

Sistem Penggajian Terintegrasi Berbasis API pada SMART UMRI

Heru Prambadi¹, Agus Urip Ari Wibowo², Emansa Hasri Putra³

¹Magister Terapan Teknik Komputer, Teknologi Informasi, Politeknik Caltex Riau

²Magister Terapan Teknik Komputer, Teknologi Informasi, Politeknik Caltex Riau

³Teknologi Rekayasa Jaringan Telekomunikasi, Teknologi Industri, Politeknik Caltex Riau

¹heru22mttk@mahasiswa.pcr.ac.id, ²agus@pcr.ac.id*, ³emansa@pcr.ac.id*

Abstract

Digital transformation in higher education requires integrated information systems that support operational efficiency, data consistency, and accountable governance. Payroll management is a critical administrative process because it involves employee master data, compensation components, taxation, and social security contributions. At Universitas Muhammadiyah Riau, payroll processing previously relied on separated data flows that increased the risk of duplicate entry, inconsistent employee data, delayed payroll generation, and calculation errors. This study aims to examine whether an API-based integrated payroll system connected to SMART UMRI can resolve these problems, using employee and payroll data sourced from the SMART UMRI database. The system was developed using Rapid Application Development and implemented as a RESTful API architecture with HMAC authentication, HTTPS, IP allowlist, role-based access control, and audit trail. Evaluation covered functional testing, API integration, payroll validation, security assessment, and performance testing with Apache JMeter. The results show successful synchronization of employee master data, payroll calculation accuracy with zero difference against manual validation, and stable API performance with 0% error rate in operational testing, confirming that the problems identified before the integration have been resolved. The system improves payroll transparency through digital payslips and traceable logs. The model is feasible for deployment in higher education institutions.

Keywords: payroll system, API integration, SMART UMRI, HMAC authentication, audit trail

Abstrak

Transformasi digital pada perguruan tinggi menuntut sistem informasi yang terintegrasi untuk mendukung efisiensi operasional, konsistensi data, dan tata kelola yang akuntabel. Pengelolaan penggajian merupakan proses administrasi yang kompleks karena melibatkan data pegawai, komponen penghasilan, perpajakan, BPJS, dan kebijakan institusi. Di Universitas Muhammadiyah Riau, proses penggajian sebelumnya belum sepenuhnya terintegrasi dengan sistem utama sehingga menimbulkan potensi duplikasi data, keterlambatan sinkronisasi, dan risiko kesalahan perhitungan *payroll*. Penelitian ini bertujuan menguji apakah implementasi Sistem Penggajian Terintegrasi berbasis *Application Programming Interface* (API) yang terhubung dengan SMART UMRI mampu mengatasi permasalahan tersebut. Data penelitian berupa data pegawai, komponen penghasilan dan potongan, serta parameter PPh Pasal 21 yang bersumber dari database SMART UMRI. Sistem dikembangkan menggunakan metode *Rapid Application Development* dan menerapkan arsitektur *RESTful* API dengan autentikasi HMAC, komunikasi HTTPS, *IP allowlist*, kontrol akses berbasis peran, serta *audit trail*. Evaluasi dilakukan melalui pengujian fungsional, pengujian integrasi API, validasi perhitungan *payroll*, pengujian keamanan, dan pengujian performa menggunakan *Apache JMeter*. Hasil penelitian menunjukkan bahwa sinkronisasi data pegawai berjalan otomatis, validasi *payroll* menghasilkan selisih nol terhadap perhitungan manual, dan pengujian performa menunjukkan *error rate* 0% pada skenario operasional normal. Hasil tersebut membuktikan bahwa permasalahan duplikasi data, keterlambatan sinkronisasi, dan risiko kesalahan perhitungan *payroll* telah teratasi. Sistem meningkatkan transparansi melalui slip gaji digital dan log aktivitas yang dapat ditelusuri. Sistem dinilai layak diterapkan pada lingkungan operasional Universitas Muhammadiyah Riau.

