ULIL ALBAB : Jurnal Ilmiah Multidisiplin
Vol.5, No.12, November 2026

Rancang Bangun Sistem Informasi Presensi Penggajian Karyawan Rumah Sakit Efarina Berbasis Web Microservices Frameworknext.Js Nest.Js

Pitus Lasro Juni Ronaldo Samosir[1], Ari Rizkita[2], Syaipuddin[3]
[1,2,3]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Efarina, Indonesia
E-mail: housepits12@gmail.com[1], aririzkita76@gmail.com[2], Syaipuddinsyaipuddin77@gmail.com[3]

Article History:
Received: 26 Oktober 2026
Revised: 27 Oktober2026
Accepted: 03 November 2026

Keywords: *Sistem Informasi, Penggajian, Presensi, Microservices, Next.js, NestJS, Waterfall.*

Abstrak: *Rumah Sakit Efarina Etaham Karawang yang sedang dalam tahap persiapan operasional penuh membutuhkan sistem manajemen sumber daya manusia yang andal untuk mengelola presensi dan penggajian karyawannya. Saat ini, pengolahan data kehadiran staf medis dari mesin fingerprint dan rekapitulasi perhitungannya yang melibatkan sistem shift 24 jam masih dilakukan secara semi-manual menggunakan spreadsheet pada akhir bulan. Hal ini menciptakan beban administratif yang berat dan sangat rentan terhadap kesalahan manusia (human error). Untuk merealisasikan transisi menuju platform digital terotomatisasi, penelitian ini bertujuan untuk merancang dan membangun sistem informasi presensi dan penggajian karyawan berbasis web dengan menerapkan arsitektur microservices. Metode perancangan penelitian yang digunakan adalah model pengembangan perangkat lunak System Development Life cycle (SDLC) dengan pendekatan Waterfall. Selanjutnya, sistem akan diuji secara fungsional menggunakan Swagger/OpenAPI untuk memvalidasi spesifikasi endpoint API dan memastikan akurasi perhitungan gaji. Penelitian ini diharapkan dapat memberikan solusi presensi dan penggajian yang akurat, efisien, dan transparan sejak awal operasional rumah sakit.*

PENDAHULUAN

Urgensi digitalisasi sistem penggajian di lingkungan fasilitas kesehatan telah mendapat perhatian serius dari para peneliti. (Meenugu, 2025) menegaskan bahwa implementasi sistem penggajian otomatis di rumah sakit merupakan solusi esensial yang mampu merampingkan alur kerja kompleks menjadi lebih akurat melalui kapabilitas pelacakan kehadiran dan perhitungan kompensasi tingkat lanjut. Akan tetapi, banyak institusi kesehatan masih bergantung pada sistem penggajian semi-manual berbasis *spreadsheet*. (Tandrio & Fianty, 2026) menegaskan bahwa ketergantungan pada metode konvensional tersebut membutuhkan intervensi manusia yang sangat tinggi sehingga rentan terhadap kesalahan (*human error*), memicu inkonsistensi data, dan memiliki keterbatasan skalabilitas seiring pertumbuhan organisasi. (Ahmed et al., 2023) membuktikan

bahwa sistem informasi penggajian berbasis web mampu mengotomatisasi perhitungan gaji secara akurat dan efisien sekaligus menghasilkan laporan secara cepat, menjadikannya solusi yang paling efektif untuk menggantikan proses manual yang rentan kesalahan dan pemborosan waktu.

Keberhasilan operasional jangka panjang sistem informasi rumah sakit sangat bergantung pada fondasi arsitektur perangkat lunak yang mendasarinya. Aplikasi web konvensional umumnya dibangun menggunakan arsitektur monolitik yang kaku dan sulit dipelihara ketika diterapkan pada skala institusi medis yang terus berkembang. Sebagai solusi, adopsi arsitektur *microservices* dinilai paling ideal untuk kebutuhan dinamis sistem informasi layanan kesehatan. (Velepucha & Flores, 2023) dalam survei komprehensif terhadap arsitektur *microservices* menegaskan bahwa pendekatan ini memberikan keunggulan nyata dalam skalabilitas, fleksibilitas teknologi, dan kemudahan pemeliharaan dibandingkan arsitektur monolitik. Hal ini sejalan dengan temuan (Mhatre, 2023) yang menyatakan bahwa arsitektur ini memecah sistem menjadi komponen layanan kecil yang independen secara *loosely coupled*, sehingga menghasilkan *platform* yang sangat skalabel, mudah dipelihara, dan tangguh terhadap kegagalan sistem.

