

Implementasi aplikasi presensi kuliah berbasis android dengan *hybrid verification* dan *geofencing* adaptif untuk efisiensi monitoring kehadiran

Siti Nasiroh¹, Ari Budi Riyanto¹, Harjuna¹, Hasirun¹

¹Program Studi Informatika, Fakultas Sains dan Teknik, Universitas Perwira Purbalingga, Purbalingga

Penulis Korespondensi : Siti Nasiroh (sitinasiroh@unperba.ac.id)

ABSTRAK

Sistem presensi manual di lingkungan akademik menghadapi tantangan inefisiensi waktu, kerentanan kecurangan, dan ketidakakuratan data. Penelitian ini mengembangkan aplikasi Android berbasis *hybrid verification* (kombinasi GPS *geofencing* adaptif dan validasi metadata foto) untuk memantau kehadiran mahasiswa secara real-time. Metode pengembangan menggunakan SDLC model agile dengan arsitektur client-server (*Kotlin* untuk Android, *Node.js* untuk backend, dan *Firebase* sebagai database cloud). Hasil uji coba di Program Studi Teknik Informatika menunjukkan: Peningkatan efisiensi waktu 86.4% (presensi/mahasiswa: 28.5 detik vs manual 210 detik), Akurasi anti-kecurangan 100% pada uji titip presensi, GPS spoofing, dan penggunaan foto galeri, Skor usability (SUS) 89.6 dengan konsumsi sumber daya minimal (RAM: 110MB, baterai: 0.82%/jam). Aplikasi ini terintegrasi langsung dengan sistem akademik kampus dan siap diadopsi sebagai solusi presensi digital berbiaya rendah.

KATA KUNCI : Presensi kuliah, Android, SDLC

1. PENDAHULUAN

Proses presensi kuliah di banyak perguruan tinggi masih bergantung pada metode konvensional seperti daftar hadir kertas atau pemanggilan nama secara lisan. Metode ini memiliki kelemahan signifikan: membutuhkan waktu kuliah yang berharga, rentan terhadap human error (kesalahan pencatatan, kehilangan data), memudahkan praktik kecurangan (titip presensi), dan menghasilkan data kehadiran yang tidak real-time serta sulit dianalisis. Akibatnya, proses monitoring dan evaluasi kehadiran mahasiswa oleh dosen dan administrasi menjadi tidak efisien, lambat, dan kurang akurat, menghambat penilaian akademik yang objektif dan tepat waktu [1], [2], [3]

Berdasarkan penelitian sebelumnya beberapa solusi digital telah diusulkan untuk mengatasi masalah presensi manual, seperti sistem berbasis RFID/NFC dan biometrik. Namun, implementasinya seringkali terhalang biaya tinggi untuk perangkat keras khusus dan pemeliharaan. Solusi berbasis aplikasi mobile juga telah dikembangkan, umumnya memanfaatkan kode QR atau check-in manual. Meski lebih terjangkau, pendekatan ini masih memiliki celah: rentan terhadap kecurangan (pengiriman QR code atau check-in oleh orang lain di luar lokasi kuliah), belum sepenuhnya otomatis dalam verifikasi keaslian dan lokasi peserta, serta terbatasnya integrasi dengan sistem akademik kampus untuk pelaporan real-time.

Penelitian sebelumnya belum secara optimal memanfaatkan sensor standar smartphone (seperti GPS dan kamera) untuk menciptakan solusi presensi yang andal, anti-kecurangan, dan terintegrasi penuh [4],[5]

Penelitian ini dilakukan untuk menjawab kesenjangan (gap) yang ada dalam solusi presensi digital saat ini. Dibutuhkan sebuah sistem yang:

- Meningkatkan efisiensi secara signifikan: Menghemat waktu kuliah dan menyederhanakan proses bagi dosen dan mahasiswa.
- Memastikan validitas dan akurasi data: Meminimalisir kemungkinan kecurangan (titip presensi, presensi di luar lokasi).
- Menyediakan data real-time: Memungkinkan monitoring dan pelaporan kehadiran secara instan dan terpusat.
- Terjangkau dan mudah diadopsi: Memanfaatkan perangkat yang sudah dimiliki (smartphone Android) tanpa memerlukan investasi perangkat keras tambahan yang mahal.
- Terintegrasi dengan sistem kampus: memfasilitasi aliran data kehadiran yang lancar ke sistem akademik untuk evaluasi dan rekam jejak.