Kata kunci: sistem penggajian, integrasi API, SMART UMRI, HMAC, audit trail

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

1. Pendahuluan

Transformasi digital telah menjadi kebutuhan strategis bagi perguruan tinggi untuk meningkatkan kualitas layanan, efisiensi proses, dan akuntabilitas tata kelola. Integrasi sistem informasi menjadi pendekatan penting karena banyak proses bisnis perguruan tinggi melibatkan data lintas unit, baik pada ranah akademik maupun non-akademik [1]. Pada layanan administrasi, pengelolaan sumber daya manusia dan penggajian membutuhkan data yang selalu mutakhir karena perubahan status pegawai, jabatan, unit kerja, tanggungan keluarga, dan kepesertaan jaminan sosial dapat mempengaruhi hasil perhitungan penghasilan [2].

Penelitian terdahulu juga menunjukkan bahwa pengelolaan gaji yang masih dilakukan secara konvensional memperlambat pekerjaan dan rentan kesalahan perhitungan, sehingga digitalisasi proses penggajian menjadi kebutuhan institusi [3,4].

Penggajian merupakan proses yang sensitif karena hasilnya berhubungan langsung dengan hak pegawai, kepatuhan perpajakan, dan pelaporan keuangan. Kesalahan data pada status pegawai, status pajak, jabatan fungsional, jabatan struktural, atau komponen potongan dapat menimbulkan perbedaan nilai gaji bersih [2,3]. Selain itu, proses penggajian juga melibatkan komponen yang kompleks seperti gaji pokok, tunjangan transport, tunjangan fungsional,

tunjangan keluarga, tunjangan khusus, BPJS Kesehatan, BPJS Ketenagakerjaan, PPh Pasal 21, koperasi, zakat, pinjaman, dan potongan lainnya.

Universitas Muhammadiyah Riau telah memiliki sistem utama bernama SMART UMRI sebagai pusat pengelolaan data institusi. Akan tetapi, proses *payroll* sebelumnya belum sepenuhnya mengalir secara otomatis dari sistem utama menuju sistem penggajian. Kondisi ini berpotensi menimbulkan duplikasi entri, ketidaksesuaian data pegawai, keterlambatan proses *payroll*, dan kesulitan audit perubahan data. Masalah tersebut menunjukkan bahwa sistem *payroll* perlu dihubungkan dengan SMART UMRI melalui mekanisme integrasi yang aman dan terdokumentasi.

Integrasi berbasis API menawarkan solusi karena memungkinkan sistem payroll mengambil data master dari sistem utama secara terjadwal maupun manual. Dalam konteks penelitian ini, API digunakan untuk menarik data pegawai, status pajak, jabatan, dan parameter referensi dari SMART UMRI, kemudian mengolahnya menjadi data penggajian. Penggunaan API juga memungkinkan hasil *payroll*, rekapitulasi, dan slip gaji dapat dikirim atau ditampilkan kembali pada ekosistem SMART UMRI. Dengan demikian, sistem *payroll* tidak lagi berdiri sebagai aplikasi terpisah, tetapi menjadi bagian dari alur informasi yang terintegrasi.

Permasalahan integrasi tidak hanya berkaitan dengan pertukaran data, tetapi juga keamanan dan akuntabilitas. data penggajian bersifat rahasia dan harus dilindungi dari akses tidak sah. Oleh karena itu, sistem yang dievaluasi pada penelitian ini menerapkan *HTTPS*, *API key*, *signature* berbasis *Hash-based Message Authentication Code (HMAC)*, *IP allowlist*, *role-based access control*, dan *audit trail*. Mekanisme tersebut dirancang agar hanya sistem yang sah yang dapat melakukan pertukaran data dan setiap aktivitas penting dapat ditelusuri kembali.

