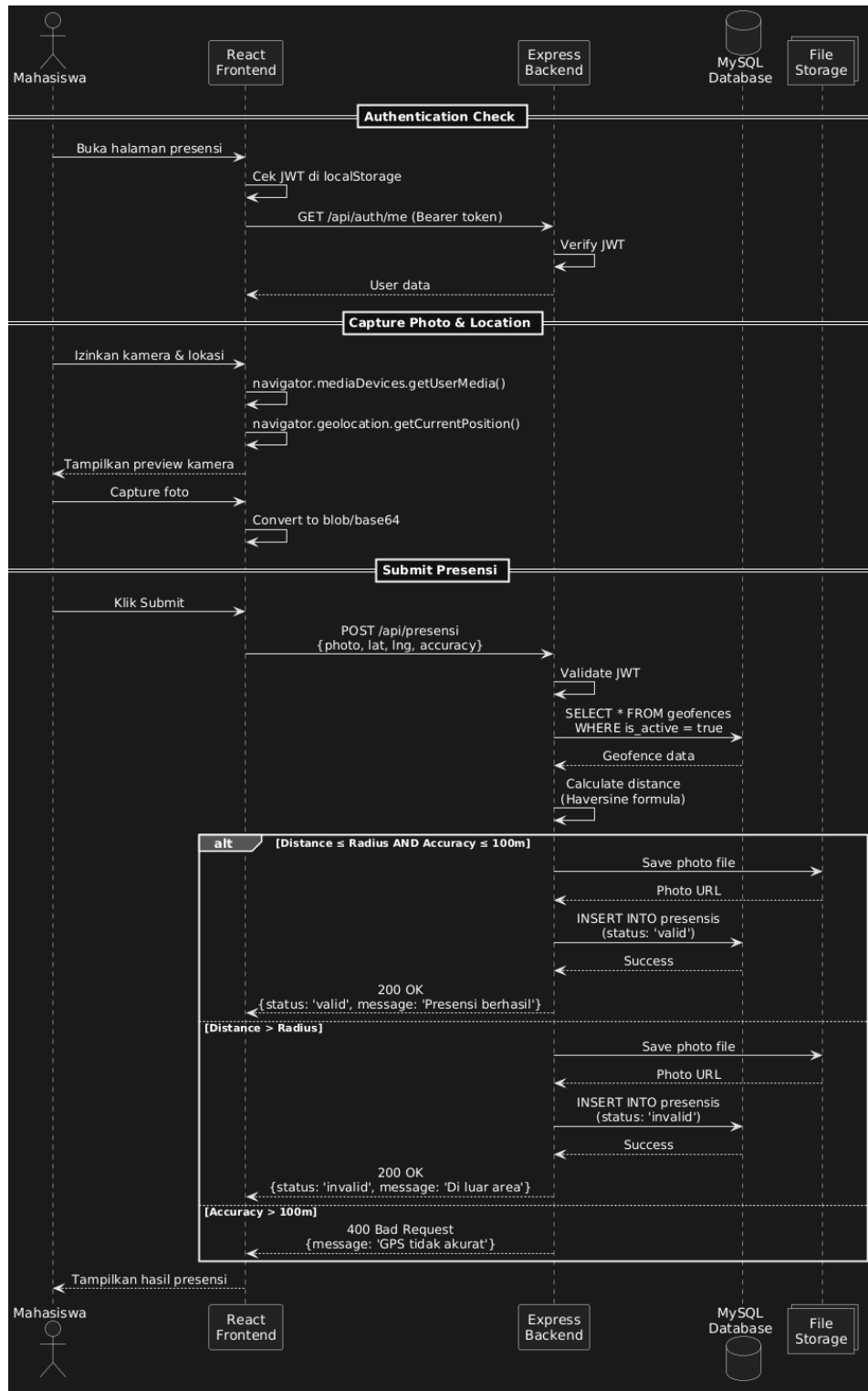
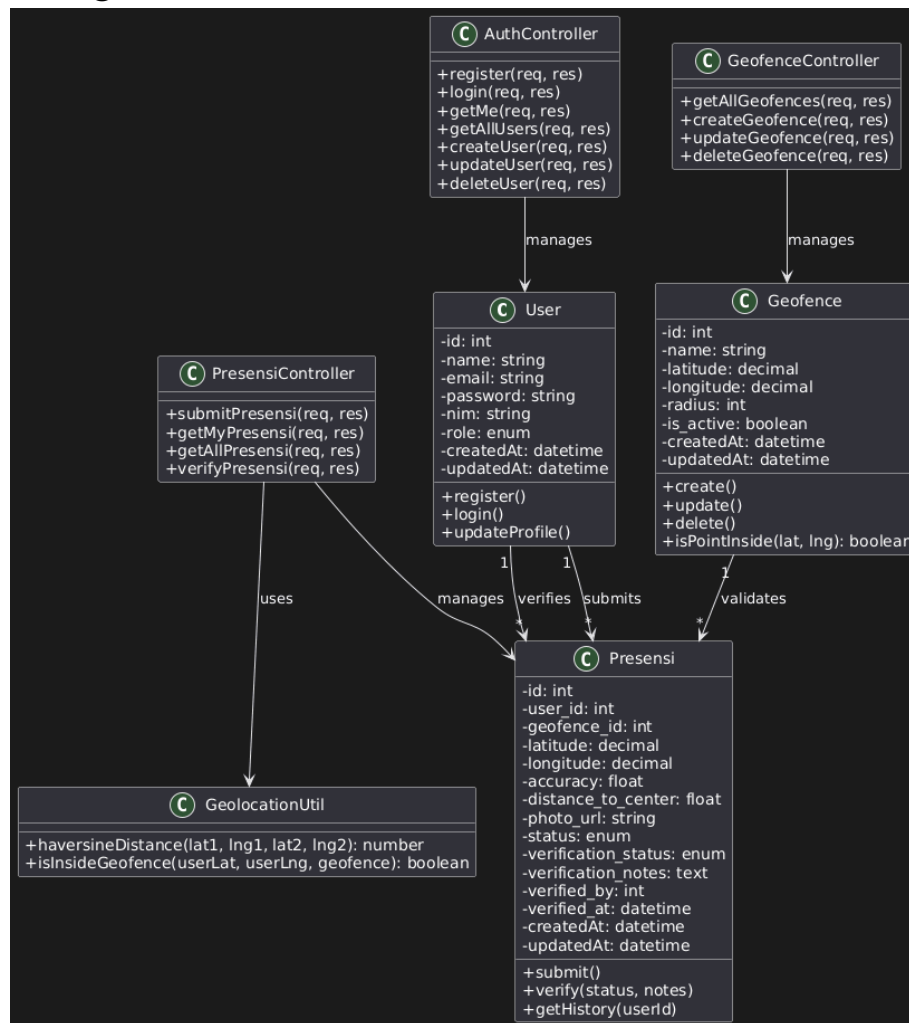