Pada kondisi saat ini, pemantauan kehadiran staf di RS Efarina Etaham Karawang telah difasilitasi dengan mesin pemindai biometrik (*fingerprint*) sebagai alat rekam data dasar. Akan tetapi, institusi ini belum memiliki sistem informasi presensi dan penggajian yang terintegrasi untuk mengolah data mentah tersebut secara otomatis. Akibatnya, log waktu kehadiran dari mesin fingerprint tetap harus diekstrak, dicocokkan dengan jadwal shift harian, dan direkapitulasi secara semi-manual menggunakan spreadsheet pada setiap akhir bulan. Kesenjangan teknologi ini menciptakan beban kerja administratif yang sangat berat dan rentan terhadap kesalahan manusia (*human error*). Kesalahan dalam pencatatan kehadiran dan perhitungan hak finansial tidak hanya berdampak pada kerugian operasional, tetapi juga berisiko menurunkan motivasi serta kepuasan kinerja staf medis.

Untuk merealisasikan transisi dari sistem pengolahan data semi-manual menuju platform digital terotomatisasi, RS Efarina Etaham Karawang membutuhkan infrastruktur perangkat lunak yang tangguh. Mengelola presensi medis dan perhitungan kompensasi yang kompleks menuntut antarmuka pengguna (*frontend*) yang responsif serta arsitektur peladen (*backend*) dengan kemampuan komputasi tinggi. Sebagai solusi, tumpukan teknologi berbasis kerangka kerja Next.js dan NestJS dinilai sebagai arsitektur yang paling ideal. Dalam rancang bangun ini, Next.js berperan krusial di sisi klien untuk memastikan perenderan antarmuka berjalan mulus dan interaktif. Di sisi lain, NestJS beroperasi di sisi peladen dengan menawarkan arsitektur modular bawaan yang sangat kompatibel dengan lingkungan *microservices*. Pemilihan kedua kerangka kerja yang bermuara pada bahasa pemrograman *TypeScript* ini menjamin bahwa sistem manajemen SDM RS Efarina Etaham Karawang tidak hanya berkinerja unggul pada fase awal implementasi, tetapi juga menjanjikan stabilitas dan kemudahan skalabilitas seiring ekspansi jumlah karyawan di masa mendatang.

LANDASAN TEORI

Sistem Informasi

Sistem informasi merupakan suatu kumpulan elemen yang saling berhubungan dan berinteraksi untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi guna mendukung pengambilan keputusan serta pengendalian dalam suatu organisasi. (Adham, 2024) mendefinisikan sistem informasi sebagai serangkaian komponen yang bekerja bersama untuk mengumpulkan, mengelola, menyimpan, memproses, dan menyebarkan informasi yang diperlukan untuk mendukung pengambilan keputusan, serta menegaskan bahwa aplikasi berbasis web

merupakan media yang paling banyak digunakan dalam pemanfaatan sistem informasi di era digital saat ini.

Sistem Presensi Karyawan

Presensi karyawan adalah proses administratif pencatatan kehadiran pegawai yang menjadi fondasi penting dalam manajemen sumber daya manusia. Data presensi yang akurat berfungsi sebagai dasar perhitungan gaji, penilaian kinerja, dan evaluasi kedisiplinan. (Pramesti & Febrianto, 2024) menegaskan bahwa implementasi sistem absensi digital terbukti meningkatkan akurasi dan efisiensi pengelolaan data kehadiran dibandingkan metode konvensional berbasis kertas yang masih banyak digunakan di berbagai institusi.

Sistem Penggajian Karyawan

Sistem penggajian adalah rangkaian proses yang mencakup perhitungan dan pendistribusian kompensasi kepada karyawan, meliputi gaji pokok, tunjangan tetap, tunjangan tidak tetap, upah lembur, serta potongan wajib seperti iuran BPJS dan Pajak Penghasilan (PPh 21). (Pamungkas et al., 2025) menegaskan bahwa manajemen penggajian yang efektif dan akurat sangat penting bagi organisasi guna menjamin transparansi serta meminimalkan kesalahan dalam perhitungan gaji yang dapat berujung pada ketidakpuasan karyawan.

Arsitektur *Microservices*

Arsitektur *microservices* adalah pendekatan pengembangan perangkat lunak yang memecah sebuah aplikasi monolitik menjadi sekumpulan layanan kecil yang independen, di mana setiap layanan bertanggung jawab atas satu fungsi bisnis yang spesifik dan berkomunikasi dengan layanan lain melalui antarmuka yang terdefinisi dengan baik, umumnya berupa HTTP REST API. (Nurrokhim et al., 2025) dalam penelitiannya tentang penerapan arsitektur *microservices* pada sistem informasi penjualan online menegaskan bahwa pendekatan ini memberikan kemudahan skalabilitas, di mana layanan dapat ditambah atau dikurangi tanpa mengganggu kinerja sistem secara keseluruhan.

