

# **Pengembangan Sistem *Monitoring* Kehadiran Dosen Real-Time Berbasis Web Menggunakan Metode Rapid Application Development**

**Devytha Nur Alfi\*<sup>1</sup>, Davina Yela Gainta<sup>2</sup>, Dewi Oktafiani<sup>3</sup>**

<sup>123</sup>Prodi S1 Informatika, <sup>123</sup>STMIK Amikom Surakarta,

<sup>123</sup>Sukoharjo, Indonesia

Email: <sup>1</sup>[devytha.10452@mhs.amikomsolo.ac.id](mailto:devytha.10452@mhs.amikomsolo.ac.id),

<sup>2</sup>[davina.10442@mhs.amikomsolo.ac.id](mailto:davina.10442@mhs.amikomsolo.ac.id), <sup>3</sup>[dewioktafiani@dosen.amikomsolo.ac.id](mailto:dewioktafiani@dosen.amikomsolo.ac.id)

## **Abstract**

*Students at STMIK Amikom Surakarta often struggle to meet lecturers because they do not know the lecturer's attendance schedule in real-time. This research develops a real-time lecturer attendance monitoring system with automatic notifications to improve academic transparency. The system is built using Next.js for display, Laravel for backend, and MySQL for data storage. Main features include real-time attendance recording, automatic notifications via Firebase Cloud Messaging, integration of offline and online class schedules, and role-based user dashboards. Testing shows all features work well (100%). Evaluation from 61 users (24 lecturers, 35 students, 2 department heads) resulted in a satisfaction score of 4.3 out of 5.0 with System Usability Scale (SUS) reaching 88 out of 100 (grade A). The system successfully reduces the time spent searching for lecturers by 75% and increases academic transparency because 92% of students can easily find out the lecturer's attendance status in real-time and 100% of notifications are delivered properly.*

**Keywords:** attendance monitoring system, automatic notification, real-time system, academic transparency, Rapid Application Development

## **Abstraksi**

*Mahasiswa di STMIK Amikom Surakarta sering kesulitan menemui dosen karena tidak mengetahui jadwal kehadiran dosen secara real-time. Penelitian ini mengembangkan sistem monitoring kehadiran dosen real-time dengan notifikasi otomatis untuk meningkatkan transparansi akademik. Sistem dibangun menggunakan Next.js untuk tampilan, Laravel untuk backend, dan MySQL untuk penyimpanan data. Fitur utama mencakup pencatatan kehadiran real-time, notifikasi otomatis melalui Firebase Cloud Messaging, integrasi jadwal kelas offline dan online, serta dashboard berbasis peran pengguna. Pengujian menunjukkan semua fitur berfungsi dengan baik (100%). Evaluasi dari 61 pengguna (24 dosen, 35 mahasiswa, 2 kepala departemen) menghasilkan skor kepuasan 4,3 dari 5,0 dengan System Usability Scale (SUS) mencapai 88 dari 100 (grade A). Sistem berhasil mengurangi waktu pencarian dosen sebesar 75% dan meningkatkan transparansi akademik karena 92% mahasiswa dapat dengan mudah mengetahui status kehadiran dosen secara real-time dan 100% notifikasi tersampaikan dengan baik..*

**Kata Kunci:** sistem monitoring kehadiran, notifikasi otomatis, real-time system, transparansi akademik, Rapid Application Development

## 1. PENDAHULUAN

Transparansi dan efisiensi dalam proses akademik menjadi kebutuhan penting di era digital [1]. Di STMIK Amikom Surakarta dengan 24 dosen aktif dan 813 mahasiswa, terdapat masalah yang sering dialami mahasiswa yaitu kesulitan menemui dosen di luar jam kuliah. Mahasiswa tidak mengetahui apakah dosen sedang ada di kampus atau tidak, sehingga berdampak negatif pada komunikasi akademik, bimbingan, dan konsultasi.

Berdasarkan survei awal terhadap 100 mahasiswa, sebanyak 78% mengalami kesulitan menemui dosen karena tidak tahu status kehadiran dosen. Mahasiswa rata-rata melakukan 3-4 kunjungan gagal per minggu, masing-masing memerlukan waktu sekitar 45 menit. Hal ini menyebabkan pemborosan waktu dan biaya transportasi [2]. Sistem pencatatan kehadiran yang masih manual tidak dapat memberikan informasi langsung kepada mahasiswa [3].

Penelitian sebelumnya menunjukkan bahwa sistem informasi berbasis web dapat meningkatkan efisiensi pengelolaan akademik [4]. Namun, sebagian besar sistem yang ada masih bersifat manual atau semi-otomatis, belum memiliki fitur notifikasi *real-time*, dan belum memenuhi kebutuhan transparansi yang melibatkan mahasiswa, dosen, dan manajemen departemen secara bersamaan [5].