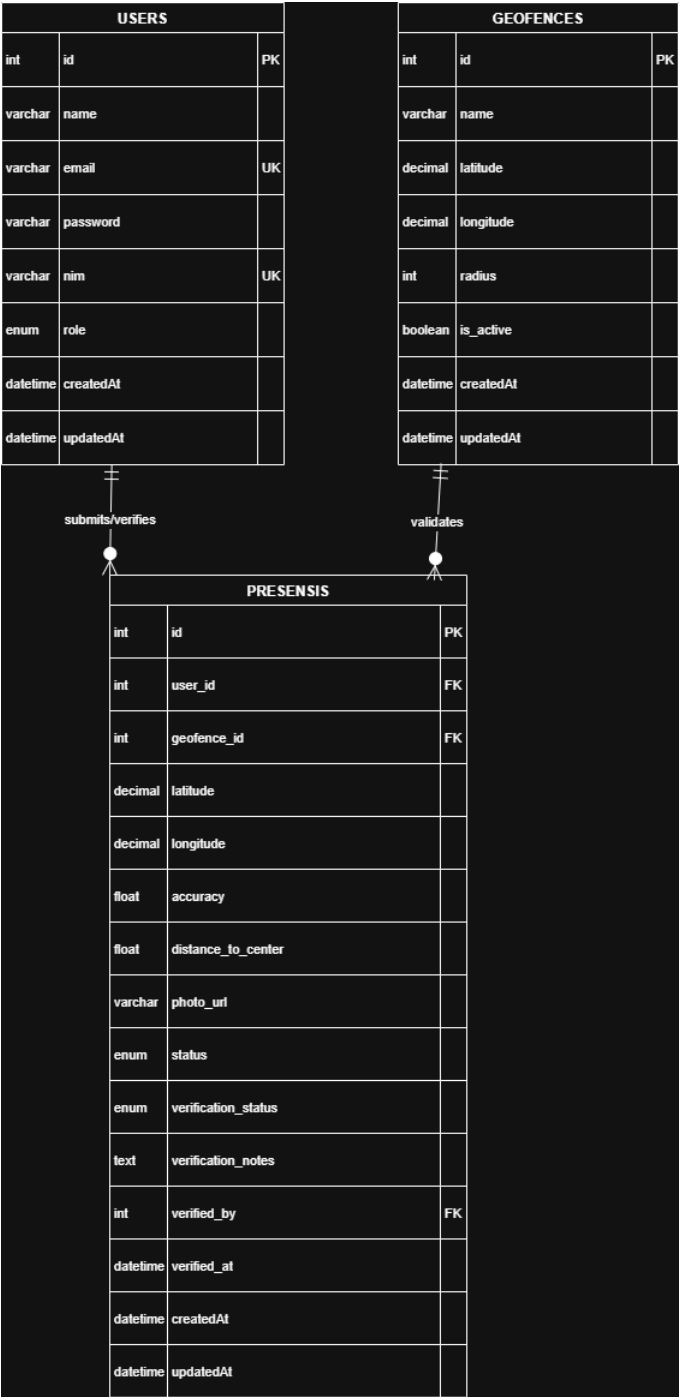


