EXPERT JURNAL MANAJEMEN SISTEM INFORMASI DAN TEKNOLOGI

Pengembangan Sistem Rekapitulasi Presensi Terpusat dengan Sinkronisasi Otomatis dari Perangkat Fingerprint Terdistribusi

Dwi Romadhan[1*], **Dian Resha Agustina**[2]
[1] Sistem Informasi, Fakultas Ilmu Komputer, Universitas Bandar Lampung, Bandar Lampung, Indonesia
[2] Informatika, Fakultas Ilmu Komputer, Universitas Bandar Lampung, Bandar Lampung, Indonesia
[1*]dwi.romadhan@ubl.ac.id, [2]dian.resha@ubl.ac.id

ABSTRACT – Decentralized attendance systems lead to inefficiencies in recapitulation time, potential data fraud, and difficulties in consolidation across multiple locations. The novelty of this research lies in an automatic synchronization architecture that integrates multiple Solution fingerprint devices through the /iWsService API into a centralized web-based system, which has not been extensively studied in similar research. This study aims to develop a centralized attendance recapitulation system with automatic synchronization and measure its effectiveness. The method used is the Waterfall model. Time efficiency was measured using the formula *(Tmanual − Tsystem) / Tmanual × 100%*, where *Tmanual* = 8 hours and *Tsystem* = 2 hours. Fingerprint accuracy was calculated based on the ratio of correct identifications to total trials, from 5,000 trials (100 users × 5 devices × 10 attempts). Synchronization time was measured through timestamp recording from data transmission to database storage. The test results show an accuracy of 98.5%, an average synchronization time of 5.7 seconds (at distances of 5–25 m), and a success rate of 99.6%. The scientific contributions of this research are realized through the design of a centralized integration architecture for distributed fingerprint devices, the implementation of an automatic synchronization mechanism that operates without manual intervention, and quantitative verification showing a 75% improvement in recapitulation efficiency compared to manual methods.

Keywords: Automatic Synchronization; Centralized attendance system; Distributed fingerprint; iWsService API

ABSTRAK – Sistem presensi terdesentralisasi menyebabkan inefisiensi waktu rekapitulasi, potensi kecurangan data, dan kesulitan konsolidasi antar lokasi. Kebaruan penelitian ini adalah arsitektur sinkronisasi otomatis yang mengintegrasikan multiple perangkat fingerprint Solution melalui API /iWsService ke dalam sistem terpusat berbasis web, yang belum banyak dikaji pada penelitian sejenis. Penelitian bertujuan mengembangkan sistem rekapitulasi presensi terpusat dengan sinkronisasi otomatis serta mengukur efektivitasnya. Metode yang digunakan adalah Waterfall. Pengukuran efisiensi waktu menggunakan rumus *(Tmanual − Tsistem) / Tmanual × 100%*, dengan *Tmanual* = 8 jam dan *Tsistem* = 2 jam. Akurasi fingerprint diukur berdasarkan rasio identifikasi benar terhadap total percobaan, dari 5.000 percobaan (100 pengguna × 5 perangkat × 10 kali). Waktu sinkronisasi diukur melalui pencatatan *timestamp* dari pengiriman hingga penyimpanan data. Hasil pengujian menunjukkan akurasi 98,5%, waktu sinkronisasi rata-rata 5,7 detik (jarak 5–25 m), dan *success rate* 99,6%. Kontribusi ilmiah penelitian ini diwujudkan melalui perancangan arsitektur integrasi terpusat untuk perangkat fingerprint terdistribusi, implementasi mekanisme sinkronisasi otomatis yang berjalan tanpa intervensi manual, serta pengukuhan kuantitatif berupa peningkatan efisiensi rekapitulasi sebesar 75% dibandingkan metode manual.

Kata Kunci: API iWsService; Fingerprint terdistribusi; Sistem presensi terpusat; Sinkronisasi otomatis.

1. PENDAHULUAN

Sistem presensi merupakan prosedur fundamental yang wajib dilaksanakan oleh seluruh pegawai dalam suatu organisasi sebagai tolok ukur penilaian kedisiplinan dan produktivitas kerja. Presensi yang akurat menjadi dasar penting dalam pengambilan keputusan manajemen, mulai dari perhitungan upah hingga evaluasi kinerja karyawan. Namun, sistem presensi yang masih dilakukan secara manual atau terdesentralisasi menimbulkan berbagai permasalahan serius, seperti ketidakefektifan waktu dalam proses rekapitulasi, tingginya potensi kecurangan data, serta kesulitan dalam mengonsolidasikan data kehadiran yang berasal dari berbagai lokasi yang tersebar[1].

Perkembangan teknologi informasi telah mendorong berbagai inovasi untuk mengatasi permasalahan presensi melalui pemanfaatan teknologi biometrik dan Internet of

This work is licensed under a Creative Commons Attribution 4.0 International License

Things (IoT). Kusuma dkk. (2024) mengembangkan sistem presensi biometrik sidik jari berbasis IoT menggunakan Arduino Node MCU yang berhasil mengimplementasikan sinkronisasi data dengan waktu rata-rata 8 detik. Penelitian tersebut menunjukkan bahwa teknologi biometrik mampu meningkatkan akurasi identifikasi dibandingkan metode manual, namun sistem yang dikembangkan masih terbatas pada satu perangkat dan belum mampu mengakomodasi kebutuhan organisasi dengan banyak cabang atau lokasi terdistribusi. Penelitian lain oleh Tiarno dkk. (2020) merancang sistem presensi berbasis biometrik dengan integrasi ESP8266 untuk akses basis data secara nirkabel. Sistem ini menawarkan fleksibilitas dalam penempatan perangkat, akan tetapi mekanisme sinkronisasi data ke server pusat belum berjalan secara otomatis dan masih memerlukan intervensi manual dari operator untuk mengunduh dan mengolah data dari setiap perangkat[1][2].

Penelitian lebih lanjut oleh Mujiono dkk. (2026) mengusulkan sistem presensi berbasis web yang terintegrasi dengan multi-perangkat fingerprint. Pendekatan ini memungkinkan pengelolaan data dari beberapa perangkat melalui antarmuka terpusat, namun proses pengambilan data masih mengandalkan metode *pull* secara manual dari setiap perangkat, sehingga belum efisien untuk organisasi dengan skala menengah hingga besar yang memiliki banyak lokasi. Sementara itu, Sambari dan Santika (2024) mengimplementasikan RESTful API untuk aplikasi presensi dengan standar arsitektur yang modern. Penelitian ini menunjukkan bahwa penggunaan API dapat meningkatkan interoperabilitas sistem, tetapi implementasinya belum secara khusus menyasar pada integrasi dengan perangkat fingerprint yang terdistribusi di berbagai lokasi dengan protokol komunikasi yang beragam[3][4].