Penelitian ini bertujuan mengembangkan sistem *monitoring* kehadiran dosen *real-time* dengan notifikasi otomatis untuk meningkatkan transparansi akademik di STMIK Amikom Surakarta. Penelitian menjawab dua pertanyaan:

1. Bagaimana merancang sistem *monitoring* kehadiran *real-time* yang dapat meningkatkan transparansi akademik?
2. Seberapa besar peningkatan efisiensi waktu dan transparansi akademik yang dapat dihasilkan?

Kontribusi penelitian ini mencakup: (1) sistem *monitoring* kehadiran *real-time* dengan notifikasi otomatis, (2) *dashboard* khusus untuk berbagai peran pengguna, (3) integrasi jadwal kuliah luring dan daring dalam satu platform, dan (4) sistem persetujuan izin dosen secara digital.

## 2. TINJAUAN PUSTAKA

### 2.1. Sistem Informasi Kehadiran Berbasis Web

Sistem informasi kehadiran adalah aplikasi untuk mencatat, mengelola, dan melaporkan data kehadiran secara terstruktur [4]. Penelitian sebelumnya menunjukkan sistem kehadiran berbasis web dapat mengurangi waktu pencatatan hingga 91%, menurunkan kesalahan dari 5% menjadi 0,2%, dan menghemat biaya operasional hingga 60% [6]. Sistem absensi berbasis web dan *mobile* menggunakan *QR Code* telah dikembangkan, namun belum memiliki fitur *monitoring real-time* dan notifikasi otomatis [7]. Jonathan (2024) merancang sistem kehadiran berbasis web dan mobile

yang terintegrasi, namun sistem tersebut masih memerlukan pengembangan lebih lanjut dalam hal fitur monitoring real-time untuk meningkatkan transparansi [8].

## 2.2. Teknologi *Real-time Monitoring*

*Real-time monitoring* adalah kemampuan sistem menampilkan informasi terkini tanpa jeda waktu yang berarti [11]. Sumarsono (2025) mengembangkan sistem absensi berbasis IoT-RFID dengan *monitoring real-time* yang memperbarui setiap 5 detik dengan akurasi 99,5% [6]. Inayah dan Maburi (2024) mengembangkan sistem dengan notifikasi otomatis melalui Telegram Bot yang meningkatkan kesadaran pengguna hingga 85% [4]. *Firestore Cloud Messaging* (FCM) banyak digunakan karena mendukung berbagai platform dan memiliki performa yang baik [6].

## 2.3. Sistem Notifikasi Otomatis

Sistem notifikasi otomatis penting untuk meningkatkan kecepatan respon dan keterlibatan pengguna [10]. *Firestore Cloud Messaging* (FCM) banyak digunakan karena mendukung berbagai platform, dapat diandalkan, dan memiliki performa baik [6].

## 2.4. Transparansi Akademik

Transparansi akademik mencakup keterbukaan informasi tentang proses pembelajaran, kehadiran, penilaian, dan aktivitas akademik lainnya [1]. Asyiyifaurrohman dan Mubin (2025) mengembangkan sistem absensi dosen dengan validasi waktu otomatis yang berhasil mengurangi waktu proses administratif sebesar 62% dan meningkatkan akurasi data hingga 100% [9].

## 2.5. Metode *Rapid Application Development* (RAD)

*Rapid Application Development* (RAD) adalah metodologi pengembangan sistem yang menekankan siklus pengembangan yang singkat dengan melibatkan pengguna secara langsung melalui prototipe [17]. Menurut Kendall dan Kendall (2011), RAD memiliki empat tahap utama yaitu perencanaan kebutuhan, *workshop* desain RAD, implementasi, dan *cutover* [17]. McConnell (1996) juga menegaskan bahwa pendekatan pengembangan cepat seperti RAD dapat mempercepat penyelesaian proyek tanpa mengurangi kualitas, selama komunikasi dengan pengguna berjalan baik dan iterasi dilakukan secara terstruktur [20]. Fauzi et al. (2023) menemukan bahwa RAD dapat menghemat waktu pengembangan antara 21 hingga 120 hari tergantung ukuran proyek dan sangat cocok untuk sistem informasi berskala kecil hingga menengah [18].

## 2.6. *Technology Acceptance Model* (TAM)

*Technology Acceptance Model* (TAM) yang dikembangkan oleh Davis (1989) adalah model yang digunakan untuk mengukur seberapa mudah pengguna menerima dan menggunakan teknologi informasi [15]. Model ini mengidentifikasi dua faktor utama yaitu persepsi tentang manfaat dan persepsi tentang kemudahan penggunaan [15]. Rosli et al. (2022) menemukan bahwa TAM tetap menjadi model yang relevan untuk penelitian teknologi informasi di lingkungan pendidikan [12].