Figure 2. Sequence Diagram - Proses Presensi

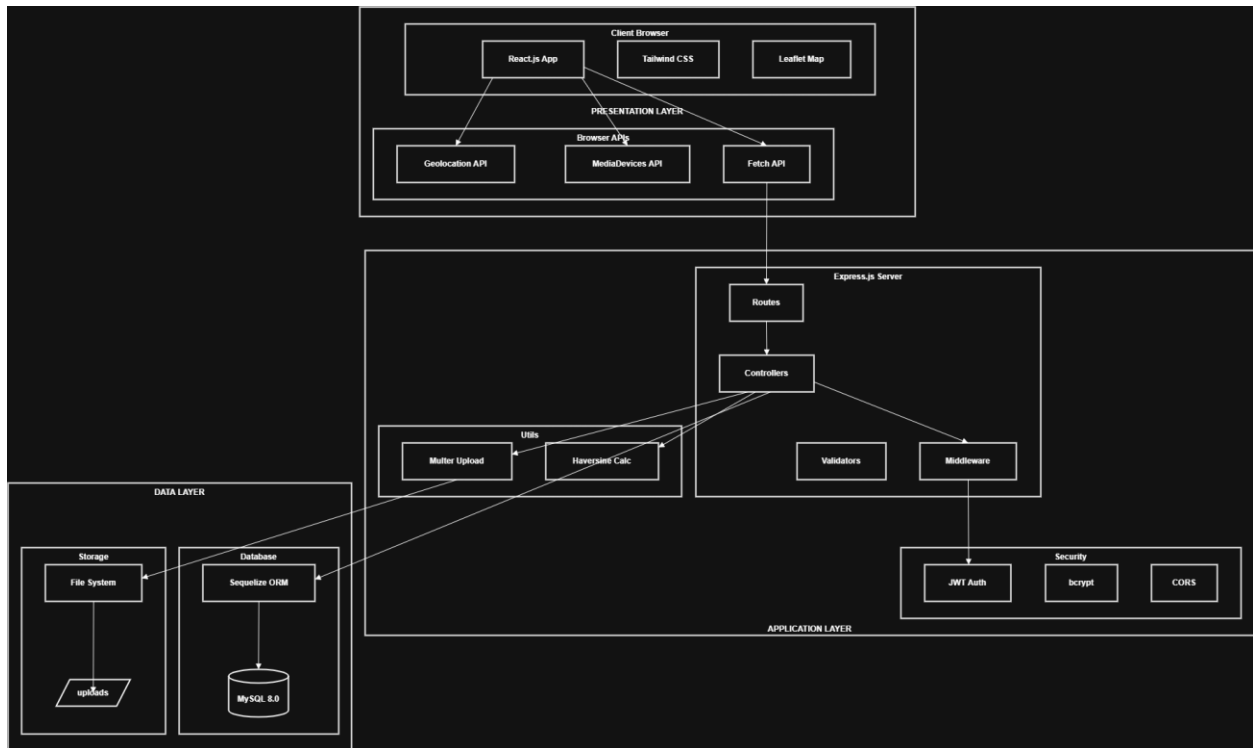
11.5 Class Diagram



11.6 ERD



11.7 Arsitektur Diagram



Penjelasan Arsitektur 3-Tier:

Layer	Komponen	Tanggung Jawab
Presentation	React, Tailwind, Leaflet, Browser APIs	UI rendering, user interaction, sensor access
Application	Express, Controllers, JWT, Multer	Business logic, authentication, file handling
Data	MySQL, Sequelize, File Storage	Data persistence, query processing

12. Penutup

12.1 Ringkasan keputusan desain

Selama pengembangan GeoProof, beberapa keputusan desain penting telah diambil:

A. Arsitektur & Teknologi

Keputusan	Alasan
3-Tier Architecture	Memisahkan concerns untuk maintainability dan scalability
React + Tailwind	Rapid development dengan component-based UI dan utility-first CSS
Node.js + Express	JavaScript full-stack, ecosystem luas, easy deployment

Keputusan	Alasan
MySQL + Sequelize	Relational data structure, mature ORM dengan migration support

B. Fitur & Keamanan

Keputusan	Alasan
Circular Geofence	Sederhana, mudah dihitung dengan Haversine formula
Photo + GPS	Multi-factor evidence untuk validitas presensi
Server-side Validation	Mencegah manipulasi data dari client
JWT Stateless Auth	Scalable, tidak perlu session storage
Accuracy Threshold	Mitigasi untuk GPS yang tidak akurat

C. UI/UX

Keputusan	Alasan
Mobile-First	Target utama adalah pengguna smartphone
Dark Mode Premium	Modern, nyaman untuk mata, trending design
Glassmorphism	Visual appeal dengan performance yang baik
Dual View (Card/Table)	Optimal untuk mobile dan desktop

12.2 Next step pengerjaan

Phase 1: Core Development

- [x] Setup project structure
- [x] Implement authentication (JWT)
- [x] Create presensi submission flow
- [x] Implement geofence validation
- [x] Photo upload integration
- [x] Basic admin features

Phase 2: Enhancement

- [x] User management (CRUD)
- [x] Geofence management with map
- [x] Premium dark mode UI
- [x] Mobile responsive design
- [x] Admin presensi monitoring