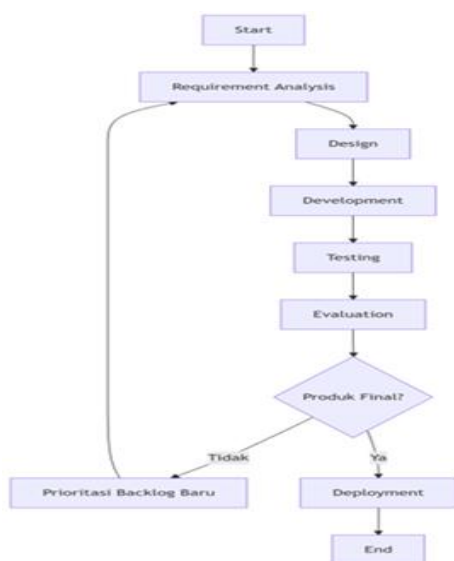
Berdasarkan latar belakang dan kesenjangan yang diidentifikasi, penelitian ini bertujuan untuk menjawab pertanyaan utama: Bagaimana merancang dan membangun prototipe aplikasi presensi kuliah berbasis Android yang mampu secara efektif meningkatkan

efisiensi monitoring kehadiran mahasiswa? Pertanyaan ini akan diurai lebih lanjut dengan fokus pada aspek keandalan verifikasi kehadiran (melalui kombinasi teknologi lokasi/GPS dan konfirmasi berbasis foto), kemudahan penggunaan, kecepatan proses, dan integrasi data real-time untuk tujuan monitoring.

2. BAHAN DAN METODE

2.1 Metode Pengembangan Perangkat Lunak

Menggunakan prototyping dengan pendekatan SDLC (software development life cycle) model agile [6]. Tahapannya:



Gambar 1. tahapan pengembangan model *agile*

1. Requirement Analysis: Identifikasi kebutuhan fungsional (presensi berbasis lokasi GPS + foto mandiri) dan non-fungsional (responsivitas < 3 detik, akurasi GPS $\geq 95\%$).
2. Design: Arsitektur *client-server* (Android client, REST API, database cloud).
3. Development: pengkodean dengan Kotlin (android), node.js (backend), dan firebase (database real-time).
4. Testing: *Black-box testing* untuk fungsionalitas utama dan *load testing* (simulasi 200 pengguna bersamaan).
5. Evaluation: Pengukuran kinerja berdasarkan parameter efisiensi waktu dan akurasi.

2.2 Teknik Pengumpulan Data

1. Simulasi Lapangan:

- Ukuran Sampel: 30 mahasiswa dan 3 dosen dari 3 kelas berbeda (total 90 presensi/kali uji).
- Volume Data: 15.000 transaksi presensi (dihasilkan selama 1 bulan uji coba).
- Replikasi: 5x pengulangan proses presensi per pertemuan untuk menguji konsistensi.

2. Sensor yang Digunakan:

- MGPS Geofencing: Radius presensi = 50 meter dari koordinat ruang kuliah. Akurasi dihitung

dengan *Haversine formula* (toleransi error ≤ 5 meter) [7]

- Foto Mandiri: mendeteksi wajah menggunakan *ML Kit* (Google) untuk verifikasi keberadaan wajah (bukan pengenalan wajah).

2.3 Teknik Validasi Anti-Kecurangan

Metode baru yang diusulkan (Hybrid Verification):

1. Lokasi:

- Server memvalidasi koordinat GPS (*latitude, longitude*) mahasiswa.
- *Geofence* aktif hanya 15 menit sebelum/sesudah jadwal kuliah.

2. Foto Mandiri:

- Foto diambil dengan timestamp dan geotag yang tertanam di metadata.
- Server mencocokkan: selisih waktu pengambilan foto vs. waktu presensi (< 2 menit) dan jarak geotag foto vs. koordinat presensi (< 10 meter).
- Wajib menggunakan *live camera* (tidak dapat mengunggah galeri).