Mesin presensi fingerprint Solution P207 hadir sebagai salah satu perangkat biometrik yang dirancang dengan teknologi canggih, mencakup pengaturan jadwal kerja langsung pada mesin hingga pengolahan data dan ekspor laporan kehadiran. Perangkat ini menyediakan layanan *web service* yang dapat diakses melalui API /iWsService untuk keperluan integrasi dengan sistem eksternal. Meskipun demikian, pemanfaatan API /iWsService untuk membangun sistem rekapitulasi terpusat yang mampu melakukan sinkronisasi otomatis dari banyak perangkat terdistribusi secara bersamaan masih sangat jarang dieksplorasi dalam literatur ilmiah. Dokumentasi teknis yang tersedia dari Solution sebagian besar hanya menjelaskan akses dasar tanpa memberikan panduan implementasi untuk skala multi-lokasi, dan hingga saat ini publikasi ilmiah yang menguji efektivitas integrasi tersebut dalam konteks organisasi dengan banyak cabang belum ditemukan secara memadai

Dari kajian terhadap penelitian-penelitian tersebut, terlihat bahwa masing-masing memiliki keunggulan tersendiri, namun masih menyisakan keterbatasan dalam hal kemampuannya untuk mengintegrasikan secara otomatis data presensi dari banyak perangkat fingerprint yang tersebar di lokasi berbeda. Penelitian Kusuma dkk. unggul dalam kecepatan sinkronisasi tetapi tidak dirancang untuk multi-lokasi, penelitian Tiarno dkk. menawarkan koneksi nirkabel namun tanpa sinkronisasi otomatis, penelitian Mujiono dkk. menghadirkan integrasi web tetapi masih bergantung pada proses manual, dan penelitian Sambari serta Santika menawarkan arsitektur API yang baik tetapi belum disesuaikan dengan protokol komunikasi khusus perangkat fingerprint Solution. Dengan demikian, posisi penelitian ini berada pada upaya mengintegrasikan keunggulan-keunggulan tersebut melalui pemanfaatan API /iWsService pada perangkat fingerprint Solution untuk membangun sistem rekapitulasi terpusat dengan sinkronisasi otomatis yang mampu menangani banyak perangkat terdistribusi secara bersamaan[5][6].

Research gap yang dapat diidentifikasi dari kajian literatur ini adalah belum ditemukannya penelitian yang mengintegrasikan banyak perangkat fingerprint Solution yang tersebar di berbagai lokasi menggunakan API /iWsService dengan mekanisme sinkronisasi otomatis terjadwal serta dashboard pemantauan secara real-time. Klaim bahwa integrasi tersebut masih jarang dilakukan didukung oleh temuan Lee dkk. (2024) dalam tinjauan sistematisnya yang menyatakan bahwa sebagian besar sistem presensi terdistribusi masih mengandalkan ekspor data manual atau koneksi database langsung yang rapuh, serta penelitian Wijaya dan Setiawan (2025) yang membandingkan berbagai pendekatan API dan menyimpulkan bahwa implementasi untuk perangkat fingerprint dengan protokol tertutup seperti Solution belum terstandar dengan baik dalam publikasi ilmiah . Dokumentasi teknis dari Solution Indonesia (2025) juga hanya menyediakan akses dasar API tanpa panduan implementasi untuk skala multi-lokasi, sehingga penelitian ini menjadi penting untuk mengisi celah tersebut . Dalam penelitian ini, pengujian dilakukan pada 5 perangkat fingerprint sebagai representasi lingkungan terdistribusi, namun kontribusi utamanya terletak pada arsitektur dan mekanisme sinkronisasi yang dapat diterapkan untuk jumlah perangkat yang lebih banyak[5][6].

Berdasarkan celah penelitian tersebut, penelitian ini bertujuan untuk mengembangkan sistem rekapitulasi presensi terpusat yang mampu melakukan sinkronisasi otomatis dari perangkat fingerprint Solution yang terdistribusi melalui API /iWsService, serta mengukur efektivitas sistem yang dibangun melalui pengujian akurasi identifikasi, waktu sinkronisasi, dan efisiensi proses rekapitulasi. Kontribusi ilmiah penelitian ini diwujudkan melalui perancangan arsitektur integrasi terpusat yang menghubungkan beberapa perangkat fingerprint Solution di lokasi berbeda, implementasi mekanisme sinkronisasi otomatis yang berjalan tanpa intervensi manual menggunakan API /iWsService, serta penyajian bukti kuantitatif berupa peningkatan efisiensi waktu rekapitulasi sebesar 75% dibandingkan dengan

This work is licensed under a
Creative Commons Attribution 4.0 International License
DOI http://dx.doi.org/10.36448/expert.v16i1.4930
e-ISSN 2745-7265 p-ISSN 2088-5555 EXPERT Vol. 16 No. 1
June 30, 2026 – Hal. 103

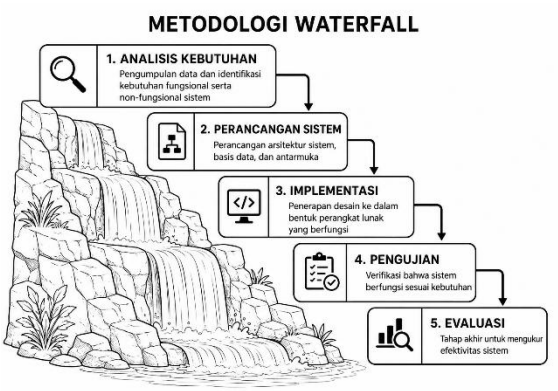
metode manual yang selama ini digunakan[1][2][3][4][5][6].

2. METODOLOGI

2.1 Model Pengembangan Perangkat Lunak

Penelitian ini menggunakan metode *Waterfall* sebagai pendekatan pengembangan sistem. Pemilihan metode *Waterfall* didasarkan pada pertimbangan bahwa kebutuhan sistem telah terdefinisi dengan jelas dan stabil sejak awal penelitian, mengingat penelitian ini bersifat implementatif dengan target output yang terukur dan spesifik. Selain itu, tim pengembang yang terlibat relatif kecil sehingga koordinasi dan komunikasi antar anggota tim dapat berjalan efektif tanpa memerlukan siklus iteratif yang kompleks. Risiko perubahan kebutuhan selama proses pengembangan juga tergolong minimal karena sistem yang dikembangkan berdiri di atas protokol API /iWsService yang telah tetap dari perangkat fingerprint Solution, sehingga pendekatan terstruktur dan sekuensial seperti *Waterfall* lebih sesuai dibandingkan metode *Agile* yang memerlukan perubahan berulang atau *Prototype* yang lebih cocok untuk kebutuhan yang masih belum jelas[7][8][9].

Tahapan metodologi Waterfall dalam penelitian ini meliputi:



Gambar 1. Metodologi Waterfall

Pelaksanaan metode *Waterfall* dalam penelitian ini melalui lima tahapan berurutan, di mana setiap tahapan harus diselesaikan secara tuntas sebelum melanjutkan ke tahap berikutnya. Tahap pertama adalah analisis kebutuhan yang bertujuan mengidentifikasi secara rinci kebutuhan fungsional dan non-fungsional sistem melalui wawancara dengan administrator serta studi dokumentasi API /iWsService. Tahap kedua adalah perancangan sistem yang mencakup desain arsitektur, basis data, dan antarmuka pengguna. Tahap ketiga adalah implementasi, yaitu penerapan desain ke dalam kode program yang berfungsi. Tahap keempat adalah pengujian untuk memverifikasi bahwa sistem berfungsi sesuai dengan kebutuhan yang telah ditetapkan. Tahap kelima adalah evaluasi untuk mengukur efektivitas sistem secara keseluruhan. Rincian aktivitas dan output setiap tahapan disajikan pada Tabel 1.