### 2.7. System Usability Scale (SUS) untuk Pengukuran Usability

System Usability Scale (SUS) adalah instrumen pengukuran kegunaan yang telah teruji secara internasional dan terdiri dari 10 pertanyaan dengan skala Likert 1-5 [19]. Deshmukh dan Chalmata (2024) memvalidasi SUS dan menemukan bahwa skor SUS di atas 80 diklasifikasikan sebagai "good" dan mencerminkan pengalaman pengguna di atas rata-rata [19].

## 3. METODE PENELITIAN

Penelitian ini menggunakan metode *Rapid Application Development* (RAD) yang menekankan pengembangan sistem secara bertahap melalui prototipe yang melibatkan pengguna langsung [13], [14], [17]. RAD dipilih karena mampu mempercepat pengembangan dengan tetap menjaga kualitas perangkat lunak melalui iterasi desain dan prototipe [13]. Seperti yang dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian dengan Metode RAD

### 3.1. Pemilihan Responden

Penelitian menggunakan *purposive sampling* dengan kriteria: (a) dosen memiliki pengalaman kerja minimal 2 tahun dan mengajar minimal 2 mata kuliah; (b) mahasiswa aktif belajar di semester 3-8 dan pernah mengalami kesulitan menemui dosen; (c) kepala akademik bekerja di bagian administrasi akademik.

### 3.2. Perencanaan Kebutuhan (*Requirements Planning*)

Tahap pertama dimulai dengan mengidentifikasi masalah-masalah yang dihadapi dalam proses *monitoring* kehadiran dosen. Proses ini melibatkan wawancara dengan 24 dosen, 35 mahasiswa, dan 2 kepala akademik, serta pengamatan alur kerja. Pengumpulan dokumen prosedur kehadiran dan survei 100 mahasiswa juga dilakukan.

### 3.3. Desain RAD (*RAD Design Workshop*)

Perancangan sistem dilakukan secara bertahap mencakup desain basis data, alur proses, dan antarmuka pengguna. Sistem dirancang dengan arsitektur tiga lapisan menggunakan *RESTful API*. Basis data terdiri dari 9 tabel utama. Pada tahap ini dibuat *Entity Relationship Diagram* (ERD), *Use Case Diagram*, dan desain antarmuka menggunakan Figma.

### 3.4. Implementasi (*Implementation*)

Pengembangan menggunakan *Next.js 14*, *TypeScript*, *Tailwind CSS*, *Firebase Cloud Messaging* untuk *frontend*; *Laravel 10*, *PHP 8.2*, *RESTful API*, *JWT Authentication* untuk *backend*; dan *MySQL 8.0* untuk basis data.

### 3.5. Cutover (Pengujian dan Implementasi)

Sistem diuji menggunakan *black-box testing* yang fokus pada fungsionalitas. Pengujian mencakup *unit testing*, *integration testing*, *system testing*, dan *User Acceptance Testing*. Kepuasan pengguna diukur menggunakan kuesioner SUS selama 1 bulan.

### 3.6. Pengumpulan Data Kepuasan Pengguna

Kepuasan pengguna dikumpulkan menggunakan kuesioner *System Usability Scale* (SUS) yang telah teruji secara internasional. Uji reliabilitas menggunakan *Cronbach's Alpha* menghasilkan nilai 0,87 yang menunjukkan instrumen dapat diandalkan. SUS terdiri dari 10 pertanyaan dengan skala Likert 1-5 yang memungkinkan pengukuran *Usability* secara menyeluruh. Proses evaluasi dilakukan selama 1 bulan untuk mengukur seberapa efisien sistem dalam menghemat waktu dan meningkatkan transparansi akademik.

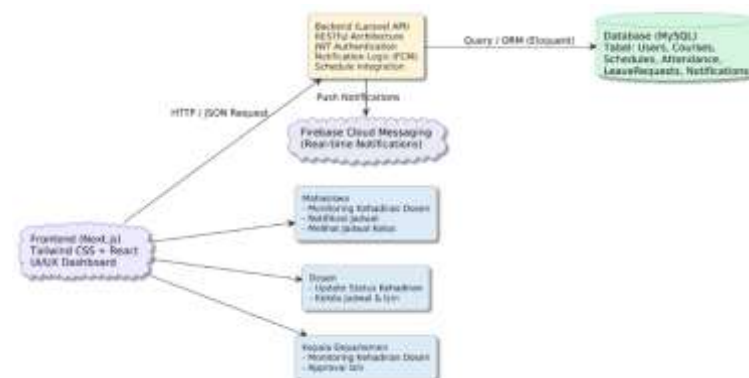
## 4. HASIL DAN PEMBAHASAN

### 4.1. Hasil Perencanaan Kebutuhan

Hasil wawancara dan survei menunjukkan prioritas kebutuhan tertinggi adalah *monitoring* status kehadiran dosen *real-time* (95%), notifikasi otomatis (88%), dan *dashboard* mahasiswa (92%), menunjukkan urgensi tinggi untuk digitalisasi proses *monitoring* kehadiran.