***Framework* Next.js**

Next.js adalah kerangka kerja berbasis *React* yang dikembangkan oleh Vercel, dirancang untuk membangun aplikasi web produksi tingkat tinggi dengan dukungan *Server-Side Rendering* (SSR), *Static Site Generation* (SSG), dan *Incremental Static Regeneration* (ISR) secara bawaan. (Srivastava et al., 2024) dalam kajian komprehensif tentang teknologi Next.js menegaskan bahwa Framework ini terbukti memberikan keunggulan signifikan dalam performa halaman web dibandingkan pendekatan *React* konvensional, khususnya melalui SSR yang menghasilkan halaman yang sudah siap ditampilkan browser tanpa perlu proses rendering di sisi klien.

***Framework* NestJS**

NestJS adalah kerangka kerja Node.js progresif yang sepenuhnya dibangun di atas TypeScript, terinspirasi dari arsitektur Angular dengan menyediakan pola modular berbasis *Module, Controller, Service*, dan *Provider* yang secara inheren mendukung pengembangan aplikasi *server-side* yang skalabel dan mudah dipelihara. NestJS juga mendukung berbagai transport layer untuk komunikasi *microservices*, termasuk HTTP/REST, gRPC, RabbitMQ, MQTT, dan Kafka, menjadikannya kerangka kerja *backend* yang paling fleksibel untuk lingkungan *microservice* (NestJS, 2023).

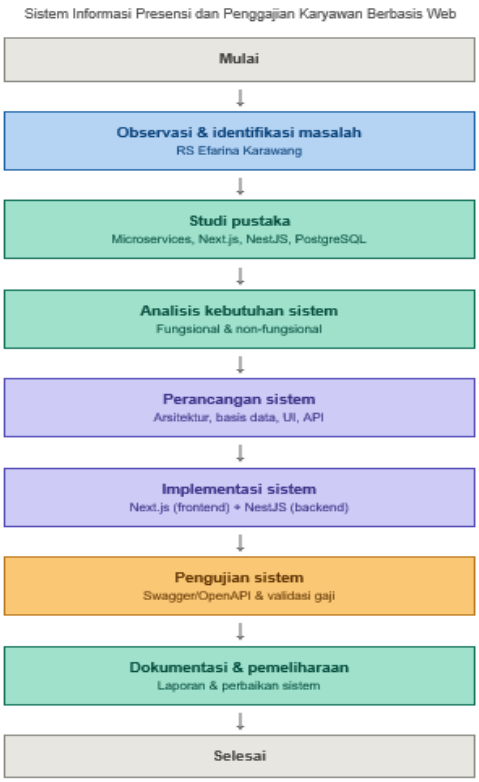
Pengujian Sistem dengan Swagger/OpenAPI

Swagger, yang kini dikenal secara resmi sebagai OpenAPI *Specification*, adalah standar terbuka yang mendefinisikan format antarmuka untuk mendeskripsikan, mendokumentasikan, dan memvisualisasikan layanan web RESTful secara interaktif. OpenAPI *Specification* mendefinisikan antarmuka standar terhadap RESTful API yang memungkinkan baik manusia maupun sistem

komputasi untuk memahami kapabilitas layanan tanpa perlu mengakses kode sumber atau melakukan inspeksi lalu lintas jaringan (Swagger, 2023). Swagger UI menyediakan antarmuka grafis berbasis web yang memungkinkan pengembang mengirim request langsung ke *endpoint* API dan memverifikasi *response*-nya secara *real-time*.