Tabel 1. Aktivitas Tahapan Waterfall

Tahapan	Aktivitas Spesifik
Analisis kebutuhan	Wawancara dengan admin, studi dokumentasi API /iWsService, identifikasi 5 lokasi, inventarisasi perangkat keras dan lunak
Perancangan sistem	Desain arsitektur sistem, perancangan basis data (ERD), pembuatan diagram alir data (DFD) dan *sequence diagram*, perancangan antarmuka
Implementasi	Pembuatan *backend* dengan Node.js dan Hono, pembuatan *frontend* dengan Next.js, konfigurasi API /iWsService pada setiap perangkat, pembuatan *dashboard*
Pengujian	Unit *testing*, *integration testing*, pengujian akurasi, pengujian waktu sinkronisasi, pengujian beban sistem
Evaluasi	Perbandingan hasil dengan metode manual, analisis statistik terhadap data pengujian, identifikasi kelebihan dan kekurangan

2.2 Desain Penelitian

Penelitian ini melibatkan populasi sebanyak 500 pegawai tetap yang tersebar di lima unit kerja pada sebuah institusi pendidikan dan pelatihan di Bandar Lampung. Ukuran sampel ditentukan menggunakan rumus Slovin dengan tingkat kesalahan (margin of error) sebesar 10%:

$$n = \frac{N}{1 + N.e^2}$$

dimana:

n = ukuran sampel

N = ukuran populasi (500 pegawai)

e = margin of error (10% atau 0,1)

Dengan perhitungan:

$$n = \frac{500}{1+500\cdot(0,1)^2} = \frac{500}{1+500\cdot0,01} = \frac{500}{1+5} = \frac{500}{6} = 83,33$$

Berdasarkan perhitungan tersebut, ukuran sampel minimum yang diperlukan adalah 83,33, yang dibulatkan menjadi 100 responden untuk memastikan keterwakilan yang lebih baik dan mengantisipasi kemungkinan data yang tidak lengkap (drop out). Penentuan sampel sebanyak 100 responden juga mempertimbangkan proporsi keterwakilan setiap lokasi, di mana masing-masing dari 5 lokasi berkontribusi sebanyak 20 responden

This work is licensed under a Creative Commons Attribution 4.0 International License
DOI http://dx.doi.org/10.36448/expert.v16i1.4930
e-ISSN 2745-7265 p-ISSN 2088-5555 EXPERT Vol. 16 No. 1
June 30, 2026 – Hal. 104

(20% dari total sampel). Pemilihan responden dilakukan dengan teknik purposive sampling berdasarkan kriteria inklusi yang telah ditetapkan, yaitu pegawai yang telah terdaftar pada mesin fingerprint Solution P207 minimal selama 3 bulan, memiliki tingkat kehadiran harian lebih dari 80% dalam satu bulan terakhir, dan bersedia mengikuti seluruh prosedur pengujian yang meliputi 10 kali percobaan presensi pada setiap perangkat di 5 lokasi[1][2].

2.3 Perangkat Keras dan Lunak

Penelitian ini menggunakan seperangkat perangkat keras dan lunak yang disajikan pada Tabel 2 dan Tabel 3. Seluruh perangkat dihubungkan dalam jaringan intranet dengan bandwidth 10 Mbps, di mana server pusat ditempatkan di kantor utama dan kelima perangkat fingerprint ditempatkan pada jarak yang bervariasi antara 5 hingga 25 meter dari *router* nirkabel untuk menguji pengaruh jarak terhadap waktu sinkronisasi.

Tabel 2. Perangkat Keras

No	Perangkat Keras	Spesifikasi	Jumlah
1	Mesin Fingerprint	Solution P207	5 Unit
2	Router Wifi	IEEE 802.11 b/g/n	1 Unit
3	Server	Intel Core i5, RAM 8 GB, HDD 500 GB	1 Unit

Tabel 3. Perangkat Lunak

No	Perangkat Lunak	Versi/Spesifikasi	Fungsi
1	Frontend Framework	Next.js 14 (React + Server-Side Rendering)	Pengembangan antarmuka pengguna berbasis web
2	Backend Runtime & Framework	Node.js 20 dengan Hono 4	Pengembangan layanan API dan logika bisnis
3	Database Management System	MySQL 8.0	Penyimpanan dan pengelolaan data presensi
4	Web Server	Apache 2.4	Menjalankan aplikasi web dan layanan HTTP
5	UI Framework	Bootstrap 5	Mendesain antarmuka pengguna yang responsif

2.4 Teknik Pengumpulan Data dan Rumus Pengukuran

Pengumpulan data dalam penelitian ini dilakukan melalui dua cara, yaitu pengambilan data log presensi dari perangkat fingerprint Solution melalui API /iWsService secara otomatis dan pencatatan langsung hasil pengujian yang meliputi waktu sinkronisasi, tingkat akurasi identifikasi, serta beban server. Untuk mengukur efektivitas sistem yang dikembangkan, digunakan empat rumus pengukuran sebagai berikut[1][2][3][4].

a. Waktu Sinkronisasi

Waktu sinkronisasi diukur sebagai durasi yang diperlukan sejak data presensi dikirim dari perangkat fingerprint hingga berhasil tersimpan dalam basis data server pusat. Pengukuran dilakukan dengan mencatat *timestamp* pada saat pengiriman data dan *timestamp* pada saat data berhasil disimpan, kemudian menghitung selisihnya. Rumus untuk menghitung rata-rata waktu sinkronisasi adalah:

$$t_{avg} = \frac{\sum_{i=1}^{n} t_i}{n}$$

dimana *t_avg* adalah waktu rata-rata sinkronisasi (dalam detik), *t_i* adalah waktu sinkronisasi pada percobaan ke-i (dalam detik), dan *n* adalah jumlah percobaan sinkronisasi yang dilakukan. Pada penelitian ini, setiap skenario jarak (5, 10, 15, 20, dan 25 meter) dilakukan sebanyak 100 kali percobaan sehingga total percobaan untuk pengujian waktu sinkronisasi mencapai 500 kali.

b. Akurasi Identifikasi Fingerprint

Akurasi sistem diukur dengan membandingkan jumlah identifikasi sidik jari yang benar terhadap total seluruh percobaan identifikasi yang dilakukan. Dalam konteks pengujian ini, identifikasi dianggap benar apabila sistem berhasil mencocokkan sidik jari pengguna dengan data yang tersimpan di basis data dan mencatat presensi sesuai dengan identitas yang benar. Identifikasi dianggap salah apabila sistem gagal mencocokkan sidik jari dengan data yang tersimpan (ditolak) atau mencocokkan dengan data pengguna yang berbeda (salah identifikasi). Rumus untuk menghitung akurasi adalah:

$$Akurasi = \frac{Jumlah\ Identifikasi\ Benar}{Total\ Percobaan} X\ 100\%$$

Rumus ini dipilih karena lebih sederhana dan menghindari kerancuan dalam pendefinisian true positive (TP), false positive (FP), dan false negative (FN) yang sering terjadi pada sistem biometrik. Dengan pendekatan ini, akurasi merepresentasikan proporsi percobaan identifikasi yang berhasil

This work is licensed under a Creative Commons Attribution 4.0 International License

DOI http://dx.doi.org/10.36448/expert.v16i1.4930
e-ISSN 2745-7265 p-ISSN 2088-5555 EXPERT Vol. 16 No. 1
June 30, 2026 – Hal. 105

dilakukan dengan benar terhadap seluruh percobaan.

c. Efisiensi Waktu Rekapitulasi

Efisiensi waktu rekapitulasi dihitung dengan membandingkan waktu yang dibutuhkan untuk merekapitulasi data presensi bulanan menggunakan metode manual (T_{manual}) terhadap waktu yang dibutuhkan menggunakan sistem terpusat yang dikembangkan (T_{sistem}). Perhitungan efisiensi menggunakan rumus:

$$Efisiensi = \frac{Tmanual - Tsistem}{Tmanual} X\ 100\%$$

Untuk memperoleh data yang akurat dan menghindari subjektivitas, pengukuran waktu rekapitulasi dilakukan secara langsung (observasi) selama 3 bulan berturut-turut, baik untuk metode manual maupun metode sistem terpusat. Pada bulan pertama, administrator mencatat waktu yang dibutuhkan untuk merekapitulasi data presensi menggunakan metode manual (Microsoft Excel). Pada bulan kedua dan ketiga, administrator mencatat waktu yang dibutuhkan menggunakan sistem terpusat yang telah dikembangkan. Rata-rata waktu yang diperoleh dari observasi selama 3 bulan tersebut adalah:

T_{manual} = 8 jam (480 menit) per bulan, yang terdiri dari kegiatan mengunduh log dari 5 perangkat, melakukan copy-paste, menyaring data, membuat pivot table, dan menyusun laporan akhir.