### 4.2. Hasil Desain RAD

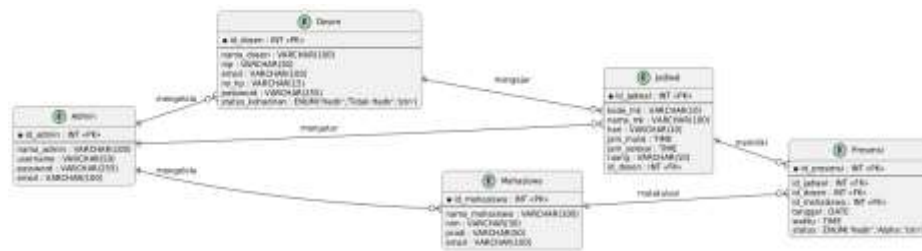
#### 4.2.1. Arsitektur Sistem



Gambar 2. Struktur Arsitektur Sistem *Monitoring* Kehadiran Dosen

Sistem dirancang dengan tiga lapisan membedakan antara tampilan, logika aplikasi, dan penyimpanan data. Komunikasi antar bagian menggunakan *RESTful API* dengan format *JSON*. *Next.js* membangun antarmuka responsif, *Laravel* mengelola logika bisnis, dan *MySQL* menyimpan data.

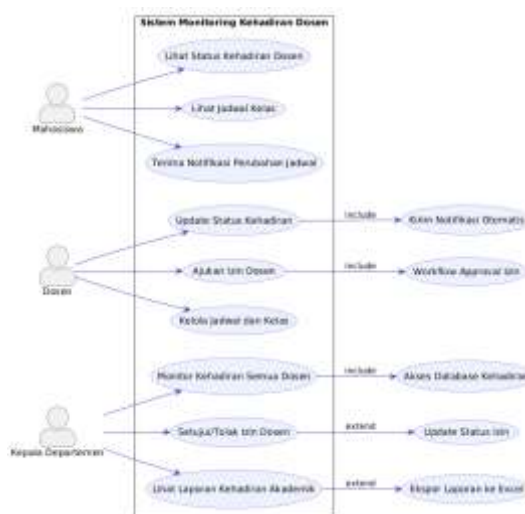
#### 4.2.2. Entity Relationship Diagram (ERD)



Gambar 3. Entity Relationship Diagram (ERD)

ERD menunjukkan hubungan antar data dalam *database* sistem. Tabel *users* menyimpan data pengguna dan terhubung dengan tabel *roles* untuk menentukan hak akses. Tabel *courses* terhubung dengan *schedules* untuk menyimpan jadwal perkuliahan. Tabel *dosen\_attendance* mencatat kehadiran dosen, sedangkan *attendance* mencatat kehadiran mahasiswa. Struktur *database* ini disesuaikan berdasarkan *workshop* desain RAD yang melibatkan semua *stakeholder*.

#### 4.2.3. Use Case Diagram

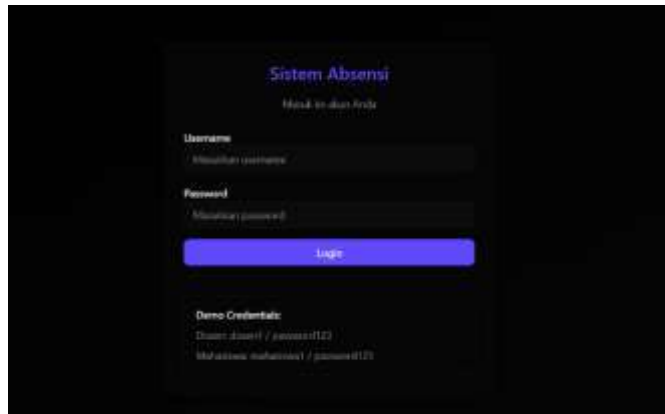


Gambar 4. Use Case Diagram

*Use Case Diagram* menggambarkan interaksi antara tiga aktor utama dengan sistem. Dosen dapat mengubah status kehadiran, mengelola jadwal, mengajukan izin, dan melihat riwayat kehadiran mereka. Mahasiswa dapat melihat status kehadiran dosen, jadwal kuliah, *link meeting online*, dan mencatat kehadiran mereka sendiri. Kepala Departemen dapat memonitor kehadiran semua dosen, menyetujui atau menolak permohonan izin, dan mengakses laporan kehadiran.