METODE PENELITIAN

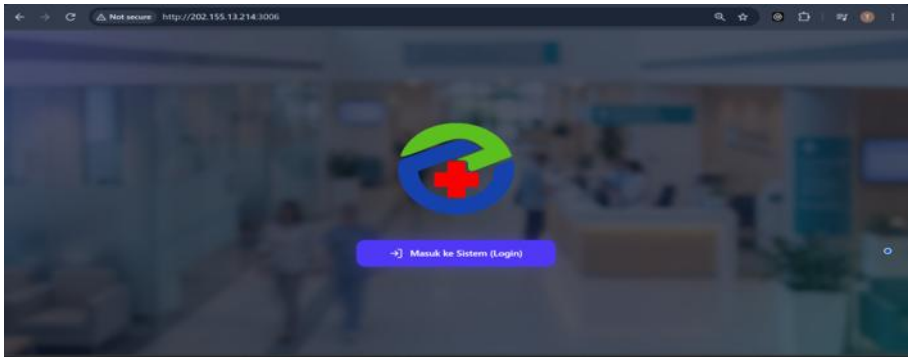
Penelitian ini dilaksanakan di RS Efarina Etaham Karawang yang berlokasi di Kecamatan Telagasari, Kabupaten Karawang, Jawa Barat. Prosedur pengumpulan data yang digunakan Adalah observasi langsung, wawancara, studi dokumentasi, dan studi Pustaka. Adapun alat analisis data yang digunakan adalah perangkat keras (*hardware*) dan perangkat lunak (*software*). Adapun instrumen penelitian yang digunakan adalah wawancara, observasi, dokumen rekapitulasi gaji dan kehadiran, dan pengujian sistem Pemodelan metode dalam penelitian ini mengacu pada arsitektur *microservices* yang diterapkan sebagai fondasi pengembangan sistem. Adapun rancangan penelitian ini adalah sebagai berikut:



Gambar 1. Rancangan Penelitian

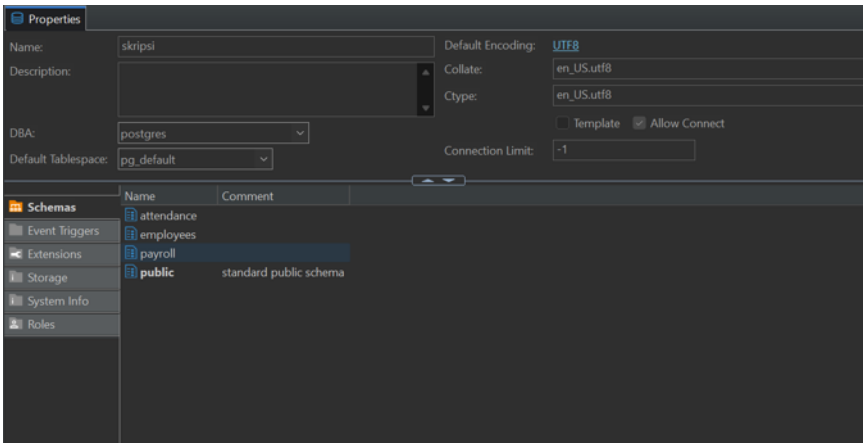
HASIL DAN PEMBAHASAN

Impementasi Sistem

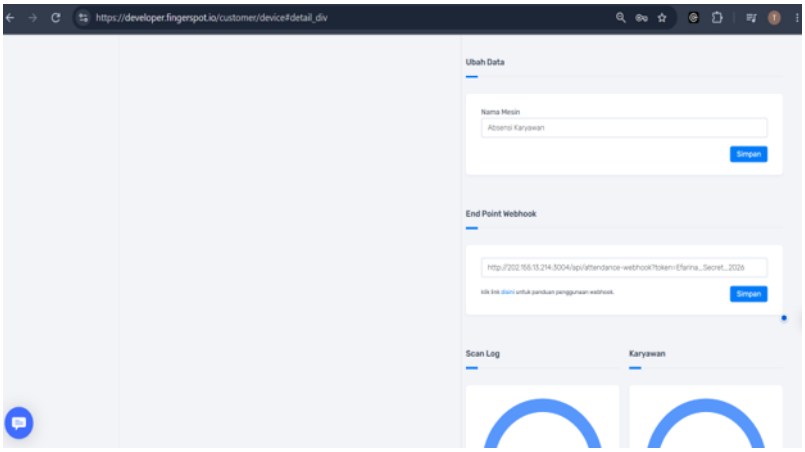


Gambar 2. Gambar tampilan utama sistem

Berdasarkan gambar 2 tersebut, ini merupakan tampilan awal sistem disaat belum melakukan login yang dimana menggunakan http://202.155.13.214:3006.