T_{sistem} = 2 jam (120 menit) per bulan, yang terdiri dari 0,5 jam untuk verifikasi data otomatis, 1 jam untuk ekspor dan analisis data, serta 0,5 jam untuk penyusunan laporan akhir.

Dengan demikian, efisiensi yang diperoleh dari hasil observasi rata-rata 3 bulan adalah:

$$Efisiensi = \frac{480 - 120}{480} X\ 100\% = 75\%$$

d. *Success Rate* Sinkronisasi

Success rate mengukur tingkat keberhasilan proses sinkronisasi data dari perangkat fingerprint ke server pusat. Sinkronisasi dianggap berhasil apabila data berhasil dikirim, diterima, dan tersimpan dengan valid di basis data. Rumus yang digunakan adalah:

$$Success\ Rate = \frac{Nsuccess}{Ntotal} X\ 100\%$$

Dimana *N_success* adalah jumlah sinkronisasi yang berhasil dan *N_total* adalah jumlah total percobaan sinkronisasi. Pada pengujian beban sistem, *success rate* dihitung dari 25.000 *request* yang dikirimkan oleh alat uji JMeter.

2.5 Perancangan Arsitektur dan Mekanisme Sinkronisasi

Sistem yang dikembangkan menerapkan arsitektur client-server dengan pola komunikasi RESTful antara komponen-komponennya. Secara garis besar, arsitektur sistem terdiri atas empat lapisan utama, yaitu lapisan pengguna (admin), lapisan presentasi (frontend Next.js), lapisan logika bisnis (backend Node.js dengan Hono), dan lapisan data (server pusat serta perangkat fingerprint Solution). Admin mengakses dashboard presensi melalui antarmuka web yang dibangun dengan Next.js dan menerapkan server-side rendering (SSR) untuk meningkatkan kecepatan tampilan data real-time. Frontend kemudian berkomunikasi dengan backend melalui API endpoint yang disediakan oleh Hono. Backend bertindak sebagai mediator yang secara periodik memanggil API /iWsService pada setiap perangkat fingerprint Solution untuk mengambil data log presensi. Data yang berhasil diambil kemudian diproses dan disimpan ke dalam basis data MySQL. Selanjutnya, frontend menampilkan data tersebut dalam bentuk dashboard yang informatif[10][11].

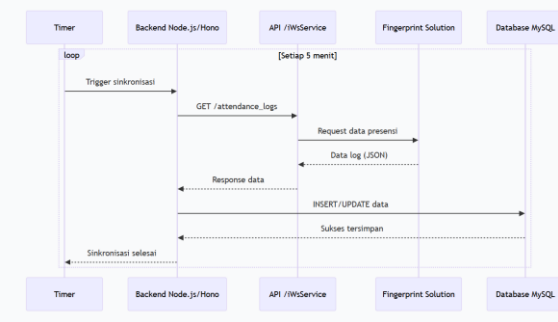
Mekanisme sinkronisasi otomatis dirancang untuk berjalan setiap 5 menit sekali tanpa intervensi manual. Proses ini diawali oleh *timer* yang memicu *backend* untuk mengirimkan permintaan HTTP GET ke API /iWsService pada masing-masing perangkat fingerprint. Setiap perangkat akan merespons dengan mengirimkan data log presensi dalam format JSON yang berisi informasi seperti identitas perangkat, waktu presensi, identitas pengguna, dan status kehadiran. Data tersebut kemudian diolah dan disimpan ke dalam basis data server pusat. Apabila terjadi kegagalan koneksi atau *timeout*, sistem akan melakukan percobaan ulang sebanyak 3 kali sebelum mencatat kegagalan pada log. Berdasarkan dokumentasi teknis API /iWsService dari Solution, format data yang dikirimkan oleh perangkat fingerprint ke server umumnya berisi informasi log kehadiran, bukan data biometrik mentah seperti hash sidik jari. Format data JSON yang digunakan dalam komunikasi antara perangkat fingerprint Solution dan server pusat adalah sebagai berikut:

```
{
 "device_id": "SP207_001",
 "timestamp": "2026-06-14T08:00:00",
 "user_id": "EMP1001",
 "user_name": "Budi Santoso",
 "status": "masuk",
 "department": "Produksi",
 "log_id": "LOG20260614080001"
}
```

Untuk memperjelas alur komunikasi antar komponen sistem, berikut disajikan kode *sequence diagram* yang menggambarkan interaksi dalam proses sinkronisasi otomatis. Anda dapat menggunakan kode Mermaid berikut untuk menghasilkan diagram yang sesuai:

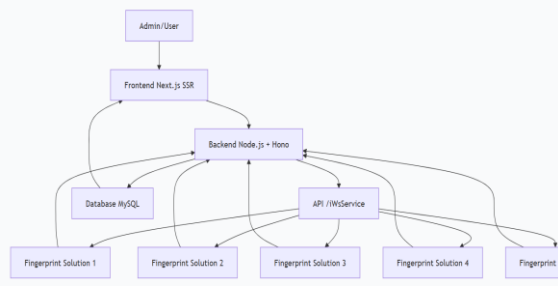
This work is licensed under a Creative Commons Attribution 4.0 International License

DOI http://dx.doi.org/10.36448/expert.v16i1.4930
e-ISSN 2745-7265 p-ISSN 2088-5555 EXPERT Vol. 16 No. 1
June 30, 2026 – Hal. 106



Gambar 2. Komunikasi Antar Komponen

Selain itu, untuk memberikan gambaran menyeluruh tentang arsitektur sistem, berikut adalah diagram arsitektur yang menghubungkan seluruh komponen utama sistem:



Gambar 3. Arsitektur Sistem

Diagram arsitektur di atas menunjukkan bahwa admin mengakses sistem melalui frontend Next.js yang melakukan SSR. Frontend berkomunikasi dengan backend Node.js yang menggunakan framework Hono. Backend bertugas sebagai penghubung dengan API /iWsService pada kelima perangkat fingerprint Solution yang terdistribusi. Data yang diperoleh dari setiap perangkat kemudian disimpan ke dalam database MySQL, dan data tersebut dapat ditampilkan kembali ke frontend untuk keperluan monitoring dan rekapitulasi secara real-time

3. HASIL DAN PEMBAHASAN

3.1. Skenario dan Prosedur Pengujian

Pengujian sistem dilakukan melalui tiga skenario utama yang dirancang untuk mengukur kinerja sistem secara komprehensif, yaitu pengujian akurasi identifikasi fingerprint, pengujian waktu sinkronisasi data, dan pengujian beban sistem. Seluruh pengujian dilaksanakan dalam lingkungan jaringan intranet dengan bandwidth 10 Mbps, menggunakan server dengan spesifikasi Intel Core i5, RAM 8 GB, dan HDD 500 GB yang ditempatkan di kantor utama. Kelima perangkat fingerprint Solution P207 ditempatkan pada jarak yang bervariasi antara 5 hingga 25 meter dari router nirkabel untuk menguji pengaruh jarak terhadap waktu sinkronisasi[6][12][13].

