

PENGEMBANGAN MODUL *PAYROLL*: INTEGRASI *FINGERPRINT* DAN OTOMASI PENGHITUNGAN GAJI BERBASIS JABATAN DENGAN KINDERFIN

Yusuf Hasan N.¹, Adhatus Solichah A.², dan Kelly Rossa S.³

^{1, 2, 3}Departemen Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: 5025211225@student.its.ac.id⁽¹⁾, adhatus@its.ac.id⁽²⁾, kelly@its.ac.id⁽³⁾

Abstrak—*Payroll* adalah sistem penting dalam pengelolaan keuangan organisasi karena berperan memastikan pembayaran gaji dilakukan dengan tepat, akurat, dan efisien. Proses manual penggajian sering kali menimbulkan risiko kesalahan perhitungan dan memakan waktu lebih lama, sehingga dikembangkan sebuah sistem *payroll* berbasis digital yang mampu mengunggah dan mengotomatisasi perhitungan gaji berdasarkan kehadiran, tunjangan, dan potongan. Penelitian ini bertujuan menganalisis pengaruh penerapan sistem *payroll* otomatis terhadap efisiensi dan akurasi proses penggajian; sistem ini dirancang dengan modul pencatatan kehadiran terintegrasi teknologi *fingerprint* untuk pencatatan data pegawai secara *real-time* dan akurat, yang kemudian diproses untuk menghitung gaji guru, termasuk komponen tunjangan, insentif, potongan keterlambatan, dan pajak penghasilan. Model perhitungan ini diimplementasikan menggunakan algoritma untuk meminimalisir kesalahan perhitungan dan memastikan transparansi data penggajian. Proses pengelolaan gaji yang sebelumnya memerlukan waktu sekitar 30 menit secara manual, kini dapat diselesaikan secara otomatis, di mana perhitungan gaji menjadi langsung tersedia dengan interaksi pengguna yang minimal.

Kata Kunci—*Payroll*, Sistem Otomasi, Kehadiran *Fingerprint*, Efisiensi, Transparansi.

I. PENDAHULUAN

SISTEM *payroll* merupakan hal krusial dalam pengelolaan keuangan suatu instansi atau sekolah, memastikan pembayaran gaji yang tepat, akurat, dan efisien. Kehadiran pegawai, yang kini mayoritas dicatat melalui teknologi sistem *fingerprint*. *Fingerprint* adalah tanda dari suatu orang yang tidak dapat ditiru atau disamakan dengan orang lain. *Fingerprint* menjadi suatu pengenalan dan tanda di berbagai aspek kehidupan seperti pendidikan, pegawaian dan bidang lainnya. *Fingerprint* merupakan salah satu ciri dari seseorang sehingga setiap orang pasti mempunyai sidik jari yang berbeda dengan orang yang lainnya dan tidak mungkin ada sistem sidik jari yang sama pada orang lain [14]. Menggunakan sistem absensi biometrik *fingerprint* mengurangi masalah yang timbul dari penggunaan sistem absensi manual [13], masalah yang timbul biasanya kesalahan perhitungan dan memakan waktu lebih lama. Hal ini juga mengakibatkan kurangnya transparansi, bahkan berpotensi manipulasi komponen gaji lainnya. Terutama di lembaga pendidikan, lalu lintas keuangan sangat bergantung pada transparansi.

Oleh karena itu, penelitian ini berfokus pada pengembangan sistem *payroll* berbasis digital yang mengintegrasikan mesin *fingerprint* untuk perhitungan gaji otomatis berdasarkan kehadiran. Tujuannya adalah meminimalkan kesalahan dalam penghitungan nominal gaji, meningkatkan efisiensi pengelolaan data, dan memastikan transparansi data. Sistem ini dirancang untuk memungkinkan guru memverifikasi rincian gaji secara langsung, sementara pihak administratif dapat melakukan perhitungan otomatis, serta pemilik yayasan dapat memantau keuangan secara transparan. Mesin *fingerprint* yang digunakan secara spesifik adalah model "*Fingerspot* Revo WF-206BNC". Selain itu, sistem ini akan mencakup pembagian peran yang lebih rinci, seperti Kepala Sekolah, Guru, dan Bendahara. Ruang lingkup sistem ini terbatas pada penghitungan *payroll* berdasarkan kehadiran yang dicatat melalui *fingerprint* dan fleksibilitas kustomisasi item pembentuk gaji.