### 4.3. Hasil Implementasi

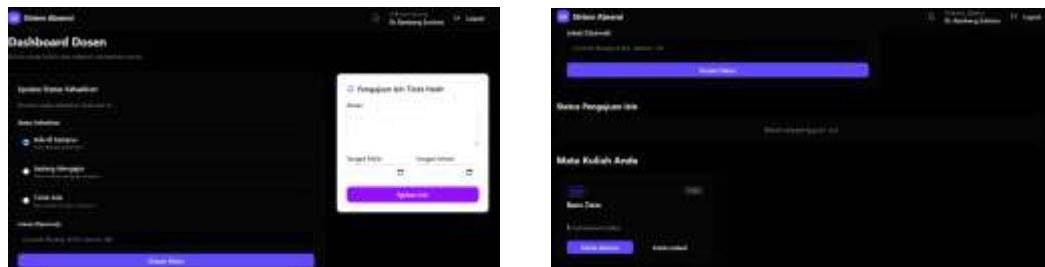
#### 4.3.1. Sistem Autentikasi dan Login



Gambar 5. Halaman *Login* Sistem

Sistem menerapkan sistem keamanan *Login* berbasis peran dengan tiga tingkat akses pengguna menggunakan JWT (*JSON Web Token*). Ketika pengguna berhasil *Login*, sistem akan mengarahkan mereka ke *dashboard* yang sesuai dengan peran mereka. Sistem menerapkan kontrol akses berbasis peran (RBAC) yang membatasi akses fitur berdasarkan peran masing-masing pengguna. Untuk memastikan keamanan akses Kepala Departemen, sistem melakukan pengecekan ganda dengan melihat *field departmentId* dan *headId* di tabel *users*.

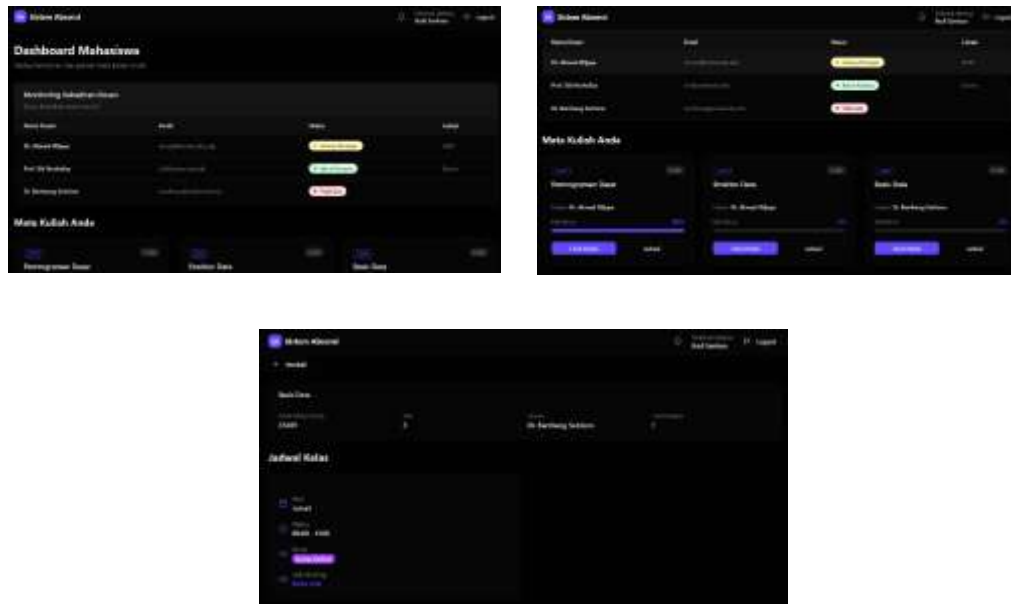
#### 4.3.2. Dashboard Dosen



Gambar 6. *Dashboard* Dosen

*Dashboard* dosen menampilkan *form Update* status kehadiran dengan tiga pilihan (Ada di Kampus, Sedang Mengajar, Tidak Ada), daftar mata kuliah yang diampu beserta jumlah mahasiswa, riwayat pengajuan izin dengan status persetujuan, dan tombol untuk mengelola absensi mahasiswa serta jadwal kelas. Implementasi menggunakan *React Server Components* dan Tailwind CSS untuk tampilan yang responsif.