Penelitian ini difokuskan pada evaluasi implementasi Sistem Penggajian Terintegrasi berbasis API pada SMART UMRI. Fokus artikel dibatasi pada integrasi API, sinkronisasi data master, keamanan HMAC dan *IP allowlist*, validasi *payroll*, akurasi perhitungan PPh Pasal 21 menggunakan Tarif Efektif Rata-rata (TER), pengujian performa menggunakan *Apache JMeter*, serta audit trail dan transparansi akses slip gaji. Dengan fokus tersebut, artikel ini tidak menekankan seluruh tampilan antarmuka aplikasi, melainkan mengevaluasi aspek teknis dan operasional yang menentukan kelayakan sistem. Tujuan penelitian ini adalah menguji apakah integrasi berbasis API mampu mengatasi permasalahan duplikasi entri, ketidaksesuaian data pegawai, keterlambatan proses *payroll*, dan kesulitan audit yang terjadi sebelum integrasi.

2. Metode Penelitian

Penelitian menggunakan pendekatan rekayasa sistem dengan metode *Rapid Application Development (RAD)*. Metode ini dipilih karena pengembangan sistem *payroll* membutuhkan proses iteratif, keterlibatan pengguna, dan validasi cepat terhadap kebutuhan bagian kepegawaian serta keuangan. Tahap penelitian meliputi analisis kebutuhan, perancangan arsitektur integrasi, implementasi modul *payroll*, pengujian sistem, dan analisis hasil.

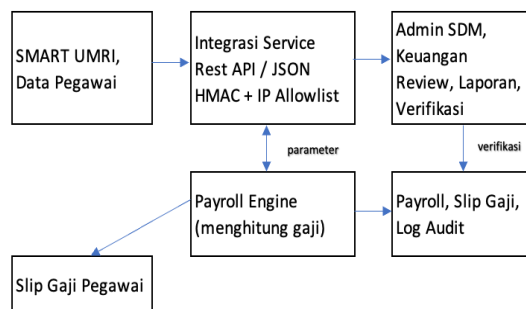
Analisis kebutuhan dilakukan melalui observasi proses *payroll* dan wawancara dengan pengguna yang terlibat dalam pengelolaan data pegawai dan penggajian. Observasi digunakan untuk memahami alur data, komponen gaji, proses verifikasi, serta kendala manual yang muncul sebelum integrasi. Wawancara digunakan untuk menggali kebutuhan pengguna seperti sinkronisasi pegawai, pengaturan komponen gaji, *generate payroll*, pelaporan, audit trail, dan akses slip gaji secara mandiri.

Data yang digunakan dalam pengujian berasal dari database SMART UMRI dan data *payroll* yang merepresentasikan proses operasional universitas. Komponen data meliputi identitas pegawai, status pegawai, status pajak, jabatan, unit kerja, rekening, NPWP, BPJS, gaji pokok, tunjangan, potongan, honor, serta parameter PPh Pasal 21. Dalam sistem, SMART UMRI diposisikan sebagai sumber data utama sehingga data master pegawai tidak perlu diinput ulang ke aplikasi *payroll*.

2.1. Arsitektur Integrasi

Arsitektur integrasi dirancang berbasis *client-server* dengan REST API sebagai mekanisme komunikasi antara SMART UMRI dan sistem *payroll*. SMART UMRI bertindak sebagai penyedia data master, sedangkan sistem payroll bertindak sebagai consumer yang melakukan permintaan data. Data dikirim menggunakan format JSON agar struktur data lebih mudah divalidasi dan dikembangkan. Integrasi dapat dijalankan otomatis melalui *scheduler* setiap hari pada pukul 02.00 dan dapat dijalankan manual jika terdapat perubahan data yang harus segera disinkronkan.

Lapisan aplikasi terdiri dari *integration service*, *payroll engine*, *reporting and audit module*, dan *security service*. *Integration service* menangani komunikasi API dan validasi data. *Payroll engine* menghitung penghasilan, potongan, BPJS, dan PPh Pasal 21. *Reporting and audit module* menghasilkan slip gaji, rekap, dan log aktivitas. *Security service* menangani autentikasi, otorisasi, validasi HMAC, dan pembatasan akses berdasarkan IP. Arsitektur integrasi antara SMART UMRI dan sistem *payroll* ditunjukkan pada Gambar 1.