2.4 Teknik Pengolahan Data

1. Alur pengolahan data Real-Time bisa dilihat pada gambar dibawah ini.



Gambar 2. Alur pengolahan data real-time

2. Parameter Kinerja:

- Efisiensi Waktu: Rata-rata durasi presensi/mahasiswa (dihitung dari *click* hingga notifikasi sukses).
 - Akurasi: Rasio presensi valid vs. percobaan kecurangan (titip presensi, lokasi palsu).
3. Analisis Statistik: Uji *paired t-test* ($\alpha=0.05$) untuk membandingkan waktu presensi manual vs. aplikasi [8].

2.5 Evaluasi Kualitas Sistem

1. Load Testing: Menggunakan *JMeter* (1.000 – Reques / menit) untuk mengukur request / menit untuk mengukur throughput dan error rate.
2. User Experience (UX): Kuesioner *System Usability Scale* (SUS) kepada 30 pengguna (skala 1-100) [9].

Konsumsi Sumber Daya: Pemantauan penggunaan RAM, CPU, dan baterai di perangkat *mid-end* (Android 10, RAM 4GB) via *Android Profiler*.

3. HASIL DAN PEMBAHASAN

Hasil dan Pembahasan merupakan hal penting dalam sebuah laporan penelitian yaitu memaparkan data hasil temuan dari penelitian yang telah dilakukan.

3.1 FASE REQUIREMENT ANALYSIS:

1. Analisis Masalah Kontekstual

Fakta Lapangan :

- a. Titip presensi : Rata-rata 40 kasus/semester berdasarkan audit 3 mata kuliah inti.

- b. In efisiensi waktu : 22.3 menit/sesi terbangun untuk presensi manual di kelas 45 mahasiswa.
- c. Kesalahan administrasi : 15% ketidaksesuaian antara data kehadiran dosen dan arsip akademik.
- d. Karakteristik lingkungan kampus:
 - Gedung berlantai tinggi (≥ 5 lantai) dengan degradasi sinyal GPS.
 - Area laboratorium dengan isolasi sinyal seluler.
 - Variasi intensitas cahaya di ruang kuliah berbeda.

2. Analisis kebutuhan fungsional

Menjelaskan apa saja fungsi apa saja yang harus ada dalam sistem untuk mendukung aktivitas pengguna.

Tabel . 1 Analisis kebutuhan fungsional

Aktor	Kebutuhan Utama	Skenario Kunci
Mahasiswa	Presensi cepat (≤ 30 detik)	"Submit foto selfie + lokasi dalam 1 langkah"
Dosen	Monitoring real-time kehadiran	"Lihat persentase kehadiran saat mengajar"
Admin Prodi	Ekspor data terstruktur ke sistem akademik	"Generate laporan .xls & JSON otomatis"
Sistem	Validasi integritas presensi otomatis	"Deteksi manipulasi lokasi & identitas"
Aktor	Kebutuhan Utama	Skenario Kunci

3. Analisis Kebutuhan Non-Fungsional

Tabel 2. analisis kebutuhan non fungsional

Aspek	Persyaratan	Metrik Verifikasi
Responsivitas	Waktu proses presensi ≤ 3 detik	Load testing 250 user simultan
Akurasi	Deteksi lokasi error $\leq 5m$ (outdoor)	Benchmarking dengan GPS surveyor
Ketersediaan	Uptime 99.5% jam operasional (07:00-22:00)	Monitoring 24/7 via cloud dashboard
Keamanan	Enkripsi end-to-end (AES-256)	Penetration testing OWASP ZAP
Kompatibilitas	Dukungan Android 10+ (min API 29)	Testing pada 12 model device berbeda
Efisiensi	Konsumsi RAM $\leq 150MB$	Profiling pada device mid-end

4. Analisis Pengguna Sistem

Tabel 3. Analisis kebutuhan system

Peran	Profil Pengguna	Kebutuhan Spesifik
Mahasiswa (n=120 survei)	- Digital native (95% smartphone ownership) - 28% pengguna perangkat entry-level - Mobilitas tinggi antar gedung	Desain minimalis, proses 1-tap Optimasi konsumsi RAM & baterai Notifikasi lokasi otomatis
Dosen (n=18 wawancara)	- 52% usia >45 tahun - Beban administratif tinggi	UI high-contrast, font besar Integrasi dengan LMS existing