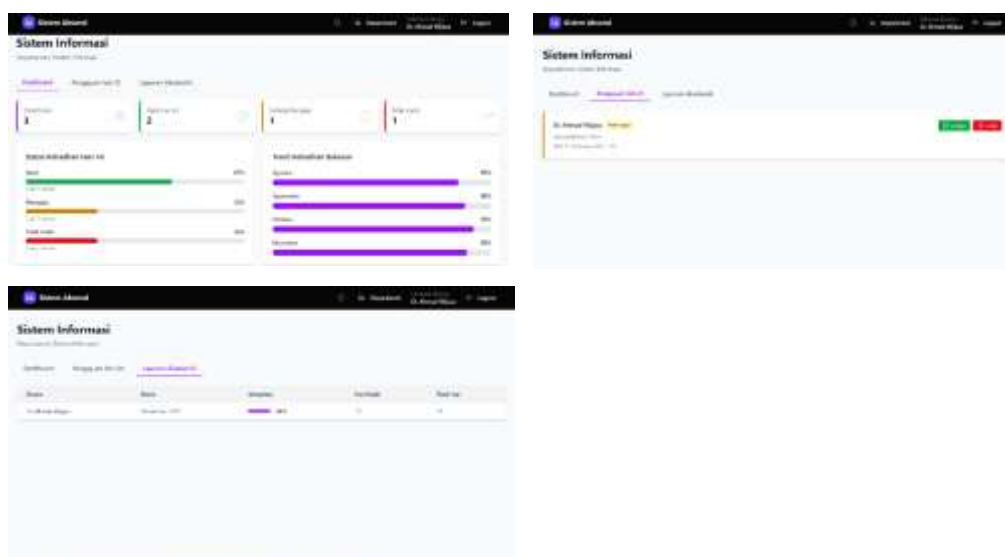
#### 4.3.3. Dashboard Mahasiswa



Gambar 7. Dashboard Mahasiswa

Dashboard mahasiswa menampilkan *monitoring* kehadiran dosen dengan sistem indikator warna visual (hijau untuk dosen yang ada di kampus, kuning untuk sedang mengajar, merah untuk tidak ada), daftar mata kuliah dengan persentase kehadiran, informasi mode kelas (*offline* dengan ruangan atau *online* dengan *link meeting*), serta jadwal perkuliahan lengkap.

#### 4.3.4. Dashboard Kepala Departemen

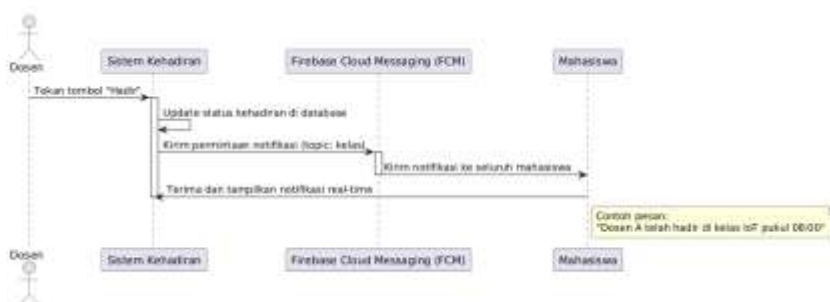


Gambar 8. Dashboard Kepala Departemen

*Dashboard* kepala departemen terdiri dari 3 tab utama: Tab *Dashboard* menampilkan statistik kehadiran dosen dengan grafik dan tabel *monitoring real-time*, Tab Pengajuan Izin menampilkan daftar pengajuan dengan tombol setuju/tolak, Tab Laporan Akademik menampilkan analisis kehadiran bulanan dengan fitur *export* Excel.

#### 4.3.5. Sistem Notifikasi Otomatis

Fitur notifikasi otomatis dikembangkan menggunakan *Firebase Cloud Messaging* (FCM). Sistem akan mengirimkan pesan ke perangkat pengguna saat terjadi perubahan status kehadiran. Misalnya, ketika dosen menekan tombol “Hadir”, sistem secara otomatis mengirimkan notifikasi ke seluruh mahasiswa di kelas tersebut.



Gambar 9. Alur Notifikasi Otomatis dalam Sistem

## 4.4. Hasil Pengujian Sistem

### 4.4.1. Pengujian Fungsionalitas

Pengujian *black-box* terhadap 8 skenario menunjukkan tingkat keberhasilan 100% untuk semua fitur utama:

Tabel 1. Hasil Pengujian Fungsional Sistem

No	Fitur	Skenario Pengujian	Hasil
1	<i>Authentication</i>	Login dengan kredensial valid	Berhasil
2	<i>Role-Based Access</i>	Validasi akses <i>dashboard</i> sesuai <i>role</i>	Berhasil
3	<i>Update Status</i>	Update status kehadiran dosen	Berhasil
4	<i>Monitoring</i>	Monitoring kehadiran <i>real-time</i>	Berhasil
5	Notifikasi	Pengiriman notifikasi otomatis	Berhasil
6	Jadwal	Pengelolaan jadwal luring/daring	Berhasil
7	Persetujuan Izin	Proses <i>approval</i> izin dosen	Berhasil
8	<i>Department Access</i>	Validasi kepala departemen	Berhasil

#### 4.4.2. Evaluasi Kepuasan Pengguna

Evaluasi 67 responden menggunakan SUS menghasilkan skor rata-rata 4,3 dari 5,0 dengan rincian: kemudahan penggunaan 4,3, kecepatan akses 4,2, akurasi informasi 4,5, kegunaan fitur 4,4, dan tampilan antarmuka 4,1. Skor SUS mencapai 88 dari 100 (*grade A*), menunjukkan tingkat kegunaan yang sangat baik dan lebih tinggi dari penelitian Inayah dan Maburi (2024) yang mencapai 82% [4]. Menurut Deshmukh dan Chalmata (2024), skor SUS di atas 80 diklasifikasikan sebagai "*good*" dan mencerminkan pengalaman pengguna yang di atas rata-rata [19].