Dengan demikian, implementasi sistem *payroll* berbasis digital ini diharapkan mampu menggantikan pengelolaan manual. Proses pengelolaan gaji yang sebelumnya memerlukan waktu sekitar 30 menit secara manual, kini dapat diselesaikan secara otomatis, di mana perhitungan gaji menjadi langsung tersedia dengan interaksi pengguna yang minimal. Hal ini memperlihatkan potensi signifikan sistem dalam memastikan data kehadiran yang akurat dan valid, sehingga perhitungan gaji menjadi lebih transparan dan dapat dipertanggungjawabkan.

II. TINJAUAN PUSTAKA

A. Sistem Payroll

Payroll adalah proses pembayaran yang diberikan kepada pegawai sebagai imbalan atas pegawaian yang telah mereka lakukan. Secara lebih spesifik, gaji adalah bentuk kompensasi finansial yang merupakan hak setiap pegawai, diterima dalam bentuk uang dari pemberi kerja atau instansi. Pembayaran gaji ini diatur sesuai dengan kontrak kerja yang telah disepakati, baik antara pegawai dan pemberi kerja, maupun berdasarkan ketentuan hukum yang berlaku. Di dalamnya juga termasuk berbagai tunjangan, seperti tunjangan untuk pegawai dan keluarganya, yang merupakan bagian dari imbalan atas kontribusi pegawai dalam melaksanakan tugas-tugas yang telah diselesaikan atau akan dilakukan di masa depan [3].

B. Interoperabilitas

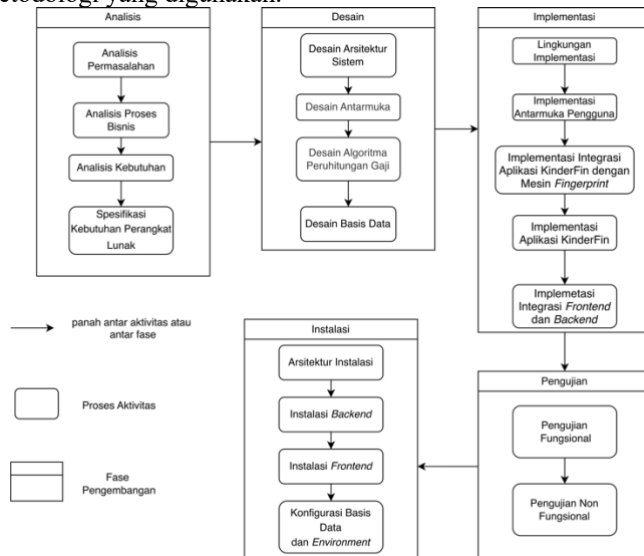
Interoperabilitas merupakan kemampuan dua atau lebih sistem untuk bertukar informasi dan memanfaatkan informasi yang telah dipertukarkan dengan mengikuti standar umum yang telah disepakati [7]. Interoperabilitas pada Tugas Akhir ini bertujuan sebagai alat pengujian untuk menguji Aplikasi KinderFin bertukar informasi dengan mesin *fingerprint* melalui server.

C. Arsitektur Web

Arsitektur web merujuk pada cara komponen-komponen dalam aplikasi web saling berinteraksi. Dalam pengembangan modul *payroll*, arsitektur yang baik mencakup pemisahan antara *frontend*, *backend*, *database*, dan *Application Programming Interface API*. Hal ini memungkinkan sistem *payroll* untuk menangani banyak pengguna dan menyediakan informasi secara *real-time*, termasuk data kehadiran dan perhitungan gaji. Struktur yang modular ini juga memberikan keleluasaan dalam pengembangan, di mana *backend* menangani logika bisnis dan penyimpanan data, sementara *frontend* berfokus pada antarmuka pengguna [2].

III. METODE PENELITIAN

Metode penelitian yang diterapkan dalam pengembangan aplikasi tugas akhir ini adalah model *Software Development Life Cycle (SDLC) Waterfall*, yang dikenal dengan pendekatannya yang sekuensial, memastikan setiap fase pengembangan diselesaikan tuntas sebelum melangkah ke fase berikutnya. Proses ini secara berurutan mencakup analisis, desain, implementasi, pengujian, dan diakhiri dengan instalasi. Gambar 1 memberikan gambaran menyeluruh tentang tahapan metodologi yang digunakan.