Gambar 1. Arsitektur integrasi SMART UMRI dan sistem payroll

2.2. Desain API dan Keamanan

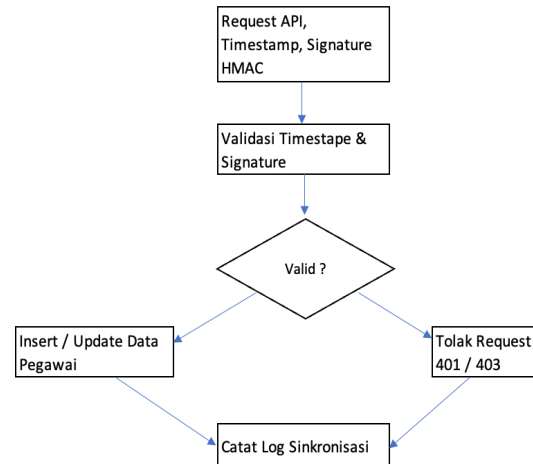
Desain API menerapkan pola *pull-based synchronization*, yaitu sistem payroll secara aktif meminta data ke SMART UMRI melalui *endpoint* tertentu. Pendekatan ini dipilih agar proses pengambilan data dapat dikendalikan oleh sistem payroll sesuai jadwal operasional dan kebutuhan validasi. Setiap permintaan API membawa API key, timestamp, dan signature HMAC. Signature dihasilkan dari kombinasi *payload*, *timestamp*, dan *secret key* yang hanya diketahui oleh kedua sistem. Jika *signature* tidak valid atau *timestamp* berada di luar batas toleransi, request ditolak.

Selain HMAC, sistem juga menerapkan IP allowlist pada sisi SMART UMRI. Dengan cara ini, hanya server payroll yang alamat IP-nya terdaftar yang dapat mengakses *endpoint*. Kombinasi HTTPS, API key, HMAC, *timestamp*, dan IP *allowlist* digunakan untuk memenuhi prinsip kerahasiaan, integritas, dan pembatasan penggunaan data. Nilai *secret key* tidak ditampilkan dalam naskah ini karena termasuk informasi sensitif yang tidak layak dipublikasikan. Struktur *endpoint* sinkronisasi pegawai dirangkum pada Tabel 1, sedangkan alur validasi request oleh sistem ditunjukkan pada Gambar 2.

Tabel 1. Struktur API sinkronisasi pegawai

Komponen	Keterangan
Endpoint	/api/v1/pegawai
Metode	GET
Header	X-API-KEY, X-SIGNATURE, X-TIMESTAMP
Format respons	JSON data pegawai
Keamanan	HTTPS, HMAC, IP allowlist
Operasi data	Insert / update berdasarkan ID pegawai

Tabel 1 memperlihatkan bahwa seluruh pertukaran data dilakukan melalui satu *endpoint* dengan *header* keamanan yang wajib disertakan pada setiap *request*. Setiap *request* yang masuk kemudian melewati tahapan pemeriksaan sebagaimana dijelaskan pada Gambar 2.



Gambar 2. Alur validasi request API dengan HMAC

2.3. Metode Pengujian

Pengujian dilakukan untuk menilai pemenuhan kebutuhan fungsional, keberhasilan integrasi API, akurasi perhitungan payroll, keamanan akses, dan performa sistem. Pengujian fungsional dilakukan dengan metode *black-box* [5] terhadap fitur sinkronisasi pegawai, perhitungan komponen gaji, generate payroll, cetak slip gaji, perhitungan pajak TER, dan API payroll. Pengujian integrasi menilai apakah *request* valid diterima, request tidak sah ditolak, dan operasi *insert* serta *update* dapat dilakukan tanpa menimbulkan data ganda.