Phase 3: Testing & Documentation

- [x] Create documentation files
- [] Unit testing implementation

- ☐ Integration testing
- ☐ User acceptance testing
- ☐ Performance optimization

Phase 4: Deployment

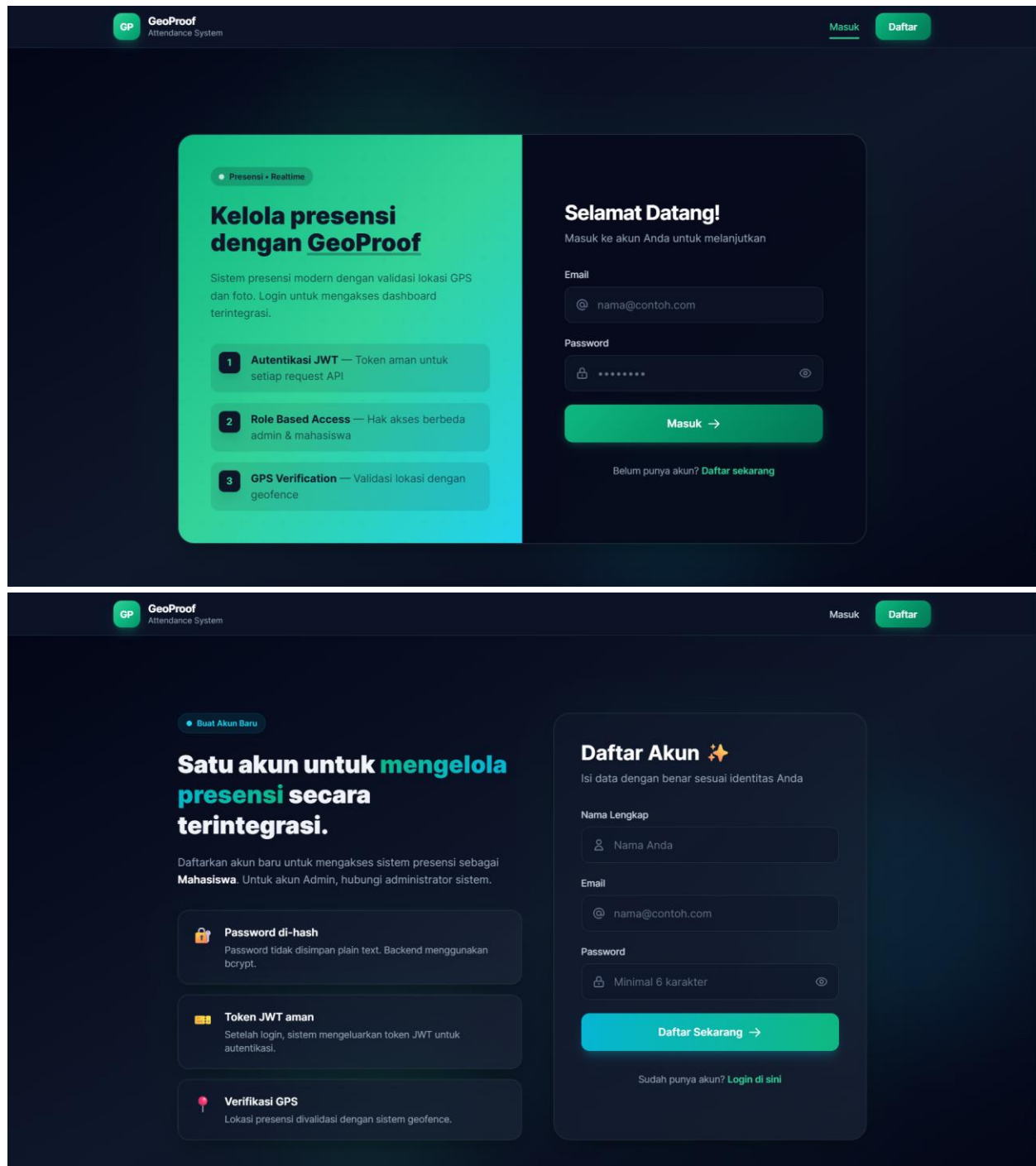
- ☐ Setup production environment
- ☐ Configure SSL certificate
- ☐ Deploy to hosting provider
- ☐ Domain configuration
- ☐ Post-deployment testing