#### 4.4.3. Analisis Efisiensi dan Transparansi

Sistem berhasil mengurangi waktu kunjungan dari rata-rata 45 menit menjadi 0 menit, dengan pengurangan frekuensi kunjungan gagal dari 3 dari 4 kunjungan menjadi 1 dari 10 kunjungan, menghasilkan peningkatan efisiensi sebesar 75%. Transparansi akademik meningkat dengan 92% mahasiswa mudah mengakses informasi kehadiran dosen *real-time*, 100% notifikasi tersampaikan dengan baik, dan kepala departemen dapat memonitor dengan pembaruan setiap 5 detik.

#### 4.5. Pembahasan

Penerapan metode RAD terbukti efektif mempercepat pengembangan sambil menjaga kualitas produk [17]. Keterlibatan pengguna langsung dalam setiap tahap RAD memastikan sistem sesuai kebutuhan nyata di lapangan [14]. Penelitian menunjukkan RAD dapat menghemat waktu pengembangan dan sangat cocok untuk sistem informasi berskala kecil hingga menengah [18].

Tingginya kepuasan pengguna (skor 88) dapat dijelaskan dari: (1) desain sistem sederhana dan intuitif membuat pengguna memahami cara penggunaan tanpa pelatihan rumit, (2) informasi *real-time* yang akurat memungkinkan pengguna membuat keputusan dengan percaya diri, dan (3) fitur notifikasi otomatis memberikan pengalaman pengguna lebih baik karena informasi datang otomatis ke perangkat pengguna.

Implementasi keamanan *Login* berbasis peran dengan JWT memberikan perlindungan memadai dan fleksibel [15]. FCM menunjukkan keandalan 100% dalam pengiriman notifikasi, sejalan dengan prinsip sistem *real-time* yang baik [16]. *Dashboard* khusus untuk setiap peran memungkinkan setiap pengguna mendapatkan informasi relevan dengan kebutuhan mereka, meningkatkan rasa tanggung jawab dosen dan memudahkan pengambilan keputusan berdasarkan data akurat.

Sistem dapat diadopsi institusi pendidikan lain dengan penyesuaian sesuai struktur organisasi. Keberhasilan implementasi menunjukkan pendekatan RAD yang melibatkan pengguna sejak awal dapat menghasilkan sistem efektif dan diterima baik oleh semua pihak.

Keterbatasan penelitian mencakup: dilakukan hanya di satu institusi, periode evaluasi singkat (1 bulan), sistem belum memiliki fitur prediksi kehadiran menggunakan *machine learning*, dan validasi lokasi belum seakurat sistem *GPS*.

## 5. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem *monitoring* kehadiran dosen *real-time* menggunakan metode *Rapid Application Development* (RAD). Pengujian menunjukkan semua fitur berfungsi dengan baik (100% berhasil). Evaluasi pengguna menunjukkan kepuasan yang sangat baik dengan skor rata-rata 4,3 dari 5,0 dan skor SUS mencapai 88 dari 100 (*grade A*).

Sistem berhasil mencapai tujuan utama yaitu meningkatkan efisiensi waktu mahasiswa dalam mencari dosen sebesar 75% dan meningkatkan transparansi akademik dengan 92% mahasiswa dapat dengan mudah mengakses informasi status kehadiran dosen secara *real-time*, serta 100% notifikasi tersampaikan dengan baik.

Implementasi sistem keamanan dengan JWT dan *Role-Based Access Control* (RBAC) memberikan perlindungan data yang memadai. Notifikasi otomatis menggunakan *Firebase Cloud Messaging* menunjukkan keandalan 100% dalam pengiriman pesan. Pendekatan RAD yang melibatkan pengguna sejak tahap awal memungkinkan penyesuaian sistem secara cepat berdasarkan masukan pengguna.

Untuk penelitian lebih lanjut, disarankan untuk: (1) mengembangkan fitur prediksi kehadiran dosen menggunakan *machine learning* untuk membantu perencanaan akademik, dan (2) menambahkan fitur validasi lokasi menggunakan *GPS* untuk membuat pencatatan kehadiran lebih akurat dan aman.