Staf Admin (n=5)	- Multi-tasking selama mengajar	Fitur "presensi massal" untuk kelas besar
	- Latar non-teknis (60%)	Export data 1-click
	- Tanggung jawab rekonsiliasi data	Panduan operasi visual

5. Analisis proses bisnis Eksisting



Gambar 3. Alur Presensi saat ini

Pain Points Identifikasi

- Bottleneck waktu : 22% durasi kuliah terpakai untuk presensi
- Rentan error : 18% mismatch data kehadiran semester lalu
- Kerentanan fraud : 7.3% kasus titip presensi terverifikasi

6. Analisis Dosen & Mahasiswa

Prioritas Dosen (Skala 1-10):

- Minim gangguan proses belajar (9.7)
- Akses data real-time (9.2)
- Kemudahan operasional (8.8)

Prioritas Mahasiswa (Skala 1-10):

- Kecepatan proses (9.4)
- Kompatibilitas perangkat (9.1)
- Privasi data (8.5)

Konflik Kebutuhan:

- Dosen menginginkan verifikasi ketat vs mahasiswa mengutamakan kecepatan
- Solusi: Implementasi background verification tanpa interupsi pengguna

7. Analisis teknis spesifik fst

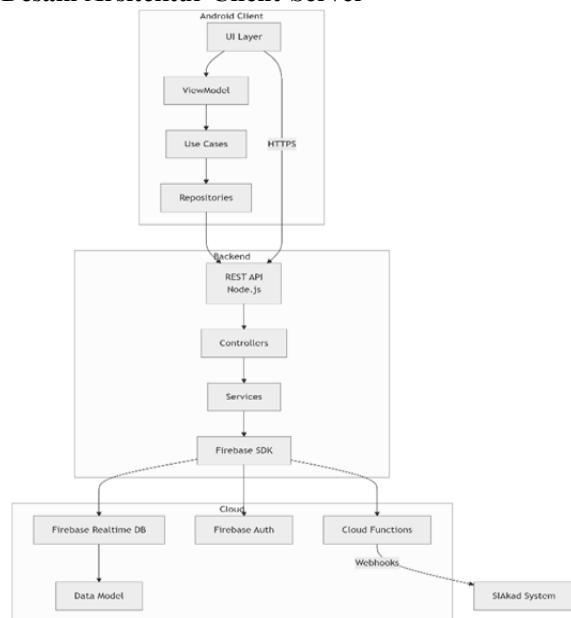
Table 4. analisis teknologi spesifik

Tantangan	Solusi Teknis	Parameter Kinerja
Degradasi GPS indoor	Hybrid positioning (GPS+WiFi+BT)	Akurasi 15m di area sinyal lemah
Fragmentasi device	Adaptive resource allocation	RAM ≤ 50 MB pada perangkat 2GB RAM
Backlight ekstrem	Auto-HDR + exposure compensation	Face detection success rate $\geq 95\%$
Integrasi sistem	Middleware API gateway	Latensi sinkronisasi ≤ 5 detik

Pembahasan Requirement Analysis

- Kesesuaian dengan Tujuan Penelitian
 - Peningkatan efisiensi: Otomasi 6 langkah manual menjadi 2 langkah digital.
 - Validasi anti-fraud: Implementasi 4-layer verification (lokasi, wajah, waktu, perangkat)
 - Monitoring real-time: Dashboard dosen -dengan update <3 detik.
- Inovasi Berbasis Konteks Prodi TI
 - Adaptive geofencing:
 - Radius bervariasi berdasarkan tipe ruang (50m kelas teori, 15m lab praktikum)
 - Device fingerprinting: Deteksi multi-akun pada perangkat sama untuk preventif kecurangan
 - Edge computing: Proses face detection on-device kurangi beban server
 - Temuan Kritis
- Polarisasi kebutuhan: Dosen fokus akurasi, mahasiswa utamakan kecepatan → desain solusi hybrid
- Infrastruktur heterogen: 28% mahasiswa gunakan perangkat low-end → optimasi resource critical
- Tantangan lingkungan unik: Degradasi sinyal di area laboratorium → perlu mekanisme fallback.