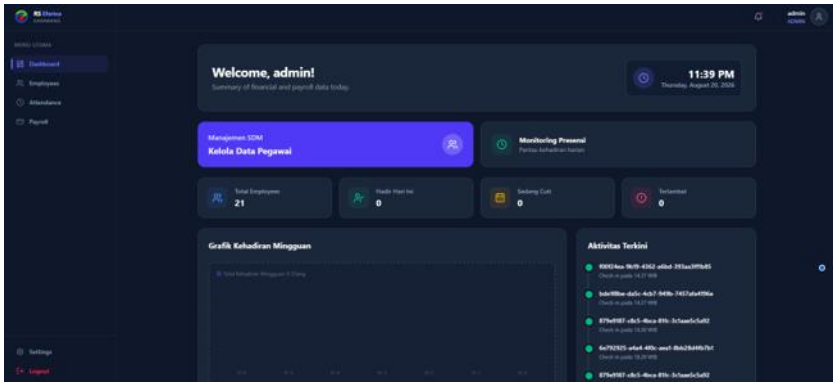
Gambar 3. Database PostgreSQL yang menggunakan schemas



Gambar 4. Konfigurasi *End Point Webhook*

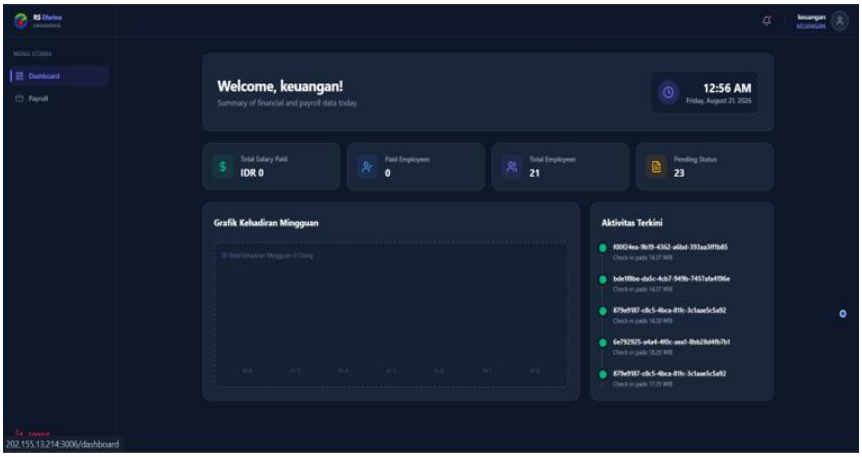
Setiap kali pegawai melakukan scan sidik jari, mesin *Fingerspot* akan mengirimkan kode

status_scan yang menandakan jenis aktivitas presensi yang dilakukan.



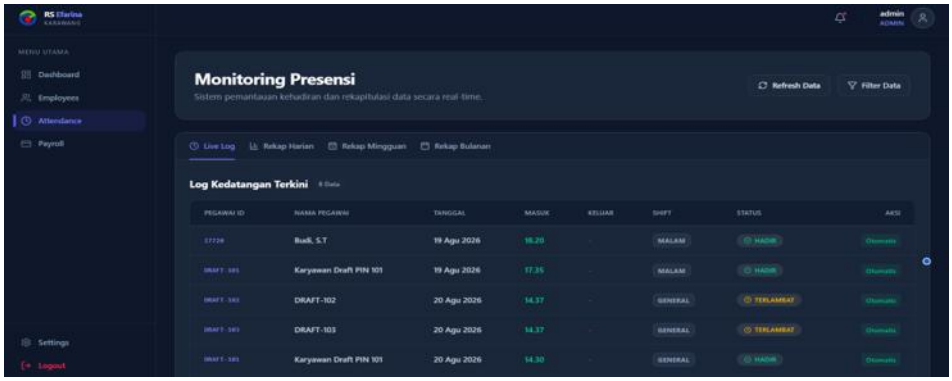
Gambar 5. *Dashboard* Admin

Tampilan *Dashboard Admin* dirancang sebagai pusat kendali utama (*Super Control Center*) dengan akses tanpa batas ke seluruh elemen aplikasi.

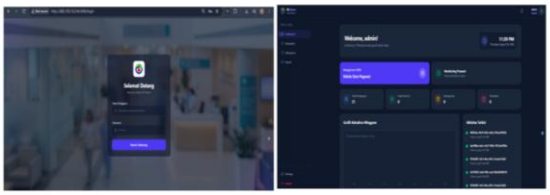


Gambar 6. *Dashboard* SDM

Pada sisi antarmuka pengguna, dashboard presensi diimplementasikan menggunakan teknik *React Query Polling* yang melakukan pengambilan data terbaru dari *server* setiap 5 detik sekali, sehingga data presensi yang ditampilkan pada layar selalu mengikuti kondisi terkini tanpa perlu dilakukan penyegaran (*refresh*) halaman secara manual oleh pengguna. Untuk memastikan nama pegawai dan ID pegawai yang ditampilkan sudah tepat, sistem menerapkan mekanisme dual-lookup mapping, yaitu pencocokan data berdasarkan PIN mesin sekaligus *User* ID pada basis data, sehingga data yang tampil pada dashboard menjadi lebih presisi.