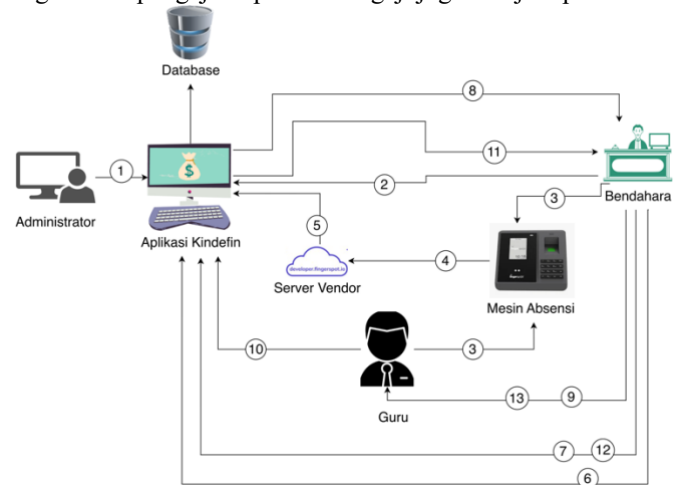


Gambar 1 Alur Diagram Pengembangan Modul Penggajian Aplikasi KinderFin 2.0

A. Analisis

Tahap analisis meliputi identifikasi permasalahan, analisis proses bisnis, dan kebutuhan sistem. Aplikasi KinderFin, yang menjadi dasar pengembangan, merupakan aplikasi yang telah

tercatat sebagai kekayaan intelektual [6]. Permasalahan utama yang diidentifikasi pada versi awalnya (KinderFin 1.0) adalah keterbatasan, kesalahan, dan kurangnya transparansi modul penggajian manual, yang tidak terintegrasi dengan data kehadiran atau digitalisasi pengajuan gaji. Sebagai solusi, KinderFin 2.0 dirancang untuk *payroll* transparan (gaji pokok, tunjangan, potongan, bonus), terintegrasi *fingerprint* untuk perhitungan otomatis, dan mendukung impor data kehadiran. Digitalisasi pengajuan perubahan gaji juga menjadi prioritas.



Gambar 2 Proses Bisnis Modul Penggajian Aplikasi KinderFin 2.0

Proses bisnis pada Gambar 2 melibatkan interaksi dengan server *Fingerspot* via API untuk data kehadiran. Administrator menambahkan pengguna, lalu Bendahara mengatur gaji, nominal, dan potongan. Data presensi dikirim otomatis atau diunggah atau *input* manual jika terkendala. Ini menghasilkan detail gaji per hari, yang divalidasi dan didistribusikan Bendahara. Pengguna dapat mengajukan perubahan gaji untuk divalidasi Bendahara. Pada Tabel 1 merupakan detail nama tabel dan aktor yang terlibat dalam setiap tugas.

Tabel 1 Penjelasan Proses Bisnis Modul Penggajian Aplikasi KinderFin 2.0

Nomor Alur Proses	Nama Alur Proses	Aktor
1	Tambah Pengguna	Administrator
2	Aktifkan label Gaji, Tambah atau ubah gaji harian, Pengaturan Potongan Gaji Harian	Bendahara
3	Presensi di Mesin <i>fingerprint</i>	Guru, Bendahara
4	Kirim data presensi ke <i>server vendor (fingerspot)</i>	Mesin Presensi
5	Teruskan data presensi dari vendor (<i>fingerspot</i>) ke Aplikasi KinderFin	Server Vendor (<i>fingerspot</i>)
6	Unggah <i>file</i> presensi	Bendahara
7	Unggah presensi manual	Bendahara
8	Validasi gaji	Bendahara
9	Terima gaji	Guru

Nomor Alur Proses	Nama Alur Proses	Aktor
10	Pengajuan perubahan gaji	Guru
11	Validasi pengajuan perubahan gaji	Bendahara
12	Unggah presensi manual perubahan gaji	Bendahara
13	Terima gaji setelah pengajuan perubahan gaji	Guru

Tiga aktor utama Administrator, Bendahara, dan Guru memiliki kebutuhan fungsional berbeda. Administrator mengelola pengguna dan hak akses. Bendahara mengelola seluruh pengaturan gaji, presensi, validasi, cetak gaji, bonus, dan log aktivitas. Guru dapat melakukan presensi, melihat rincian gaji, dan mengajukan perubahan.