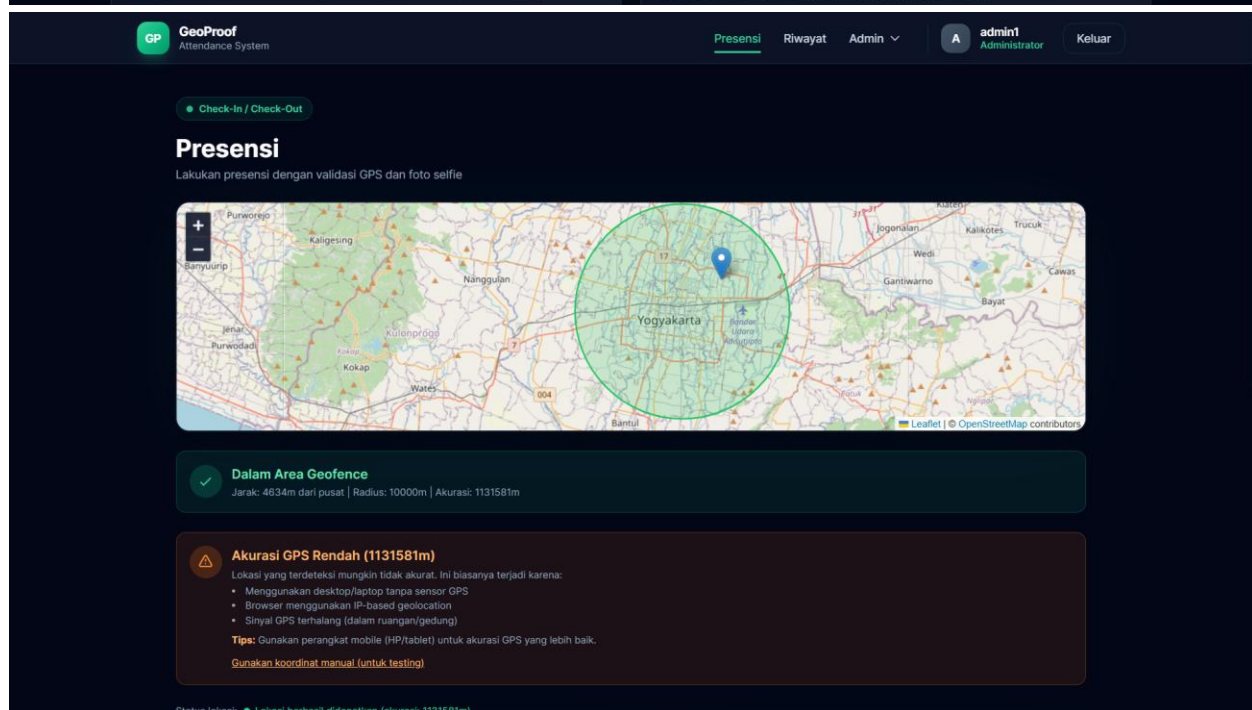
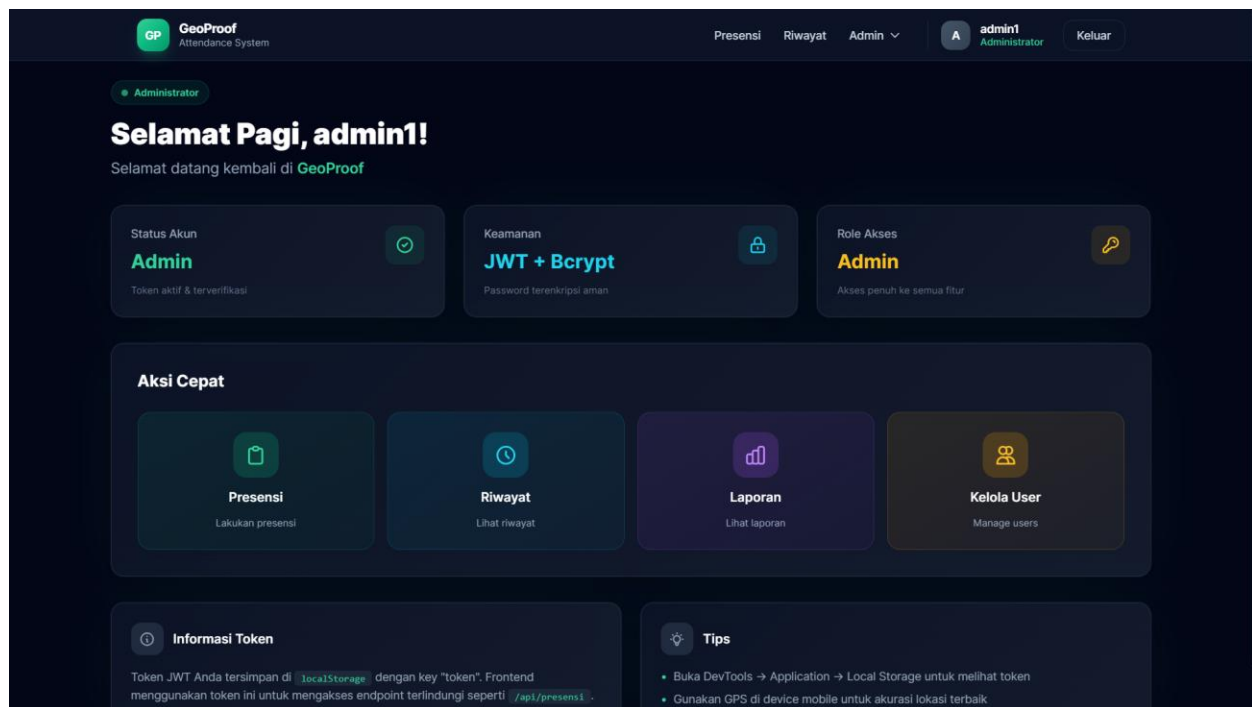
Phase 5: Future Improvements