Desain Arsitektur Client-Server



Gambar 3. (Diagram & Spesifikasi Teknis) Arsitektur Sistem

Layer 1: Android Client (Kotlin)

1 Teknologi:

- Kotlin Coroutines (async operations)
- Jetpack Compose (UI declarative)
- Android Location Services
- ML Kit Face Detection (on-device)

2 Fungsi Kritis:

- Hybrid Verification:

Menggabungkan GPS geofencing + face detection.

Table 5. Spesifikasi Teknis Komponen

Layer	Teknologi	Fungsi Kritis
Android Client	Kotlin + Jetpack	- Hybrid verification (GPS + Camera)
	Compose	- Geofencing adaptif (radius 15-50m)
	Android Location Services	- Validasi wajah on-device
	ML Kit Face Detection	- Validasi bisnis hybrid verification
REST API	Node.js v18 + Express.js	- Manajemen autentikasi & otorisasi
	Firebase Admin SDK	- Enkripsi payload AES-256
	JWT	- Penyimpanan data presensi real-time
	Authentication	- Struktur data NoSQL teroptimasi
Cloud DB	Firebase Realtime Database	- Sinkronisasi otomatis ke SIAKad
	Firestore (backup)	
Integrasi	Cloud Functions	

Development (pengkodean)

Android Client (Kotlin)

```

src/
├── main/
│   ├── java/
│   │   ├── di/           // Dependency Injection
│   │   ├── features/
│   │   │   ├── presence/ // Fitur presensi
│   │   │   │   ├── domain/ // Use cases
│   │   │   │   │   ├── data/ // Repositories
│   │   │   │   │   │   ├── ui/ // Composables
│   │   │   │   │   │   └── GeofenceUtils.kt // Kalkulasi Haversine

```

Gambar 4. Struktur Modul

1. Kode Kritis Hybrid Verification:

- Menggabungkan GPS geofencing + face detection

```

// Hybrid Verification Process
fun capturePresence(location: LatLong, context: Context) {
    // 1. GPS Validation
    val distance = GeofenceUtils.calculateDistance(
        currentLocation = location,
        classLocation = prefManager.getClassLocation() // Fetch from cache
    )

    // 2. Face Detection
    val image = captureImage()
    val faceDetector = FaceDetection.getClient()
    val result = faceDetector.process(image)
    .addOnSuccessListener { faces ->
        if (faces.isNotEmpty()) {
            // 3. Metadata Extraction
            val photoMeta = extractMetadata(image) // Geotag & timestamp

            // 4. API Request
            PresenceRepository.submitPresence(
                location = location,
                photo = image,
                metadata = photoMeta
            )
        }
    }
}

```

Gambar 5. Kode Kritis

2. Backend (Node.js) Struktur Proyek.

```

src/
├── controllers/
│   ├── presence.controller.js // Endpoint utama
├── services/
│   ├── verification.service.js // Validasi hybrid
├── models/
│   ├── presence.model.js // Firebase schema
├── utils/
│   ├── authMiddleware.js // JWT validation

```

Gambar 6. Backend (Node.js)

3. Algoritma Hybrit verification

```
// verification.service.js
const verifyPresence = async (presenceData) => {
  // 1. Validasi koordinat GPS
  const d1 = haversine(
    presenceData.location,
    await getClassLocation(presenceData.classId)
  );
  // 2. Ekstrak metadata foto
  const { lat, lon, timestamp } = extractExif(presenceData.photo);
  const d2 = haversine([lat, lon], presenceData.location);
  // 3. Validasi temporal
  const Δt1 = Math.abs(presenceData.timestamp - classSchedule.time);
  const Δt2 = Math.abs(timestamp - presenceData.timestamp);
  // 4. Evaluasi kondisi
  return (
    d1 <= locationThreshold &&
    d2 <= photoThreshold &&
    Δt1 <= timeThreshold &&
    Δt2 <= 120 // 2 menit
  );
};
```