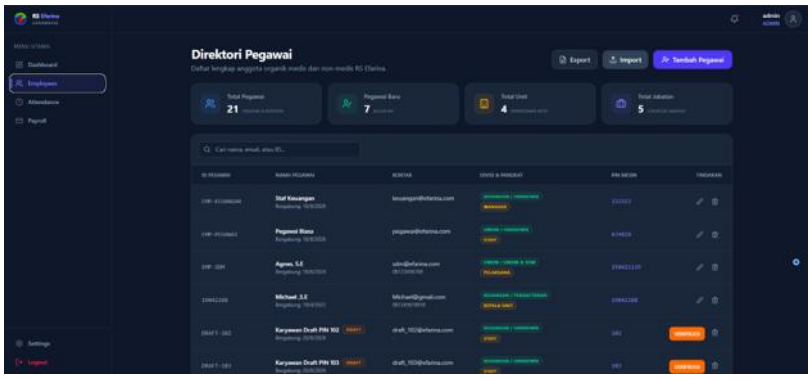


Gambar 7. Monitoring presensi

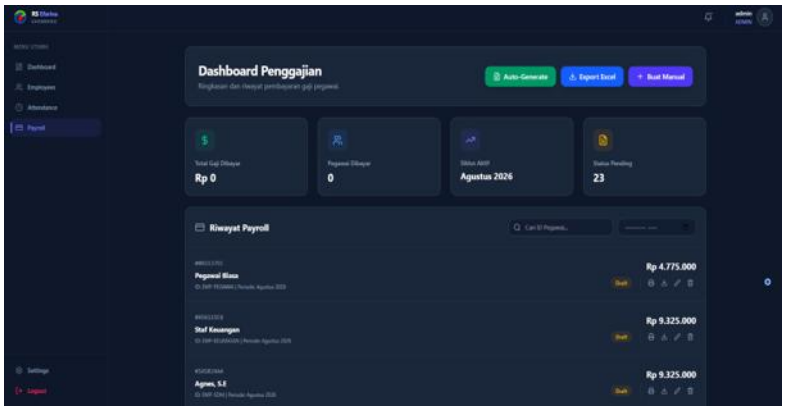


Gambar 8. Halaman *Login* dan *Dashboard* Utama

Halaman *Login* dan *Dashboard* Utama, yang menampilkan ringkasan informasi presensi dan penggajian sesuai dengan hak akses pengguna yang sedang login. Halaman Penggajian (*Payroll*) dan Ekspor Laporan, yang digunakan untuk melihat, menghitung, serta mengunduh laporan gaji pegawai dalam format PDF maupun Excel. Halaman Master Data Karyawan, yang digunakan untuk mengelola data pegawai, departemen, dan jabatan.