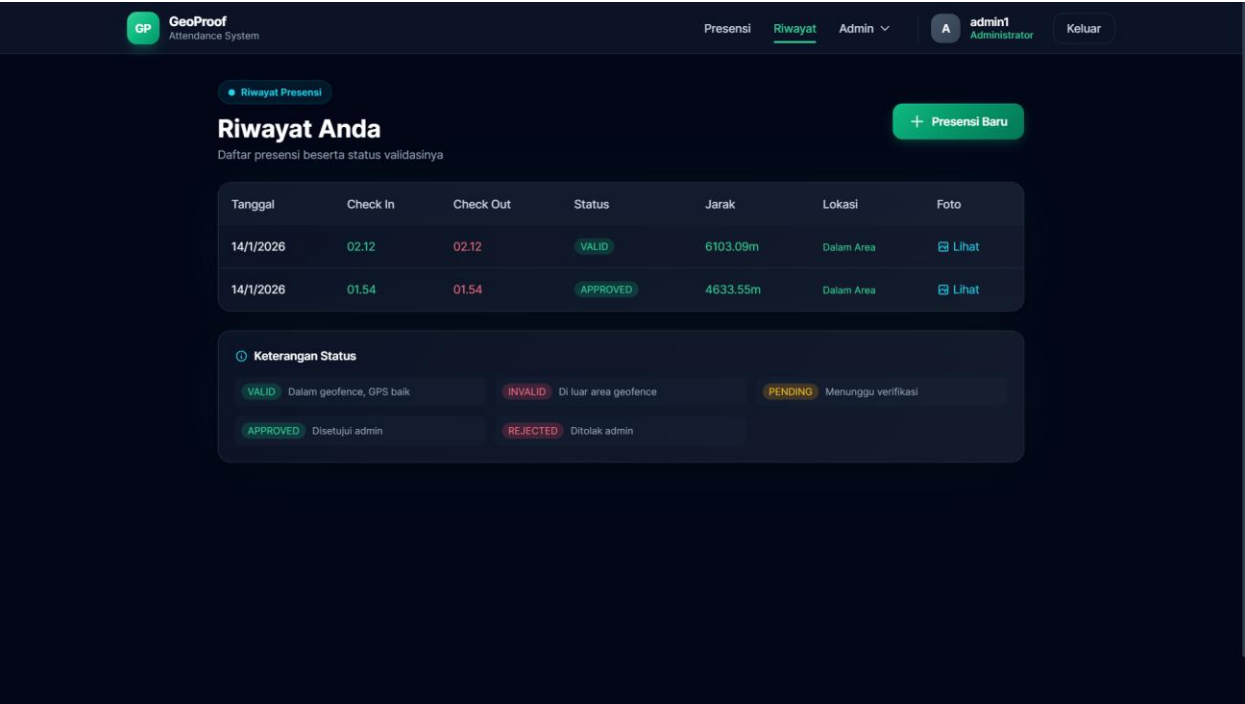
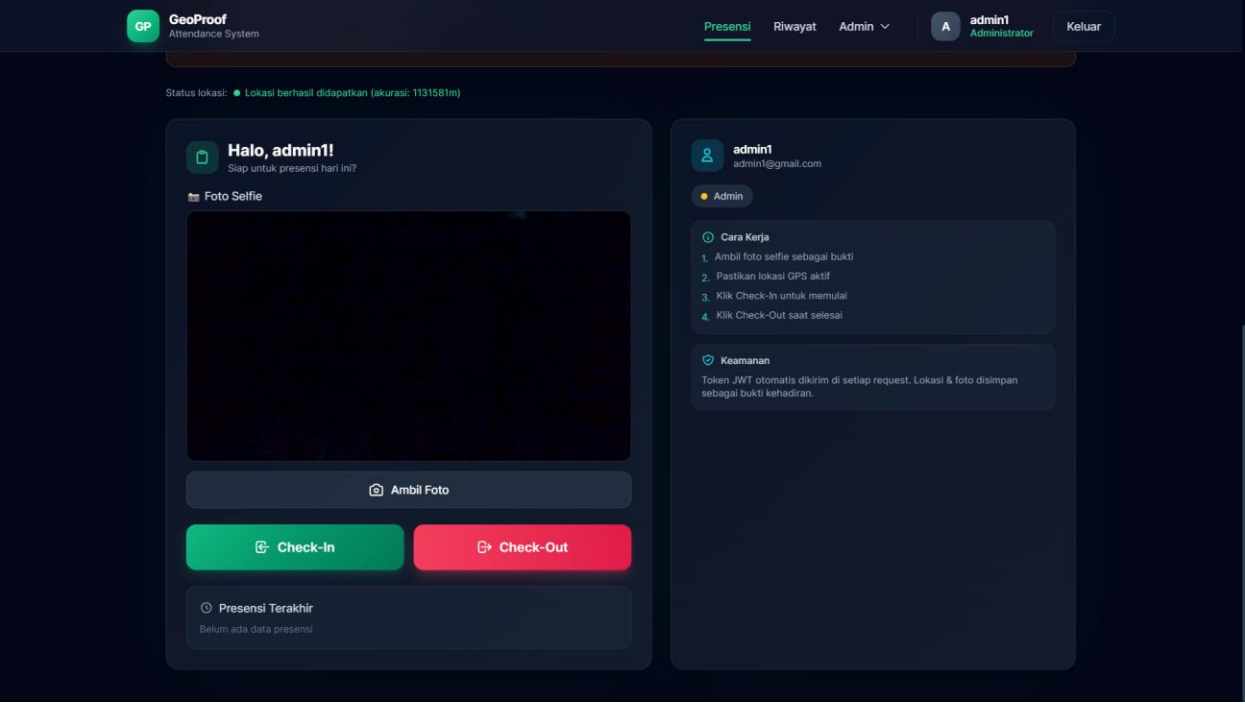
Feature	Priority	Status
Export rekap ke Excel/PDF	Medium	Planned
Push notification	Low	Backlog
Offline mode (PWA)	Low	Backlog
Multiple geofence support	Medium	Partial
Face detection (basic)	Low	Backlog
QR Code presensi	Low	Backlog