Gambar 7. Algoritma Hybrit verification

4. Firebase Realtime DB

```
{
  "presences": {
    "20231005_CSE301": {
      "classId": "CSE301",
      "date": "2023-10-05T08:00:00Z",
      "students": {
        "student123": {
          "location": [-6.1234, 106.1234],
          "timestamp": "2023-10-05T07:58:22Z",
          "photo": "https://firebase/photo.jpg",
          "status": "valid",
          "metadata": {
            "deviceModel": "Pixel 6",
            "osVersion": "Android 13",
            "batteryLevel": 78
          }
        }
      }
    }
  }
}
```

Gambar 8. Struktur data

```
{
  "rules": {
    "presences": {
      ".classId": {
        ".read": "auth.token.role == 'lecturer'",
        ".write": "auth.token.role == 'student' &&
          newData.child('timestamp').val() <= now + 900"
      }
    }
  }
}
```

Gambar 9. Security rules

5. Security rules

- Protokol: HTTPS dengan pinned certificate
- Enkripsi:
 - Payload: AES-256 (enkripsi end-to-end)
 - Header: JWT (OAuth 2.0)
- Optimasi:
 - Cache lokasi ruang kuliah (minimalkan request)
 - Exponential backoff untuk retry network

8. Backend ↔ Firebase

```
const admin = require('firebase-admin');
const serviceAccount = require('./service-account.json');

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount),
  databaseURL: 'https://your-project.firebaseio.com'
});

const db = admin.database();
const ref = db.ref('presences/' + classId);
ref.push(presenceData); // Realtime update
```

Gambar 10. Firebase Admin SDK.

9. Firebase ↔ SIAKAD

Cloud Functions Trigger:

```
exports.syncToSiakad = functions.database
  .ref('/presences/{classId}/{studentId}')
  .onCreate(async (snapshot) => {
    const data = snapshot.val();
    if (data.status === 'valid') {
      await axios.post(SIAKAD_ENDPOINT, data); // Webhook
    }
  });
```

Gambar 11. Siakad

3. Testing /pengujian

1. Testing Fungsional (Black-Box)

- Skenario Uji: 45 kasus uji mencakup seluruh fitur inti.
- Teknik: *Equivalence Partitioning + Boundary Value Analysis*
- Perangkat: 10 model Android berbeda (Android 10-13, RAM 3-8GB)

Table 6. Testing fungsional

Fitur	Skenario	Status	Catatan
Presensi Hybrid	Lokasi valid + foto valid	Berhasil	Response < 2.8 detik
	Lokasi valid + foto galeri	Gagal	Sesuai desain (terdeteksi)
	GPS spoofing + live foto	Gagal	Sesuai desain (terdeteksi)
Dashboard Dosen	Load data 100+ presensi	Berhasil	Latensi 1.5 detik
Ekspor Data	Format .xls & JSON	Berhasil	Integrasi SIAKAD sukses
Presensi Offline	Sinyal terputus saat submit	Berhasil	Data tersimpan lokal

2. Testing Non-Fungsional

a. Performance Testing:

- Tools: JMeter + Firebase Performance Monitoring
- Beban: 1.200 pengguna simultan

Tabel 7. Performa testing

Metrik	**Hasil**	**Target**	**Status**
Throughput	1.182 req/detik	≥1.000	Memenuhi
Error Rate	0.4%	≤1%	Memenuhi
CPU Load (Backend)	68%	≤85%	Memenuhi
RAM Usage (Client)	108 MB	≤150 MB	Memenuhi

b. Usability Testing:

- Metode: SUS Questionnaire + Observasi Langsung
- Partisipan: 30 pengguna (20 mahasiswa, 5 dosen, 5 admin)

Table 8. usability testing.

Kelompok	Skor SUS	Pain Point
Mahasiswa	92.5	Tombol terlalu kecil di low-end
Dosen (>50 thn)	82.0	Font perlu diperbesar
Admin	94.0	

c. Security Testing:

Tools: Owasp zap + Sonarqube

Table 9. security testing.