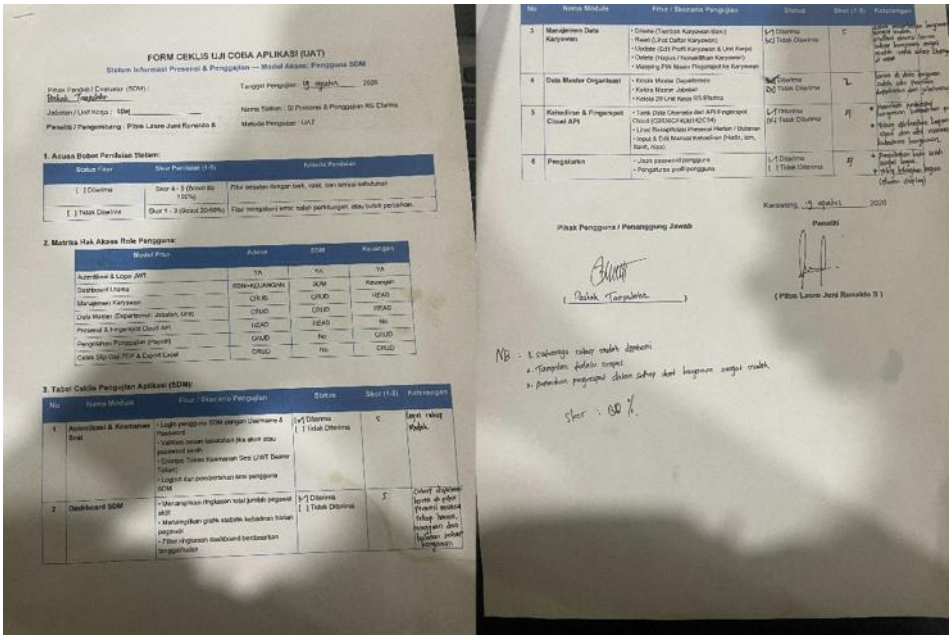
Gambar 9. Halaman Monitoring



Gambar 10. Halaman Penggajian dan Ekspor Laporan

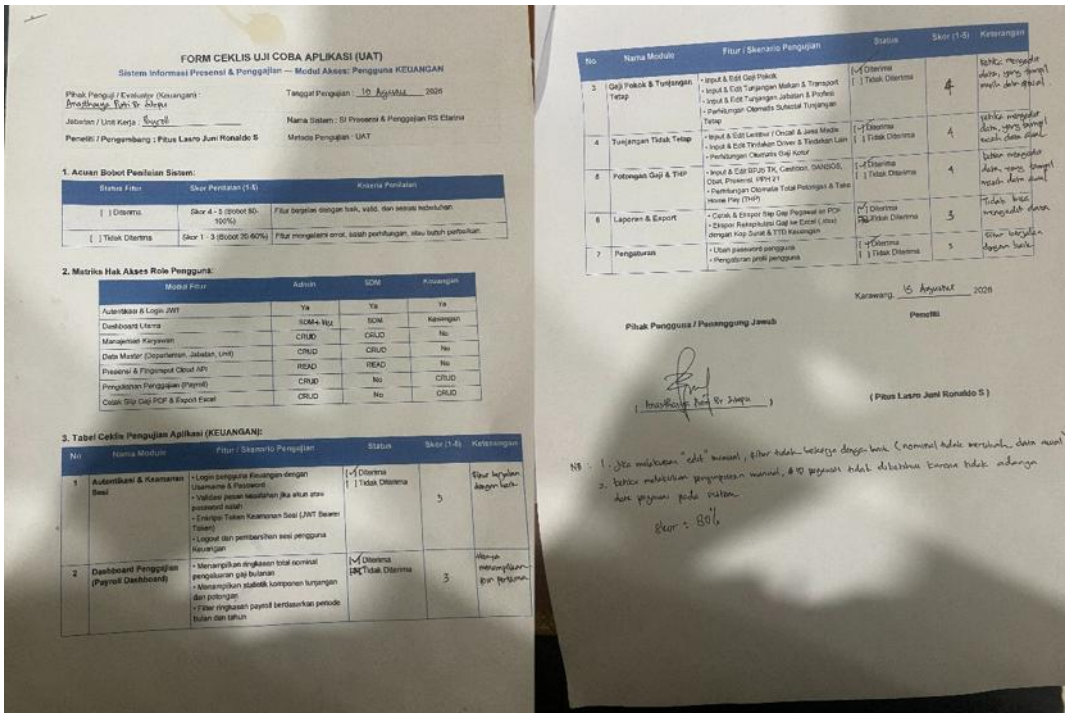
Pengujian Sistem

Setelah tahap implementasi selesai dilakukan, tahap selanjutnya adalah pengujian sistem yang bertujuan untuk memastikan bahwa seluruh fitur yang telah dibangun dapat berjalan sesuai dengan fungsinya masing-masing serta memenuhi kebutuhan pengguna. Pengujian pada penelitian ini dibagi menjadi dua bagian, yaitu pengujian fungsional dan pengujian integrasi *real-time* beserta performa *webhook*.



Gambar 11. Hasil pengujian UAT pada user SDM

Berdasarkan hasil pengujian fungsional pada gambar 11, dapat disimpulkan bahwa seluruh skenario pengujian yang dilakukan menunjukkan hasil yang valid dan sesuai dengan hasil yang diharapkan, sehingga sistem dinyatakan telah memenuhi kebutuhan fungsional yang telah dirumuskan pada tahap analisis kebutuhan.walaupun terdapat banyak kekurangan yang di miliki oleh sistem saya ini.



Gambar 12. Hasil pengujian UAT pada user KEUANGAN

Berdasarkan hasil pengujian fungsional pada gambar 12, dapat disimpulkan bahwa seluruh skenario pengujian yang dilakukan menunjukkan hasil yang valid dan sesuai dengan hasil yang diharapkan, sehingga sistem dinyatakan telah memenuhi kebutuhan fungsional yang telah dirumuskan pada tahap analisis kebutuhan.

Hasil pengujian menunjukkan bahwa keseluruhan proses tersebut, mulai dari mesin sidik jari melakukan pengiriman data hingga data tersebut tampil dan diperbarui pada layar *dashboard web frontend*, tercatat memiliki *latency* lebih dari 1 (satu) detik. Dengan demikian, dapat disimpulkan bahwa integrasi real-time pada sistem ini telah berjalan dengan baik dan mampu memberikan informasi presensi yang aktual (*up to date*) kepada pengguna.