B. Desain

Pada tahap desain, perancangan meliputi beberapa aspek fundamental, yaitu arsitektur sistem, antarmuka pengguna (UI/UX), algoritma perhitungan gaji, dan basis data. Perancangan antarmuka ini memegang peranan krusial untuk memastikan penyajian data yang jelas serta kemudahan penggunaan bagi seluruh peran pengguna, yang mencakup administrator, bendahara, maupun guru. Untuk mencapai tujuan tersebut, arsitektur sistem mengadopsi model "*Clean Architecture*". *Clean Architecture* didefinisikan sebagai suatu pendekatan rekayasa perangkat lunak yang menekankan pemisahan antar lapisan aplikasi secara jelas, yang bertujuan untuk meningkatkan kemudahan pemeliharaan, pengujian, dan pengembangan [11]. Dalam implementasinya pada aplikasi ini, arsitektur tersebut menstrukturkan sistem ke dalam empat lapisan utama yaitu, lapisan *Domain* sebagai fondasi aturan bisnis, lapisan *Application* yang bertanggung jawab atas eksekusi logika bisnis seperti perhitungan gaji, lapisan *Infrastructure* yang menangani implementasi teknis seperti konektivitas basis data, dan lapisan *Presentation* yang berfungsi sebagai gerbang interaksi antara pengguna dengan sistem.

Hasil dari perancangan ini adalah beberapa halaman utama yang dibuat, seperti *dashboard* untuk bendahara sebagai halaman utama, halaman untuk mengatur komponen gaji, halaman data jabatan, serta halaman untuk melihat rincian gaji. Guru juga memiliki dasbor pribadi untuk informasi gaji. Untuk membangun Aplikasi KinderFin 2.0, digunakan *framework full-stack*. Bagian *backend* (sisi server) memakai *framework* Express.js [4] yang berjalan di atas Node.js [12], sedangkan bagian *frontend* (tampilan pengguna) memakai Next.js [8]. Dengan pendekatan ini, aplikasi bisa dibuat lebih dinamis dan cepat, sekaligus mempermudah proses penyamaan data absensi dari mesin sidik jari dan data gaji dengan Aplikasi KinderFin 2.0 [10].

Algoritma perhitungan gaji dirancang untuk mengotomatisasi proses secara penuh dan meminimalkan kesalahan. Tahapan utamanya meliputi persiapan data *input*,

pemrosesan jam masuk dan jam pulang untuk deteksi keterlambatan atau *pulang cepat* dengan penerapan potongan berjenjang, dan penghitungan gaji final setelah potongan serta bonus opsional. Basis data aplikasi menggunakan *PostgreSQL*, dirancang dengan tabel-tabel inti untuk modul penggajian, mencakup *master_jabatan*, *master_jabatan_pokok*, *activity_logs*, *detail_salary*, *pengajuan_perubahan_gaji*, *pengaturan_gaji_aktif*, *potongan_keterlambatan*, dan *rekap_bonus*.



Gambar 3 Diagram *Physical Data Model* (PDM) Aplikasi KinderFin 2.0 Modul Penggajian

Untuk lebih memperjelas struktur basis data yang telah dirancang, Gambar 3 menyajikan visualisasi dari *Physical Data Model* (PDM) modul penggajian. PDM ini mengilustrasikan entitas, atribut, tipe data, serta hubungan antar tabel dalam basis data, yang menjadi fondasi penyimpanan dan pengelolaan data aplikasi KinderFin 2.0. Setiap tabel didesain dengan *Primary Key* (PK) untuk identifikasi unik *record* data dan *Foreign Key* (FK) diimplementasikan untuk membangun konektivitas antar tabel, mereferensikan *Primary Key* di tabel lain, yang sebagian besar membentuk relasi *one to many* (1:M).

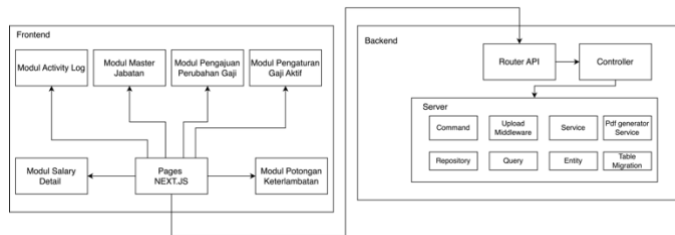
C. Implementasi

Pada tahap implementasi, semua rancangan diwujudkan menjadi aplikasi nyata menggunakan perangkat keras MacBook Air M1 serta perangkat lunak seperti Visual Studio Code, Postman, Neon Database, Render, Vercel, dan GitHub.

Di sisi *frontend*, dilakukan penyesuaian dari aplikasi KinderFin 1.0 yang dikembangkan sebelumnya [9]. Beberapa fitur disempurnakan dan tampilan baru dibuat, seperti halaman Tambah, Edit, dan Hapus Pengguna untuk Administrator. Halaman lain yang dikembangkan adalah Pengaturan Gaji Aktif, Nominal Gaji (Master Jabatan), dan Potongan Gaji. Selain itu, dibuat juga halaman Rekap Gaji Pegawai yang memungkinkan tarik data presensi, proses unggah *file* presensi,

input manual, dan verifikasi rekap gaji. Sistem juga menangani jenis foto bukti yang diunggah oleh guru saat mengajukan perubahan gaji.