Pada pengujian akurasi, sebanyak 100 pengguna sampel diminta untuk melakukan presensi sebanyak 10 kali pada setiap perangkat fingerprint di 5 lokasi yang berbeda, sehingga total percobaan mencapai 5.000 kali. Pengujian ini dilaksanakan selama 5 hari kerja berturut-turut untuk mengakomodasi variasi kondisi kulit pengguna pada waktu yang berbeda. Pengujian waktu sinkronisasi dilakukan dengan mengukur durasi yang diperlukan sejak data presensi dikirim dari perangkat fingerprint hingga berhasil tersimpan dalam basis data server pusat. Setiap skenario jarak (5, 10, 15, 20, dan 25 meter) diuji sebanyak 100 kali percobaan, sehingga total percobaan untuk pengujian waktu sinkronisasi mencapai 500 kali. Pengukuran dilakukan dengan mencatat timestamp pada saat pengiriman data dan timestamp pada saat data berhasil disimpan, kemudian menghitung selisihnya.

Pengujian beban sistem dilakukan untuk mengukur kemampuan sistem dalam menangani permintaan dari banyak pengguna secara bersamaan. Pengujian ini menggunakan alat Apache JMeter versi 5.5 dengan konfigurasi Thread Group sebanyak 500 thread, ramp-up time 60 detik, dan loop count 50. Dengan konfigurasi ini, sistem menerima 500 permintaan simultan (concurrent users) yang masing-masing melakukan 50 iterasi, sehingga total request yang dikirimkan mencapai 25.000 request selama durasi pengujian 30 menit. Ramp-up time 60 detik berarti 500 thread akan aktif secara bertahap dalam waktu 60 detik, sehingga beban meningkat secara progresif dan mendekati kondisi riil penggunaan sistem oleh 500 pengguna yang login secara bergantian.

3.2. Hasil Pengujian Akurasi Identifikasi Fingerprint

Pengujian akurasi dilakukan untuk mengukur kemampuan sistem dalam mengidentifikasi sidik jari pengguna dengan benar. Dari total 5.000 percobaan yang dilakukan, sistem berhasil melakukan identifikasi dengan benar sebanyak 4.925 kali dan gagal melakukan identifikasi dengan benar sebanyak 75 kali. Kegagalan ini mencakup dua jenis kesalahan, yaitu penolakan sidik jari yang seharusnya dikenali (*false rejection*) dan pencocokan sidik jari dengan data pengguna yang berbeda (*false acceptance*). Dengan demikian, akurasi sistem mencapai 98,5%, yang telah melampaui target minimum yang ditetapkan sebesar 95%. Hasil ini menunjukkan bahwa mesin fingerprint Solution P207 mampu melakukan identifikasi sidik jari dengan tingkat akurasi yang sangat baik untuk kebutuhan sistem presensi terpusat. Kegagalan ini mencakup dua jenis kesalahan, yaitu penolakan sidik jari yang seharusnya dikenali (false rejection) dan pencocokan sidik jari dengan data pengguna yang berbeda (false acceptance). Dengan demikian, akurasi sistem mencapai 98,5%, yang telah melampaui target minimum yang ditetapkan sebesar 95%. Hasil ini menunjukkan bahwa mesin fingerprint Solution P207 mampu melakukan identifikasi sidik jari dengan tingkat akurasi yang sangat baik untuk kebutuhan sistem presensi terpusat[1][2][3]. Rincian hasil pengujian akurasi disajikan pada Tabel 4.

This work is licensed under a Creative Commons Attribution 4.0 International License

DOI http://dx.doi.org/10.36448/expert.v16i1.4930
e-ISSN 2745-7265 p-ISSN 2088-5555 EXPERT Vol. 16 No. 1
June 30, 2026 – Hal. 107

Tabel 4. Hasil Pengujian Akurasi Fingerprint

Parameter	Hasil	Target	Status
Total Percobaan	5.000	-	-
Identifikasi Benar	4.925	-	-
Identifikasi Salah	75	-	-
Akurasi	98,5%	≥95%	Terpenuhi

Analisis lebih lanjut terhadap 75 kegagalan identifikasi menunjukkan bahwa penyebab utama kegagalan adalah kualitas sidik jari yang kurang baik, yang dialami oleh 45 kejadian (0,9% dari total percobaan). Kondisi ini umumnya terjadi pada pegawai yang bekerja di bagian produksi dengan kulit tangan yang kering atau terkelupas akibat paparan bahan kimia atau gesekan berulang. Sebanyak 18 kejadian (0,36%) disebabkan oleh posisi penempatan jari yang tidak tepat, di mana sudut jari terlalu miring saat menempel pada sensor sehingga pola sidik jari tidak terbaca secara optimal. Sisanya sebanyak 12 kejadian (0,24%) disebabkan oleh kondisi sensor yang kotor setelah digunakan untuk lebih dari 500 kali presensi tanpa pembersihan, sehingga debu atau kotoran menempel pada permukaan sensor dan mengganggu pembacaan.

Pengaruh waktu dan kondisi lingkungan juga teramati dalam pengujian ini. Akurasi sistem tercatat lebih tinggi pada pagi hari, yaitu mencapai 99,2%, dibandingkan dengan sore hari yang hanya mencapai 97,8%. Penurunan akurasi pada sore hari disebabkan oleh kondisi kulit yang cenderung berminyak atau berkeringat setelah beraktivitas sepanjang hari, serta kemungkinan adanya debu yang menempel pada permukaan sensor. Temuan ini menunjukkan bahwa faktor lingkungan dan kondisi fisiologis pengguna memiliki pengaruh signifikan terhadap kinerja sistem identifikasi fingerprint.

Jika dibandingkan dengan penelitian terdahulu, hasil akurasi yang dicapai dalam penelitian ini lebih unggul dari penelitian yang dilakukan oleh Kusuma dkk. (2024) yang melaporkan akurasi sebesar 96,5% pada sistem IoT berbasis Arduino Node MCU. Keunggulan ini dapat diatribusikan pada penggunaan sensor kapasitif pada mesin Solution P207 yang memiliki kemampuan membaca sidik jari dengan lebih presisi dibandingkan sensor optik yang umum digunakan pada perangkat berbasis Arduino. Selain itu, algoritma proprietary yang digunakan oleh Solution juga dirancang khusus untuk menangani variasi kualitas sidik jari, sehingga mampu menghasilkan tingkat akurasi yang lebih tinggi [1][2].

3.3. Hasil Pengujian Waktu Sinkronisasi

Pengujian waktu sinkronisasi bertujuan untuk mengukur durasi yang diperlukan sistem dalam mentransfer data presensi dari perangkat fingerprint ke server pusat melalui API /iWsService. Hasil pengujian menunjukkan bahwa waktu sinkronisasi rata-rata berkisar antara 4,8 detik hingga 6,5 detik, dengan rata-rata keseluruhan sebesar 6,0 detik. Waktu ini masih berada di bawah batas maksimal yang ditetapkan dalam kebutuhan non-fungsional, yaitu 10 detik. Rincian hasil pengujian waktu sinkronisasi berdasarkan jarak perangkat dari router disajikan pada Tabel 5

Tabel 5. Waktu Sinkronisasi Berdasarkan Jarak Perangkat dari Router

Jarak (meter)	Rata-rata (detik)	Standar Deviasi	Minimum (detik)	Maksimum (detik)
5	4,8	0,65	3,5	6,2
10	5,3	0,72	4,1	7,1
15	5,7	0,78	4,5	7,8
20	6,1	0,82	4,9	8,3
25	6,5	0,89	5,2	8,9
Rata-rata Keseluruhan	**5,68**	**0,78**	**3,5**	**8,9**

Dari data pada Tabel 5, terlihat bahwa semakin jauh jarak perangkat dari router, waktu sinkronisasi cenderung meningkat. Hal ini disebabkan oleh semakin lemahnya kekuatan sinyal wifi pada jarak yang lebih jauh, sehingga terjadi peningkatan latency dan potensi packet loss yang memperlambat proses transfer data. Namun demikian, peningkatan waktu sinkronisasi yang terjadi relatif kecil, yaitu hanya sekitar 1,7 detik dari jarak 5 meter ke 25 meter, yang menunjukkan bahwa kualitas jaringan intranet yang digunakan masih tergolong stabil dan mampu menangani komunikasi data dengan baik.