Validasi perhitungan dilakukan dengan membandingkan hasil sistem terhadap perhitungan manual pada dua sampel pegawai yang mewakili variasi status dan komponen gaji. Validasi mencakup gaji pokok, tunjangan, BPJS, PPh Pasal 21, potongan, dan total penerimaan. Pengujian performa dilakukan menggunakan Apache JMeter 5.6.3 pada *endpoint* sinkronisasi pegawai, API payroll, dan HTTP *request generate payroll*. Indikator yang diamati meliputi *average response time*, *minimum response*, *maximum response*, *standard deviation*, dan *error rate*, yang merepresentasikan karakteristik efisiensi kinerja perangkat lunak [6]. Parameter eksperimen yang digunakan dirangkum pada Tabel 2.

Tabel 2. Parameter eksperimen sistem

Parameter	Nilai
Sampel validasi manual	2 pegawai
Jenis pegawai	Dosen dan tenaga kependidikan
Metode pajak	PPh 21 TER
Basis perhitungan	Penghasilan neto
Alat uji performa	Apache JMeter 5.6.3
Virtual users	1-10 threads
Metode uji	Fungsional, integrasi, validasi, performa

3. Hasil dan Pembahasan

Hasil evaluasi menunjukkan bahwa sistem payroll terintegrasi mampu menjalankan fungsi utama sesuai

kebutuhan operasional. Pembahasan disusun berdasarkan empat aspek: keberhasilan fungsional dan sinkronisasi data, akurasi *payroll*, performa sistem, serta keamanan dan auditabilitas. Pendekatan ini digunakan agar hasil penelitian dapat menjawab kebutuhan implementasi sekaligus menunjukkan kontribusi teknis dari integrasi API. Setiap aspek pengujian diarahkan untuk membuktikan apakah permasalahan yang diidentifikasi pada pendahuluan telah teratasi, yaitu duplikasi entri diuji melalui mekanisme sinkronisasi, ketidaksesuaian data diuji melalui validasi *payroll*, keterlambatan proses diuji melalui pengujian performa, dan kesulitan audit diuji melalui pemeriksaan *audit trail*.

3.1. Hasil Pengujian Fungsional dan Sinkronisasi

Pengujian fungsional menunjukkan bahwa fitur sinkronisasi pegawai, pengelolaan komponen gaji, *generate payroll*, pencetakan slip gaji, perhitungan pajak TER, serta API *payroll* berhasil dijalankan. Hasil ini menunjukkan bahwa sistem mampu memenuhi kebutuhan dasar penggajian yang sebelumnya membutuhkan penyesuaian data secara terpisah. Ringkasan hasil pengujian fungsional disajikan pada Tabel 3.

Sinkronisasi data master dilakukan berdasarkan ID pegawai sebagai identitas tunggal. Ketika data dari SMART UMRI diterima, sistem memeriksa apakah data tersebut telah tersedia pada database payroll. Jika data belum ada, sistem melakukan insert; jika data telah ada, sistem melakukan update. Mekanisme ini mencegah duplikasi data dan menjaga konsistensi antara sistem utama dan sistem *payroll*. Untuk data yang mempengaruhi perhitungan pajak, seperti status nikah dan tanggungan, sistem menggunakan effective date agar perubahan tidak mengganggu periode penggajian berjalan.

Temuan ini memperlihatkan bahwa prinsip single source of truth dapat diterapkan pada proses payroll di lingkungan perguruan tinggi. SMART UMRI menjadi sumber data utama, sedangkan sistem payroll menjadi pengolah transaksi penggajian. Dengan pembagian peran tersebut, perubahan data pegawai tidak perlu dilakukan pada banyak aplikasi sehingga risiko perbedaan data dapat ditekan.

Tabel 3. Ringkasan hasil pengujian fungsional

Fungsi	Hasil	Keterangan
Sinkronisasi pegawai	Berhasil	Data diterima dan diperbarui
Komponen gaji	Berhasil	Gaji, tunjangan, potongan tersimpan
Generate payroll	Berhasil	Payroll terbentuk sesuai periode
Cetak slip gaji	Berhasil	Slip digital tersedia
PPh 21 TER	Berhasil	Kategori TER terbaca
API payroll	Berhasil	GET/POST berjalan

3.2. Validasi Akurasi Payroll

Validasi manual dilakukan pada dua pegawai sampel. Pegawai X memiliki total bruto Rp5.339.300 dan PPh 21 bulanan 0% karena termasuk kategori TER B pada rentang penghasilan yang sesuai. Pegawai Y memiliki total bruto Rp7.553.600 dan PPh 21 sebesar Rp113.304 berdasarkan tarif efektif kategori A sebesar 1,5%. Pada kedua sampel, seluruh komponen hasil perhitungan sistem sama dengan perhitungan manual.