13. Documentation

13.1 Tampilan Web per Homepage







GP

GeoProof
Attendance System

PresensiRiwayatAdmin

Aadmin1
AdministratorKeluar

Admin Panel

Kelola User

Kelola akun mahasiswa dan admin sistem presensi

+ Tambah UserHapus Akun Saya

Total User

2

Mahasiswa

0

Admin

2

User	Role	Terdaftar	Aksi
admin1 (Anda) admin1@gmail.com	admin	14 Jan 2026	
admin0 admin0@gmail.com	admin	02 Des 2025	Admin

GP

GeoProof
Attendance System

PresensiRiwayatAdmin

Aadmin1
AdministratorKeluar

+ Tambah Geofence

Manajemen Geofence

Kelola area geofence untuk validasi presensi

Daftar Geofence

ringroadAktif

area ringroad

Koordinat: -7.79442578, 110.37564051 | Radius: 10000m

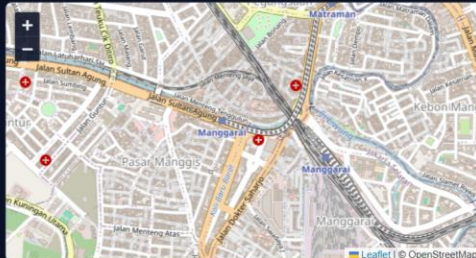
Edit

Manajemen Geofence

Kelola area geofence untuk validasi presensi

Tutup Form

Tambah Geofence Baru



Nama Geofence *

Contoh: Kampus A

Deskripsi

Deskripsi lokasi (opsional)

Latitude *

-6.2088

Longitude *

106.8456

Radius (meter) *

100

- Aktifkan geofence ini (hanya satu yang bisa aktif)

Simpan

Batal

Daftar Geofence

ringroad Aktif

- Monitoring Real-time

Monitoring Presensi

Lihat dan verifikasi semua data presensi mahasiswa

Dari Tanggal

Sampai Tanggal

Semu

mm/dd/yyyy

mm/dd/yyyy

- Mencurigakan

Filter

Reset

• Admin Only

Laporan Presensi Harian



Lihat dan filter data presensi harian seluruh mahasiswa

Cari berdasarkan nama
Nama Mahasiswa

Dari tanggal
mm/dd/yyyy

Sampai tanggal
mm/dd/yyyy

Terapkan Filter

NAMA	CHECK-IN	CHECK-OUT	BUKTI FOTO
admin1	14/1/2026, 01.54.37	14/1/2026, 01.54.44	 Lihat
admin1	14/1/2026, 02.12.13	14/1/2026, 02.12.14	 Lihat