Sementara itu, di sisi *backend*, logika programnya mencakup modul-modul baru seperti Pengaturan Gaji Aktif, Master Jabatan, Potongan Keterlambatan, Salary Detail, Rekap Bonus, dan Pengajuan Perubahan Gaji. Namun, khusus untuk modul *Users*, pengembangannya mengacu pada API KinderFin 1.0 yang sudah ada [1]. REST API adalah teknologi yang digunakan untuk menghubungkan suatu aplikasi dengan aplikasi lainnya melalui komunikasi *Representational State Transfer* (REST) [5]. Pada akhirnya, *frontend* dan semua modul *backend* tersebut diintegrasikan agar sistem dapat berfungsi secara utuh.



Gambar 4 Ilustrasi Integrasi Frontend dan Backend Aplikasi KinderFin 2.0

Gambar 4 menunjukkan bagaimana kedua komponen ini saling terhubung. Pengujian dilakukan melalui *blackbox testing* untuk fungsionalitas, serta pengujian interoperabilitas untuk konektivitas dengan server *developer.fingerspot.io*. Instalasi aplikasi menggunakan arsitektur *cloud-native*, dengan *frontend* di *Vercel*, *backend* di *Render*, dan basis data *PostgreSQL* di *Neon*, termasuk konfigurasi variabel *environment*.

D. Pengujian

Setelah implementasi, pengujian dilakukan untuk memverifikasi fungsionalitas dan pemenuhan kebutuhan pengguna. Pengujian ini terbagi menjadi metode fungsional dan non-fungsional. Pengujian fungsional berfokus pada validasi kebutuhan pengguna, menggunakan metode *blackbox testing* untuk memeriksa kesesuaian fitur dengan alur normal dan alternatif, serta terpenuhinya kondisi awal dan akhir setiap kasus penggunaan. Sementara itu, pengujian non-fungsional, khususnya interoperabilitas, melibatkan koneksi Aplikasi KinderFin 2.0 dengan sistem lain, yaitu server *developer.fingerspot.io*. Pengujian ini memastikan data dapat diambil dari server *fingerprint* dan diproses oleh aplikasi, termasuk simulasi penerimaan data berformat JSON dari server *developer.fingerspot.io*. Simulasi ini diperlukan karena adanya kendala dalam pengambilan data langsung dari server *developer.fingerspot.io* ke mesin *fingerprint* di lapangan.

E. Instalasi

Setelah pengujian, proses instalasi dilakukan agar aplikasi dapat diakses publik. Instalasi menggunakan arsitektur *cloud-native* dengan pemisahan komponen *frontend*, *backend*, dan basis data untuk skalabilitas dan keandalan. *Frontend* diinstal menggunakan platform *Vercel*, sementara *backend* di *Render* sebagai *hosting server*. Basis data Aplikasi KinderFin 2.0

menggunakan *Neon* sebagai platform *PostgreSQL cloud*. Instalasi *backend* melibatkan perintah *npm install*, *npm run build*, dan *npm run start*, dengan penyambungan akun *GitHub* ke *Render*. Instalasi *frontend* juga menggunakan *Vercel* dengan perintah serupa (*yarn build*, *yarn start*) dan penyambungan *GitHub*. Konfigurasi *environment variables* (seperti *DB_HOST*, *JWT_SECRET_KEY*, *IMAGEKIT_PUBLIC_KEY* untuk *backend* dan *NEXT_PUBLIC_API_URL* untuk *frontend*) diperlukan untuk komunikasi basis data dan API.

IV. UJI COBA DAN EVALUASI

Setelah melakukan pengujian pada Aplikasi KinderFin 2.0 dengan menggunakan dua metode, yaitu *blackbox testing* untuk aspek fungsional dan pengujian interoperabilitas untuk aspek non-fungsional, ab ini akan membahas hasil serta evaluasi dari pengujian tersebut. Uji coba fungsional dilakukan dengan mencocokkan kesesuaian fitur dengan alur penggunaan yang telah ditentukan, sementara pengujian interoperabilitas berfokus pada konektivitas aplikasi dengan sistem eksternal.