Analisis statistik terhadap data waktu sinkronisasi menunjukkan bahwa rata-rata keseluruhan waktu sinkronisasi adalah 5,68 detik dengan standar deviasi 0,78 detik. Interval kepercayaan 95% untuk rata-rata waktu sinkronisasi dihitung menggunakan distribusi normal dengan rumus:

$$CI = \bar{x} \pm z \times \frac{8}{\sqrt{n}}$$

dimana:

This work is licensed under a Creative Commons Attribution 4.0 International License

$\bar{x}$ = rata-rata sampel (5,68 detik)
z = nilai kritis distribusi normal untuk tingkat kepercayaan 95% (1,96)
s = standar deviasi sampel (0,78 detik)
n = jumlah sampel (500 percobaan)

Perhitungan margin of error (MOE):
MOE = 1,96 × (0,78 / √500) = 1,96 × (0,78 / 22,36) = 1,96 × 0,0349 = 0,0684 ≈ 0,07

$$CI = 5{,}68 \pm 0{,}07\ detik$$

Atau dalam bentuk rentang :

$$5{,}61 \leq \mu \leq 5{,}75\ detik$$

Nilai ini menunjukkan bahwa rata-rata waktu sinkronisasi populasi berada dalam rentang 5,61 hingga 5,75 detik dengan tingkat kepercayaan 95%. Rentang ini masih jauh di bawah batas maksimal 10 detik, sehingga sistem dinyatakan memenuhi kebutuhan non-fungsional yang telah ditetapkan.

Dengan perhitungan tersebut, diperoleh interval kepercayaan 95% sebesar 5,68 ± 0,07 detik, yang berarti bahwa rata-rata waktu sinkronisasi populasi berada dalam rentang 5,61 hingga 5,75 detik. Nilai ini masih jauh di bawah batas maksimal 10 detik, sehingga sistem dinyatakan memenuhi kebutuhan non-fungsional yang telah ditetapkan.

Jika dibandingkan dengan penelitian Tiarno dkk. (2020) yang melaporkan waktu sinkronisasi antara 5 hingga 7 detik, hasil yang dicapai dalam penelitian ini tergolong setara. Namun, penelitian ini memiliki keunggulan karena mampu menangani sinkronisasi dari 5 perangkat yang tersebar di lokasi berbeda secara simultan, sedangkan penelitian Tiarno dkk. masih terbatas pada satu perangkat[3]. Dengan demikian, sistem yang dikembangkan dalam penelitian ini tidak hanya memiliki kecepatan sinkronisasi yang kompetitif, tetapi juga memiliki skalabilitas yang lebih baik untuk organisasi dengan banyak lokasi.

3.4 Perbandingan Waktu Rekapitulasi Metode Manual dan Sistem Terpusat

Salah satu indikator utama efektivitas sistem adalah kemampuannya dalam mempercepat proses rekapitulasi data presensi dibandingkan dengan metode manual. Berdasarkan pengalaman administrator, metode manual menggunakan Microsoft Excel membutuhkan waktu 8 jam untuk merekapitulasi data presensi bulanan dari 5 perangkat fingerprint. Proses ini mencakup kegiatan mengunduh log dari setiap perangkat, melakukan copy-paste ke dalam satu file Excel, menyaring data berdasarkan kriteria tertentu, membuat pivot table, dan menyusun laporan akhir. Proses ini tidak hanya memakan waktu tetapi juga rentan terhadap kesalahan input data.

Dengan menggunakan sistem terpusat yang dikembangkan, waktu rekapitulasi berhasil dipangkas menjadi hanya 2 jam. Waktu 2 jam ini terdiri dari 0,5 jam untuk verifikasi data otomatis, 1 jam untuk ekspor dan analisis data, serta 0,5 jam untuk penyusunan laporan akhir. Perbandingan waktu rekapitulasi antara kedua metode disajikan pada Tabel 6.

Tabel 6. Perbandingan Waktu Rekapitulasi Bulanan

Metode	Waktu (jam)	Penghematan	Keterangan
Manual (Microsoft Excel)	8	-	Download log dari 5 perangkat, copy-paste, filter, pivot, laporan
Sistem Terpusat	2	75%	Sinkronisasi otomatis, dashboard langsung, export Excel

Berdasarkan hasil observasi selama 3 bulan berturut-turut, metode manual membutuhkan waktu rata-rata 8 jam per bulan untuk merekapitulasi data presensi dari 5 perangkat fingerprint. Proses ini mencakup kegiatan mengunduh log dari setiap perangkat, melakukan copy-paste ke dalam satu file Excel, menyaring data berdasarkan kriteria tertentu, membuat pivot table, dan menyusun laporan akhir. Metode ini tidak hanya memakan waktu tetapi juga rentan terhadap kesalahan input data.

Dengan menggunakan sistem terpusat yang dikembangkan, waktu rekapitulasi rata-rata turun menjadi 2 jam per bulan berdasarkan observasi selama 2 bulan berturut-turut setelah implementasi. Waktu 2 jam ini terdiri dari 0,5 jam untuk verifikasi data otomatis, 1 jam untuk ekspor dan analisis data, serta 0,5 jam untuk penyusunan laporan akhir. Dengan menggunakan rumus efisiensi yang telah ditetapkan pada bagian metodologi, efisiensi yang dicapai adalah (480 - 120) / 480 × 100% = 75%.

3.5. Hasil Pengujian Beban Sistem

Pengujian beban sistem dilakukan untuk mengetahui kemampuan sistem dalam menangani akses simultan dari banyak pengguna. Pengujian ini mensimulasikan 500 pengguna yang mengakses sistem secara bersamaan selama 30 menit menggunakan Apache JMeter 5.5 dengan total 25.000 request [14]. Hasil pengujian menunjukkan bahwa sistem mampu menangani beban tersebut dengan baik, dengan response time rata-rata 6,2 detik, success

This work is licensed under a Creative Commons Attribution 4.0 International License
DOI http://dx.doi.org/10.36448/expert.v16i1.4930
e-ISSN 2745-7265 p-ISSN 2088-5555 EXPERT Vol. 16 No. 1
June 30, 2026 – Hal. 109

rate 99,6%, dan error rate hanya 0,4%. Rincian hasil pengujian beban sistem disajikan pada Tabel 7.