KESIMPULAN

Hasil pengujian fungsional maupun pengujian integrasi *real-time* yang telah dilakukan menunjukkan bahwa sistem yang dibangun telah berjalan sesuai dengan kebutuhan dan tujuan penelitian yang telah dirumuskan sebelumnya. Guna pengembangan dan penyempurnaan sistem pada penelitian selanjutnya, penulis memberikan beberapa saran. Pertama, pengembangan aplikasi *mobile* berbasis presensi *geolocation* GPS. Fitur ini dapat ditambahkan agar pegawai yang bertugas di luar kantor atau bekerja secara *remote* tetap dapat melakukan presensi melalui smartphone berdasarkan titik koordinat lokasi meerkat. Kedua, integrasi notifikasi otomatis melalui *WhatsApp* atau Email Bot. Fitur ini dapat dikembangkan untuk mengirimkan pengingat kepada pegawai yang belum melakukan presensi, maupun mengirimkan bukti presensi harian secara otomatis langsung ke *WhatsApp* masing-masing pegawai. Terakhir, penerapan mekanisme caching menggunakan Redis. *Redis Caching* dapat diterapkan pada API *Gateway* maupun *Attendance Service* untuk mempercepat proses rekap data presensi bulanan, khususnya apabila jumlah data pegawai yang diproses sudah mencapai skala ribuan pegawai.

DAFTAR REFERENSI

Adham, M. F. (2024). Analisis Implementasi Sistem Informasi: Studi Literatur Analysis Of Information System Implementation: Literature Review. *Jurnal Teknologi Sistem Informasi*, *5*(1), 264–275. https://doi.org/10.35957/jtsi.v5i1.7815

Ahmed, A. M., Mohammed, C. N., & Ahmad, A. M. (2023). Web-based payroll management system: design, implementation, and evaluation. *Journal of Electrical Systems and Information Technology*, *10*(1), 17. https://doi.org/10.1186/s43067-023-00082-5

Meenugu, S. (2025). Automating Complex Payroll Systems in Healthcare: A Technical Implementation Study. *International Journal of Management Technology*, *12*(3), 56–66. https://doi.org/10.37745/ijmt.2013/vol12n35666

Mhatre, A. L. (2023). Microservices Architecture for Healthcare. *Journal of Artificial Intelligence*, *1*(4), 1590–1592. https://doi.org/10.51219/jaimld/anand-laxman-mhatre/356

NestJS. (2023). *Documentation | NestJS - A progressive Node.js Framework*.

Nurrokhim, R. F., Siregar, F. F., Firdaus, F. Al, Aditiya, R., Yahya, F. M., Pohan, L., & Pratama, R. S. (2025). Penerapan Arsitektur Microservices pada Sistem Informasi Penjualan Online untuk Meningkatkan Skalabilitas dan Kinerja Sistem. *Jurnal Teknik Mesin, Industri, Elektro Dan Informatika*, *4*(3), 37–47. https://doi.org/10.55606/jtmei.v4i3.5825

Pamungkas, R., Pratiwi, N. A., Sayyidah, N., & Prawira, M. A. R. (2025). Perancangan Sistem Informasi Penggajian Karyawan Berbasis Web Pada PT. Bangun Sumber Daya Mandiri. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *9*(4), 6036–6042. https://doi.org/10.36040/jati.v9i4.13960

Pramesti, S., & Febrianto, P. T. (2024). Implementasi Sistem Absensi Digital untuk Meningkatkan Efisiensi Pencatatan Kehadiran Guru di Sekolah Dasar. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(2), 2429–2434. https://doi.org/10.36040/jati.v8i2.9521

Srivastava, S., Shukla, H., Landge, N., Srivastava, A., & Jindal, D. (2024). A Comprehensive Review of Next.js Technology: Advancements, Features, and Applications. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4831070

Swagger. (2023). *Swagger Documentation*.

Tandrio, F., & Fianty, M. I. (2026). Web-Based Payroll System Development Using The Prototyping Method and Structured Database Design. *JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer)*, *11*(3), 851–863. https://doi.org/10.33480/jitk.v11i3.7044

Velepucha, V., & Flores, P. (2023). A Survey on Microservices Architecture: Principles, Patterns and Migration Challenges. *IEEE Access*, *11*, 88339–88358. https://doi.org/10.1109/ACCESS.2023.3305687