A. Hasil Uji Coba Fungsional

Pada bagian Hasil Uji Coba Fungsional, menunjukkan keberhasilan signifikan di seluruh peran pengguna. *Administrator* (Pengguna Uji Coba 1) dan *Guru* (Pengguna Uji Coba 3) mencapai nilai rata-rata 10 pada kuesioner pengujian fungsional, menandakan aplikasi sepenuhnya memenuhi kebutuhan mereka. Sementara itu, *Bendahara* (Pengguna Uji Coba 2) mendapatkan nilai rata-rata 9,1 untuk seluruh pengujian, meskipun satu kasus yaitu Kasus Pengujian Ambil Data Langsung dari server *developer.fingerspot.io* dinyatakan Gagal.

B. Hasil Uji Coba Non Fungsional

Pada bagian Hasil Uji Coba Non Fungsional, pengujian difokuskan pada kemampuan sistem mengolah data presensi dari server eksternal. Pengujian pengambilan data langsung dari server *developer.fingerspot.io* mengalami kegagalan karena adanya perbedaan Nomor Seri antara mesin *fingerprint* yang digunakan di lokasi pengujian dan Nomor Seri yang terdaftar di server *developer.fingerspot.io*. Ketidakcocokan ini menyebabkan server *developer.fingerspot.io* tidak dapat melanjutkan perintah, sehingga aplikasi tidak menerima respons atau data dan menampilkan pesan 'Respons dari API Fingerspot tidak valid'. Bukti terdapat pada LAMPIRAN. Meskipun demikian, kendala ini dapat diatasi dengan metode unggah *file* presensi manual dari mesin *fingerprint*, yang ditunjukkan berhasil dengan pesan sukses, atau *input data manual*.

C. Pembahasan Hasil Uji Coba Fungsional

Data dari kuesioner pengujian menunjukkan bahwa Aplikasi KinderFin 2.0 secara fungsional telah memenuhi kebutuhan pengguna. Nilai sempurna (10/10) untuk peran *Administrator* dan *Guru* mengonfirmasi bahwa fitur-fitur yang dirancang untuk mereka berfungsi sesuai harapan. Nilai 9,1 untuk peran *Bendahara* juga menunjukkan tingkat keberhasilan yang tinggi, dengan satu-satunya kendala terletak pada kasus integrasi yang

gagal secara teknis. Dari sisi efisiensi, evaluasi menunjukkan bahwa sistem ini berhasil mengurangi waktu pengelolaan gaji secara drastis. Proses yang sebelumnya memerlukan waktu 30 menit secara manual, kini dapat diselesaikan dalam 5 menit menggunakan aplikasi, atau terjadi peningkatan efisiensi sebesar 83,33%. Selain itu, sistem juga mampu mengurangi kesalahan pencatatan gaji hingga 95%.

D. Pembahasan Hasil Uji Coba Interoperabilitas

Kegagalan dalam uji coba interoperabilitas untuk penarikan data secara langsung disebabkan murni oleh ketidakcocokan data Nomor Seri mesin. Hal ini menyebabkan server *developer.fingerspot.io* tidak dapat melanjutkan perintah, sehingga aplikasi menampilkan pesan "Respons dari API *Fingerspot* tidak valid". Meskipun demikian, sistem terbukti mampu mengolah data presensi jika data tersebut berhasil diterima, seperti yang ditunjukkan pada pengujian simulasi. Kendala ini menunjukkan bahwa logika aplikasi sudah tepat, dan masalah terletak pada konfigurasi antara perangkat keras dan server vendor. Masalah ini dapat diatasi dengan metode alternatif yang telah disediakan dalam sistem, yaitu fitur unggah file presensi atau input data manual, yang keduanya terbukti berhasil.