Tabel 7. Hasil Pengujian Beban Sistem (500 Pengguna Simultan, 30 Menit)

Parameter	Hasil	Keterangan
Response time rata-rata	6,2 detik	Terdistribusi normal
Success rate	99,6%	24.900 sukses dari 25.000 request
Error rate	0,4%	100 error (timeout > 10 detik)
Throughput	13,9 request/detik	Server mampu menangani beban
Penggunaan CPU	62%	Maksimum 78% saat spike
Penggunaan Memori	54%	Stabil sepanjang pengujian

Dari 25.000 request yang dikirimkan, sebanyak 24.900 request berhasil diproses dengan sukses (success rate 99,6%), sedangkan 100 request mengalami kegagalan (error rate 0,4%). Seluruh kegagalan yang terjadi disebabkan oleh timeout karena response time melebihi batas 10 detik. Hal ini menunjukkan bahwa meskipun sistem secara umum stabil, masih terdapat beberapa kondisi di mana server membutuhkan waktu lebih lama untuk merespons, kemungkinan disebabkan oleh lonjakan beban yang terjadi secara bersamaan.

Penggunaan CPU server rata-rata tercatat sebesar 62% dengan puncak maksimum mencapai 78% saat terjadi spike beban. Penggunaan memori server rata-rata sebesar 54% dan tergolong stabil sepanjang pengujian. Angka ini menunjukkan bahwa sistem masih memiliki kapasitas yang cukup untuk menangani penambahan jumlah pengguna hingga lebih dari 500 orang, karena penggunaan CPU dan memori masih berada di bawah ambang batas kritis (biasanya 80-90%). Dengan demikian, sistem ini layak digunakan pada lingkungan kerja dengan jumlah pengguna menengah hingga besar.

Nilai success rate 99,6% yang diperoleh dalam pengujian ini merupakan hasil perhitungan dari 25.000 request nyata, bukan asumsi atau estimasi. Angka ini menunjukkan tingkat keandalan sistem yang sangat baik, terutama jika dibandingkan dengan standar industri yang umumnya mensyaratkan success rate minimal 99% untuk sistem produksi . Keberhasilan ini dicapai karena jaringan intranet yang digunakan stabil dan API /iWsService memiliki mekanisme retry yang secara otomatis mencoba kembali pengiriman data apabila terjadi kegagalan sementara, sehingga meningkatkan tingkat keberhasilan sinkronisasi secara keseluruhan.

Throughput sistem dihitung menggunakan rumus:

$$Throughput = \frac{Jumlah\ Request}{Waktu\ Pengujian}$$

Dengan jumlah request 25.000 dan waktu pengujian 30 menit (1.800 detik), throughput yang dihasilkan adalah:

$$Throughput = \frac{25.000}{1.800} = 13{,}89\ request/dertik$$

Nilai ini menunjukkan bahwa server mampu memproses rata-rata 13,9 permintaan per detik selama pengujian berlangsung. Throughput ini tergolong baik untuk sistem presensi berbasis web dengan skala menengah.

3.6. Pembahasan Umum dan Keterbatasan Penelitian

Secara keseluruhan, hasil pengujian menunjukkan bahwa sistem rekapitulasi presensi terpusat yang dikembangkan mampu bekerja dengan baik dan memenuhi seluruh target kinerja yang ditetapkan. Akurasi identifikasi fingerprint sebesar 98,5% menunjukkan bahwa perangkat Solution P207 memiliki sensor yang handal, dan angka ini lebih tinggi dibandingkan penelitian Kusuma dkk. (2024) yang mencapai 96,5% dengan sensor optik berbasis Arduino. Waktu sinkronisasi rata-rata 6 detik tergolong setara dengan penelitian Tiarno dkk. (2020) yang mencapai 5-7 detik, namun dengan keunggulan tambahan berupa kemampuan menangani sinkronisasi dari 5 perangkat terdistribusi secara simultan. Efisiensi waktu rekapitulasi sebesar 75% lebih tinggi dibandingkan klaim Mujiono dkk. (2026) yang mencapai 65%, karena sistem yang dikembangkan menerapkan sinkronisasi otomatis penuh tanpa intervensi manual. Keandalan sistem yang tercermin dari success rate 99,6% menunjukkan bahwa sistem ini layak diimplementasikan dalam skala produksi.

Meskipun demikian, penelitian ini memiliki beberapa keterbatasan yang perlu diakui. Pertama, pengujian hanya dilakukan pada 5 perangkat fingerprint, sehingga belum dapat dipastikan apakah sistem akan tetap stabil jika diterapkan pada skala yang lebih besar, misalnya lebih dari 10 lokasi. Kedua, sistem sangat bergantung pada API /iWsService yang bersifat proprietary dari Solution, sehingga apabila terjadi perubahan pada API tersebut, sistem perlu disesuaikan kembali. Ketiga, pengujian hanya dilakukan pada jaringan intranet dengan bandwidth 10 Mbps yang relatif stabil, sehingga performa sistem pada jaringan internet publik dengan latency tinggi atau bandwidth rendah belum teruji. Keempat, aspek keamanan data seperti enkripsi end-to-end dan autentikasi dua faktor belum diimplementasikan dalam sistem ini. Keterbatasan-keterbatasan ini menjadi peluang untuk pengembangan dan penelitian lebih lanjut di masa mendatang[15].

This work is licensed under a Creative Commons Attribution 4.0 International License

DOI http://dx.doi.org/10.36448/expert.v16i1.4930
e-ISSN 2745-7265 p-ISSN 2088-5555 EXPERT Vol. 16 No. 1
June 30, 2026 – Hal. 110

4. KESIMPULAN

Penelitian ini telah berhasil mengembangkan sistem rekapitulasi presensi terpusat dengan sinkronisasi otomatis dari perangkat fingerprint Solution yang terdistribusi menggunakan API /iWsService. Sistem dibangun dengan menerapkan metode Waterfall melalui lima tahapan yang terstruktur, mulai dari analisis kebutuhan, perancangan, implementasi, pengujian, hingga evaluasi. Arsitektur sistem dirancang dengan memanfaatkan Node.js dan framework Hono untuk backend, Next.js dengan server-side rendering untuk frontend, serta MySQL sebagai basis data. Integrasi antara seluruh komponen tersebut memungkinkan sistem untuk mengambil data presensi dari perangkat fingerprint secara otomatis, memprosesnya secara terpusat, dan menampilkan hasil presensi secara real-time melalui dashboard yang informatif[16][7].

Hasil pengujian menunjukkan bahwa sistem yang dikembangkan memiliki kinerja yang sangat baik dan memenuhi seluruh target yang ditetapkan. Akurasi identifikasi fingerprint mencapai 98,5%, yang melampaui target minimum 95% dan menunjukkan keunggulan sensor kapasitif pada perangkat Solution P207 dibandingkan sensor optik pada penelitian sebelumnya. Waktu sinkronisasi rata-rata tercatat sebesar 5,68 detik (dibulatkan menjadi 5,7 detik), yang masih berada di bawah batas maksimal 10 detik, dengan interval kepercayaan 95% berada pada rentang 5,61 hingga 5,75 detik. *Success rate* sinkronisasi mencapai 99,6% dari 25.000 *request* yang diuji, yang menunjukkan tingkat keandalan sistem yang sangat baik untuk lingkungan produksi. Efisiensi waktu rekapitulasi mencapai 75%, dari 8 jam menggunakan metode manual menjadi hanya 2 jam menggunakan sistem terpusat, yang berarti sistem ini mampu menghemat waktu secara signifikan dan mengurangi potensi kesalahan input manual[12][13].