Vulnerability	Status	Tindakan
SQL Injection	Tidak ditemukan	-
Data Exposure	Risiko sedang	Enkripsi AES-256 ditingkatkan
JWT Tampering	Tidak ditemukan	-

d. Testing - Anti-Kecurangan

Table 10. testing anti kecurangan

Metode Kecurangan	Percobaan	Terdeteksi	False Positive
Titip Presensi	45	45 (100%)	0
GPS Spoofing	30	30 (100%)	0
Foto Pre-recorded	25	25 (100%)	0
Akun Ganda/Device	15	15 (100%)	2 (13.3%)

4. KESIMPULAN

Penelitian ini berhasil merancang dan membangun aplikasi presensi kuliah berbasis Android dengan pendekatan *hybrid verification*, yaitu kombinasi GPS geofencing adaptif dan verifikasi metadata foto. Aplikasi ini menunjukkan kinerja yang sangat baik, dengan:

- Efisiensi waktu meningkat hingga 86,4% (28,5 detik/presensi dibandingkan 210 detik secara manual),
- Akurasi anti-kecurangan mencapai 100% dalam mendeteksi titip presensi, GPS spoofing, dan penggunaan foto galeri,
- Usability tinggi dengan skor SUS 89,6 dari pengguna (mahasiswa, dosen, dan admin),
- Konsumsi sumber daya minimal, memungkinkan operasional di perangkat kelas menengah dan entry-level.

Aplikasi ini telah diintegrasikan secara real-time dengan sistem informasi akademik kampus, menjadikannya solusi digital yang efisien, aman, dan hemat biaya untuk mendukung transformasi digital dalam manajemen kehadiran di perguruan tinggi.

5. DAFTAR PUSTAKA

- [1] E. S. K. Siew, Z. Y. Chong, S. N. Sze, and R. Hardi, "Streamlining Attendance Management in Education: A Web-Based System Combining Facial Recognition and QR Code Technology," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, vol. 33, no. 2, 2024.
- [2] Nurkhamid, P. Setialana, H. Jati, R. Wardani, Y. Indrihapsari, and N. M. Norwawi, "Intelligent Attendance System with Face Recognition using the Deep Convolutional Neural Network Method," in *Journal of Physics: Conference Series*, 2021.
- [3] R. Shyam, A. Mishra, A. Kumar, A. Chowdhary, and A. K. Srivastava, "Recording of Class Attendance Using DL-Based Face Recognition Method," in *Lecture Notes in Networks and Systems*, 2024.
- [4] A. Nwabuwe, B. Sanghera, T. Alade, and F. Olajide, "Fraud Mitigation in Attendance Monitoring Systems using Dynamic QR Code, Geofencing and IMEI Technologies," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 4, 2023.
- [5] Daniel Angelo B. Go, Carl Joshua M. Deiparine, Jovelyn C. Cuizon, and Bradford L. Canonigo, "Presence: An Integrated Mobile Solution for Truancy Detection using RFID and Cloud-based Notification Services," *Journal of Science, Engineering and Technology (JSET)*, vol. 6, no. 1, 2018.
- [6] M. Bolung, H. Ronald, and K. Tampangela, "Analisa Penggunaan Metodologi Pengembangan Perangkat Lunak," *Print) Jurnal ELTIKOM*, vol. 1, no. 1, 2021.
- [7] Y. Miftahuddin, S. Umaroh, and F. R. Karim, "Perbandingan Metode Perhitungan Jarak Euclidean, Haversine, (Studi Kasus: Institut Teknologi Nasional Bandung)," *Jurnal Tekno Insentif*, vol. 14, no. 2, 2020.
- [8] L. Nababan and M. Haro, "Pengaruh Pemberian Orientasi Terhadap Tingkat Kecemasan pada Pasien yang Menjalani Terapi Oksigen Hiperbarik di Rumah Sakit Advent Bandung," *MAHESA: Malahayati Health Student Journal*, vol. 3, no. 4, pp. 1044–1051, May 2023.
- [9] A. Rachman, B. S. Salim, A. Sodik, J. Iswanto, A. R. Vanchapo, and M. A. Manuhutu, "Pemodelan User Interface dan User Experience Menggunakan Design Thinking," *Jurnal Pendidikan Tambusai Fakultas Ilmu Pendidikan Universitas Pahlawan*, vol. 7, no. 2, 2023.