V. KESIMPULAN DAN SARAN

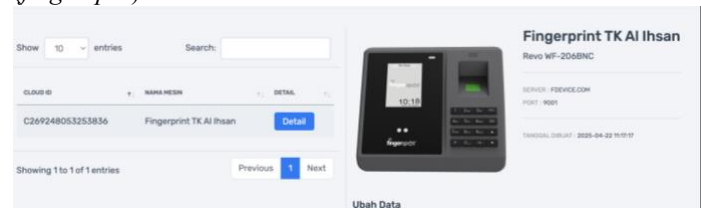
Pengembangan modul payroll pada Aplikasi KinderFin 2.0 telah membuktikan bahwa otomatisasi dapat meningkatkan efisiensi dan transparansi dalam manajemen keuangan sekolah secara mendasar. Pentingnya hasil penelitian ini tidak hanya terletak pada keberhasilan teknis integrasi sistem dengan *fingerprint Fingerspot*, tetapi juga pada hasil terukur yang dicapai, yaitu peningkatan efisiensi kerja Bendahara sebesar 83,33% dan penurunan kesalahan pencatatan gaji hingga 95%. Keberhasilan ini didukung oleh fitur-fitur seperti modul "Master Jabatan" dan "Potongan Keterlambatan" yang memastikan perhitungan gaji berjalan akurat sesuai jabatan dan aturan yang berlaku. Walaupun ditemukan kendala pada integrasi data secara langsung, hal tersebut dapat diatasi melalui simulasi pengolahan data mentah dari respons API menjadi nominal gaji. Keberhasilan dalam mengatasi kendala ini mendukung temuan bahwa sistem tetap berfungsi baik dan diterima oleh pengguna, yang ditunjukkan melalui skor kesesuaian rata-rata 9,7 dari 10. Hal ini menegaskan bahwa KinderFin 2.0 merupakan solusi menyeluruh yang telah berjalan sesuai harapan.

Untuk pengembangan Aplikasi KinderFin selanjutnya, ada beberapa saran yang dapat menjadi prioritas. Pertama, penyempurnaan integrasi langsung dengan *Fingerspot* perlu dilakukan untuk mengatasi masalah perbedaan nomor seri antara mesin dan server, dengan tujuan agar penarikan data kehadiran dapat berjalan otomatis tanpa perlu tindakan manual. Kedua, berdasarkan masukan pengguna, perlu dilakukan peningkatan pada tampilan antarmuka (UI) dan pengalaman pengguna (UX), khususnya untuk peran Bendahara, agar alur kerja menjadi lebih mudah dipahami. Ketiga, disarankan untuk menambah fitur notifikasi langsung (*real-time*) untuk proses

persetujuan perubahan gaji oleh Bendahara serta untuk pemberitahuan status gaji kepada guru, karena hal ini akan meningkatkan kelancaran komunikasi dan transparansi dalam sistem.

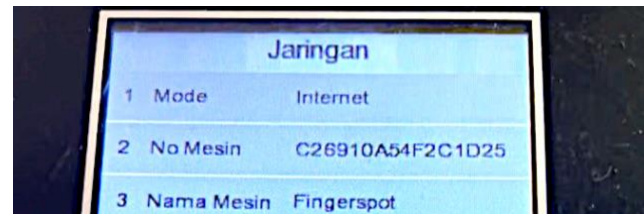
LAMPIRAN

Pada kondisi ideal, sistem seharusnya dapat mengambil data secara langsung dari *developer.fingerspot.io*. Namun, terdapat kendala di lapangan dimana mesin *fingerprint* memiliki *serial number* yang berbeda dengan yang terdaftar di server. Hal ini menyebabkan ketidakcocokan antara data yang ada di mesin dengan yang tersimpan di server *developer.fingerspot.io* (*fingerspot*).



Gambar 5 Mesin *Fingerprint* pada server *developer.fingerspot.io*

Pada Gambar 5 merupakan mesin *fingerprint* yang terdaftar pada server *developer.fingerspot.io*. Sedangkan Gambar 6 merupakan mesin *fingerprint* yang telah didaftarkan pada *developer.fingerspot.io* dengan menggunakan *Serial Number*. Setelah pendaftaran *Cloud ID* adalah nama lain atau nama lain Nomor Mesin pada mesin *fingerprint* Namun dapat dilihat dari Gambar 5 dan Gambar 6 *Cloud ID* dan nomor mesin berbeda, pada akhirnya menyebabkan perintah tertunda pada server *developer.fingerspot.io* seperti Gambar 7. Server *developer.fingerspot.io* tidak dapat melanjutkan mengirimkan perintah dari pengguna ke mesin *fingerprint* karena memiliki *Cloud ID* atau Nomor Mesin yang berbeda dengan keadaan Aktual.