## DAFTAR PUSTAKA

- [1] H. El Mrabet and A. A. Moussa, "IoT-School attendance system using RFID technology," *Int. J. Interact. Mob. Technol.*, vol. 14, no. 14, pp. 95–108, 2020, doi: 10.3991/ijim.v14i14.14625.
- [2] M. Al Imran, M. Fikry, and S. Retno, "Enhancing academic security with RFID-based smart locks and real-time attendance tracking system," in *Proc. 2nd Int. Conf. Multidisciplinary Engineering*, vol. 2, 2024, pp. 1–12.
- [3] T. W. Chiang et al., "Development and evaluation of an attendance tracking system using smartphones with GPS and NFC," *Appl. Artif. Intell.*, vol. 36, no. 1, pp. 1–18, 2022, doi: 10.1080/08839514.2022.2083796.
- [4] S. Inayah and A. H. Maburri, "Real-time attendance and discipline monitoring through EvoCard," in *Proc. Int. Conf. Educ. Soc. Humanity*, vol. 2, no. 2, 2024, pp. 586–592.
- [5] S. D. Jadhav, P. L. Chavan, A. H. Kamble, S. T. Lad, and S. S. Patil, "A modern web-based student attendance management system," *Int. J. Innov. Res. Technol.*, vol. 11, no. 10, pp. 2793–2797, Mar. 2025.
- [6] Sumarsono and A. Mayasari, "Real-time Attendance and Security Monitoring System Using IoT-RFID-Webserver-Android: A low-cost solution," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 15, no. 3, pp. 764–779, 2025, doi: 10.18517/ijaseit.15.3. 20294.
- [7] M. R. Wayahdi and F. A. Dzikri, "Integrating facial recognition and GPS technology for efficient attendance management in educational institutions," in *Proc. Int. Conf. Electr. Eng. Informatics*, vol. 1, no. 2, 2024, pp. 85–93.

- [8] R. A. Z. Jonathan, "Case study of attendance system design based on web and mobile," *J. La Multiapp*, vol. 5, no. 5, pp. 759–785, 2024, doi: 10.37899/journallamultiapp.v5i5.1663.
- [9] F. Asyifaurohmah, A. F. Mubin, and A. Tabrani, "Development of a web-based teacher attendance system in government agencies: A case study at the Ministry of Religious Affairs," *Intechno J. Inf. Technol. J.*, vol. 7, no. 1, pp. 24–33, Jul. 2025, doi: 10.24076/intechnojournal.2025v7i1.2233.
- [10] K. Ishaq and S. Bibi, "IoT based smart attendance system using RFID: A systematic literature review," *arXiv preprint arXiv:2308.02591*, 2023,
- [11] S. Nawale, P. Sonawane, H. Tilekar, N. Kharde, and O. Vikhe, "Real time attendance monitoring system," *Int. J. Innov. Res. Multidisciplinary Perspect. Sci.*, vol. 12, no. 3, pp. 1–6, 2024.
- [12] M. S. Rosli, N. S. Saleh, A. Md. Ali, S. Abu Bakar, and L. M. Tahir, "A systematic review of the Technology Acceptance Model for the sustainability of higher education during the COVID-19 pandemic and identified research gaps," *Sustainability*, vol. 14, no. 18, art. 11389, Sep. 2022, doi: 10.3390/su141811389.
- [13] A. Rosyidi, I. A. Saputro, and F. S. Nugraha, "Rancang bangun sistem invoice digital CV Putra Alami Sejahtera menggunakan metode Rapid Application Development," *JSI (Jurnal Sistem Informasi) Universitas Dirgantara Marsekal Suryadarma*, vol. 12, no. 1, pp. 156–163, Jul. 2025, doi: 10.35968/jsi.v12i2.1542.
- [14] R. Baqi, P. Choirina, U. M. Jannah, and K. Kunci, "Rancang bangun sistem informasi manajemen administrasi PKL dan skripsi menggunakan metode RAD," in *Seminar Nasional Sistem Informasi*, 2022, pp. 3423–3432.
- [15] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Q. Manage. Inf. Syst.*, vol. 13, no. 3, pp. 319–340, Sep. 1989, doi: 10.2307/249008.
- [16] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment," *J. Assoc. Comput. Mach.*, vol. 20, no. 1, pp. 46–61, Jan. 1973, doi: 10.1145/321738.321743.
- [17] K. E. Kendall and J. E. Kendall, *Systems Analysis and Design*, 9th ed. Boston, MA, USA: Pearson, 2011.
- [18] M. A. Fauzi, H. Tribiakto, A. Moniva, F. Amir, I. K. Ilyas, and E. Utami, "Systematic literature reviews on Rapid Application Development Information System," *Bull. Comput. Sci. Electr. Eng.*, vol. 4, no. 1, pp. 57–64, Jun. 2023, doi: 10.25008/bcsee.v4i1.1181.
- [19] A. M. Deshmukh and R. Chalmeta, "Validation of System Usability Scale as a Usability metric to evaluate voice user interfaces," *PeerJ Comput. Sci.*, vol. 10, art. e1918, Feb. 2024, doi: 10.7717/peerj-cs.1918.
- [20] S. McConnell, *Rapid Development: Taming Wild Software Schedules*. Redmond, WA, USA: Microsoft Press, 1996.