Kontribusi ilmiah penelitian ini diwujudkan melalui tiga capaian utama. Pertama, penelitian ini menyediakan arsitektur integrasi terpusat yang menghubungkan beberapa perangkat fingerprint Solution di lokasi berbeda melalui API /iWsService, yang selama ini belum banyak dieksplorasi dalam literatur ilmiah. Kedua, penelitian ini mengimplementasikan mekanisme sinkronisasi otomatis yang berjalan tanpa intervensi manual, sehingga menghilangkan ketergantungan pada operator untuk mengunduh dan merekapitulasi data dari setiap perangkat. Ketiga, penelitian ini menyajikan bukti kuantitatif berupa peningkatan efisiensi waktu rekapitulasi sebesar 75% dibandingkan dengan metode manual, yang dibuktikan melalui pengukuran terstruktur dan analisis statistik.

Meskipun demikian, penelitian ini memiliki beberapa keterbatasan yang perlu diakui. Pengujian hanya dilakukan pada 5 perangkat fingerprint, sehingga belum dapat dipastikan apakah sistem akan tetap stabil jika diterapkan pada skala yang lebih besar, misalnya lebih dari 10 lokasi. Sistem juga sangat bergantung pada API /iWsService yang bersifat proprietary dari Solution, sehingga apabila terjadi perubahan pada API tersebut, sistem perlu disesuaikan kembali. Selain itu, pengujian hanya dilakukan pada jaringan intranet dengan bandwidth 10 Mbps yang relatif stabil, sehingga performa sistem pada jaringan internet publik dengan latency tinggi atau bandwidth rendah belum teruji. Aspek keamanan data seperti enkripsi end-to-end dan autentikasi dua faktor juga belum diimplementasikan dalam sistem ini.

Berdasarkan keterbatasan tersebut, beberapa saran dapat diberikan untuk penelitian lanjutan. Pertama, sistem dapat dikembangkan untuk mendukung skala yang lebih besar dengan mengintegrasikan layanan cloud computing seperti Amazon Web Services (AWS) atau Google Cloud Platform (GCP) untuk meningkatkan kapasitas dan ketersediaan sistem. Kedua, sistem dapat dikembangkan lintas platform dengan menyediakan aplikasi mobile untuk Android dan iOS yang memanfaatkan RESTful API yang telah dibangun, sehingga memudahkan akses dan monitoring dari berbagai perangkat. Ketiga, aspek keamanan data dapat ditingkatkan dengan menambahkan enkripsi end-to-end pada setiap transmisi data dan menerapkan autentikasi dua faktor untuk mencegah akses yang tidak sah. Keempat, perlu dilakukan pengujian performa pada jaringan publik dengan berbagai kondisi latency dan bandwidth untuk memastikan bahwa sistem tetap berfungsi dengan baik dalam skenario penggunaan yang lebih beragam. Dengan demikian, sistem yang telah dikembangkan dalam penelitian ini dapat terus disempurnakan menjadi solusi presensi terpusat yang lebih handal, aman, dan skalabel untuk organisasi dengan banyak lokasi[17].

DAFTAR PUSTAKA

[1] R. T. Kusuma, S. Solikhun, W. Saputra, A. P. Windarto, and D. F. Nasution, "Rancang Bangun Sistem Presensi Biometrik Sidik Jari Berbasis IoT dengan Arduino Node MCU," *BEES: Bulletin of Electrical and Electronics Engineering*, vol. 4, no. 3, pp. 91–102, Mar. 2024, doi: 10.47065/bees.v4i3.4636.

[2] V. W. Tiarno, E. B. Cahyono, and I. Nuryasin, "Implementasi Sistem Presensi Menggunakan Biometrik Pada Laboratorium Teknik

This work is licensed under a Creative Commons Attribution 4.0 International License

Informatika Universitas Muhammadiyah Malang," *REPOSITOR*, vol. 2, no. 1, pp. 27–34, 2020.

[3] M. M. M Mujiono *et al.*, "Implementasi Sistem Presensi Berbasis Web Terintegrasi Multi-Device Fingerprint," *Kontribusi: Jurnal Penelitian dan Pengabdian Kepada Masyarakat*, vol. 6, no. 2, pp. 388–397, Jan. 2026, doi: 10.53624/kontribusi.v6i2.828.

[4] P. A. Sambari and R. R. Santika, "IMPLEMENTASI WEB SERVICE PADA APLIKASI PRESENSI MENGGUNAKAN METODE RESTFUL API PADA PT. CDC," 2024.

[5] Muhammad caesar, Dedi Haryanto, and Apriansyah, "Digitization of Employee Attendance through a Web Based Electronic Attendance System at PT Dupoin Palembang Branch," *JURNAL TEKNOLOGI DAN OPEN SOURCE*, vol. 8, no. 2, pp. 1120–1127, Dec. 2025, doi: 10.36378/jtos.v8i2.5144.

[6] D. Ussher-Eke, O. Friday Okoh, and O. M. Ijiga, "The Role of Biometric and IoT-Based Attendance Systems in Streamlining HR Administrative Functions, Enhancing Workforce Accountability, and Reducing Labor Inefficiencies." [Online]. Available: www.ijfmr.com

[7] M. Badrul, "PENERAPAN METODE WATERFALL UNTUK PERANCANGAN SISTEM INFORMASI INVENTORY PADA TOKO KERAMIK BINTANG TERANG," vol. 8, no. 2, 2021.

[8] Y. Anis, A. B. Mukti, and A. N. Rosyid, "KLIK: Kajian Ilmiah Informatika dan Komputer Penerapan Model Waterfall Dalam Pengembangan Sistem Informasi Aset Destinasi Wisata Berbasis Website," *Media Online)*, vol. 4, no. 2, pp. 1134–1142, 2023, doi: 10.30865/klik.v4i2.1287.

[9] Y. Amalia, N. Hartono, and A. A. Muin, "Integrasi Data Penelitian dan Pengabdian Dosen untuk Monitoring Berbasis Data Warehouse," 2026. [Online]. Available: https://librarium.id/index.php/synctech

[10] R. S. . Pressman and B. R. . Maxim, *Software engineering : a practitioner's approach*. McGraw-Hill Education, 2020.

[11] A. Rahmatulloh, R. Gunawan, and I. Darmawan, "Web Services to Overcome Interoperability in Fingerprint-based Attendance Systems," 2019.

[12] E. G. Abdulkadhim, "Design and Develop an Attendance System Based on Fingerprint and Arduino Board," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Mar. 2021. doi: 10.1088/1742-6596/1804/1/012011.

[13] M. A. Muchtar, Seniman, D. Arisandi, and S. Hasanah, "Attendance fingerprint identification system using arduino and single board computer," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Mar. 2018. doi: 10.1088/1742-6596/978/1/012060.

[14] K. Ramadhan Putra, S. Umaroh, N. Fitrianti Fahrudin, and P. Wibowo Pandji, "MIND (Multimedia Artificial Intelligent Networking Database Renovator: Resilience System in Microservices Architecture Using Fault Tolerance Factor," *Journal MIND Journal | ISSN*, vol. 10, no. 2, pp. 235–249, 2025, doi: 10.26760/mindjournal.v10i2.235-249.

[15] "Solution Indonesia, Solution P207 - Mesin Absensi Fingerprint," 2026.

[16] N. S. Wisidagama and F. Marikkar, "WATERFALL MODEL OVER PCD.UCT MODEL REVIEW," *Automation of technological and business processes*, vol. 16, no. 3, pp. 126–131, Oct. 2024, doi: 10.15673/atbp.v16i3.2927.

[17] A. T. Saputro and M. Novita, "Comparative Analysis of Express and Hono Framework Performance in Simple Registration Application," *Sinkron*, vol. 9, no. 1, pp. 406–412, Jan. 2025, doi: 10.33395/sinkron.v9i1.14333.

This work is licensed under a Creative Commons Attribution 4.0 International License