Gambar 6 Mesin *Fingerprint* pada Lokasi pengetesan (TK Al Ihsan Surabaya)

Perintah Tertunda

CLOUD ID	TRANS ID	PERINTAH	DIBUAT
C269248053253836	1	GET_USER_INFO	2025-04-22 12:30:26
C269248053253836	1	GET_USER_INFO	2025-04-22 14:45:35
C269248053253836	1	GET_USER_INFO	2025-04-22 14:45:36
C269248053253836	1	GET_USER_INFO	2025-04-22 14:50:14
C269248053253836	1	GET_USER_INFO	2025-04-22 14:59:57
C269248053253836	1	RESET_FK	2025-04-22 15:19:17
C269248053253836	1	RESET_FK	2025-04-22 15:22:43

Gambar 7 Tampilan Perintah Tertunda di Server *developer.fingerspot.io*

UCAPAN TERIMA KASIH

Puji syukur penulis ucapkan kepada Tuhan Yang Maha Esa atas pimpinan, penyertaan, dan karunia-Nya sehingga penulis dapat menyelesaikan penelitian ini. Penulis juga mengucapkan terima kasih kepada orang tua, keluarga penulis, rekan - rekan penulis juga kepada Ibu Adhatus Solichah Ahmadiyah dan Ibu Kelly Rossa Sungkono selaku dosen pembimbing penulis dan kepada semua pihak yang telah memberikan dukungan baik secara langsung maupun tidak langsung selama penulis melakukan penelitian ini.

- [14] D. Widhiantoro, Suryadewi, and E. Arindhita, "Sistem informasi pengiriman nilai dengan SMS," *Jurnal Ilmiah Elite Elektro*, vol. 4, no. 1, 2013.

DAFTAR PUSTAKA

- [1] D. F. Akbar, "Pengembangan API aplikasi KinderFin: Sistem monitoring keuangan kelompok belajar dan taman kanak-kanak berbasis website," Laporan kerja praktik, Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember, 2024.
- [2] T. Berners-Lee, M. Fischetti, and D. Ramage, "Architecture of the World Wide Web, Volume One," W3C Recommendation, 2001. [Online]. Available: <https://www.w3.org/TR/webarch/>
- [3] Y. Chairunnisa, D. S. Rusdianto, and E. M. A. Jonemaro, "Pengembangan sistem penggajian karyawan berbasis web di CV. Mufidah Terminal Print," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, pp. 3811–3820, 2023.
- [4] Express.js, "Express - Node.js web application framework," 2024. [Online]. Available: <https://expressjs.com/>. [Diakses 10 November 2024].
- [5] H. Y. Hermansyah and Maryam, "Implementasi teknologi application programming interface pada perancangan aplikasi absensi pegawai," *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, vol. 8, no. 3, pp. 744–754, 2023. [Online]. Available: <https://doi.org/10.29100/jipi.v8i3.3890>
- [6] KinderFin, "Pencatatan Kekayaan Intelektual: EC002024237589 / 000810021," Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan HAM, 28 November 2024.
- [7] A. Kurniawan, "Penyusunan rencana strategis sistem informasi dan teknologi informasi di rumah sakit jiwa grhasia daerah istimewa yogyakarta," *Journal of Information Systems for Public Health*, vol. 6, no. 3, p. 43, 2021. [Online]. Available: <https://doi.org/10.22146/jisph.46182>
- [8] Vercel Inc. (2025). Next.js [Online]. Available: <https://nextjs.org>
- [9] H. S. Puspa and F. M. Putra, "Pengembangan aplikasi berbasis web untuk pengelolaan keuangan kelompok belajar dan taman kanak-kanak (KinderFin)," Laporan kerja praktik, Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember, 2024.
- [10] M. Raza, S. Aslam, and M. Khushi, "Web development frameworks: A comparison of Node.js and Django," *International Journal of Modern Education and Computer Science*, vol. 11, no. 2, pp. 1–11, 2019. [Online]. Available: <https://doi.org/10.5815/ijmecs.2019.02.01>
- [11] Resty, Annisa, Rachmat, Agung Ananda, & Wahyu Eko Sulistiono, "Implementasi Golang Clean Architecture pada Perancangan Backend Point of Sales Website," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, 2024. [Online]. Available: <http://dx.doi.org/10.23960/jitet.v12i2.4668>
- [12] S. Sauda and M. Barokah, "Penerapan NodeJS dan PostgreSQL sebagai backend pada aplikasi ecommerce Localla," *INFOTECH Journal*, vol. 8, no. 2, pp. 101–112, 2022. [Online]. Available: <https://doi.org/10.31949/infotech.v8i2.2944>(<https://www.google.com/search?q=https://doi.org/10.31949/infotech.v8i2.2944>)
- [13] E. Supriyadi, M. Sofiana, R. Agoestiyowati, F. Aryani, and Juardi, "Efektifitas Implementasi Teknologi Fingerprint Terhadap Otomatisasi Absensi Pengajar Di Sekolah Menengah Atas Dengan Menggunakan Analisis Swot," *JISAMAR: Journal of Information System, Applied, Management, Accounting and Research*, vol. 7, no. 3, pp. 503–509, 2023.