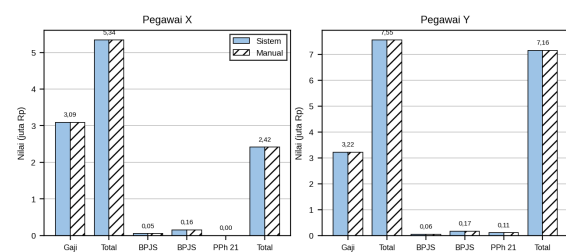
Komponen yang divalidasi meliputi gaji pokok, tunjangan jabatan, tunjangan fungsional, tunjangan keluarga, tunjangan transport, tunjangan khusus, tunjangan natura, BAD, BPJS Kesehatan, BPJS Ketenagakerjaan, PPh 21, koperasi, zakat, majalah, potongan lainnya, dan total penerimaan. Seluruh komponen menghasilkan selisih nol. Dengan demikian, pada data sampel yang diuji, akurasi sistem mencapai 100%. Rincian hasil validasi disajikan pada Tabel 4 berikut.

Hasil ini penting karena perhitungan *payroll* tidak hanya membutuhkan formula aritmetika, tetapi juga logika bisnis yang mengikuti status pegawai, status pajak, dan parameter institusi. Kesamaan hasil sistem dan perhitungan manual menunjukkan bahwa implementasi formula payroll telah sesuai dengan kebutuhan operasional UMRI. Selain itu, perhitungan TER telah mampu mengelompokkan status PTKP ke kategori yang tepat.

Tabel 4. Ringkasan validasi payroll manual dan sistem

Komponen	Pegawai X	Pegawai Y	Selisih
Gaji pokok	3.089.300	3.220.000	0
Total bruto	5.339.300	7.553.600	0
BPJS	53.393	56.636	0
Kesehatan			
BPJS Tenaga Kerja	160.179	169.908	0
PPh 21	0	113.304	0
Total penerimaan	2.417.022	7.155.752	0

Tabel 4 memperlihatkan bahwa seluruh komponen perhitungan pada kedua pegawai sampel menghasilkan selisih nol. Gambar 3 kemudian menyandingkan nilai-nilai tersebut dalam bentuk grafik sehingga kesamaan hasil perhitungan sistem dan manual dapat diamati langsung pada setiap komponen.



Gambar 3. Perbandingan hasil perhitungan sistem dan validasi manual

3.3. Pengujian Performa Menggunakan JMeter

Pengujian performa dilakukan menggunakan Apache JMeter terhadap tiga endpoint utama, yaitu API sinkronisasi pegawai, API *payroll*, dan HTTP *request*

generate payroll. Pengujian dilakukan tiga kali pada kondisi koneksi yang sama untuk melihat konsistensi respons. Total rata-rata eksekusi pada pengujian pertama adalah 4,139 detik, pengujian kedua 4,031 detik, dan pengujian ketiga 4,044 detik. Ketiga pengujian menghasilkan *error rate* 0%. Ringkasan hasil pengujian performa disajikan pada Tabel 5.

Endpoint API sinkronisasi pegawai memiliki waktu respons paling rendah karena hanya melakukan pengambilan dan pembaruan data master. Pada pengujian ketiga, rata-rata waktu respons endpoint ini adalah 815 ms dengan nilai maksimum 1.197 ms. Nilai tersebut menunjukkan bahwa proses sinkronisasi berada pada kategori responsif untuk kebutuhan integrasi terjadwal. Karena sinkronisasi dijalankan pada waktu tertentu, waktu respons di bawah satu hingga sekitar satu detik masih sangat memadai untuk operasional.

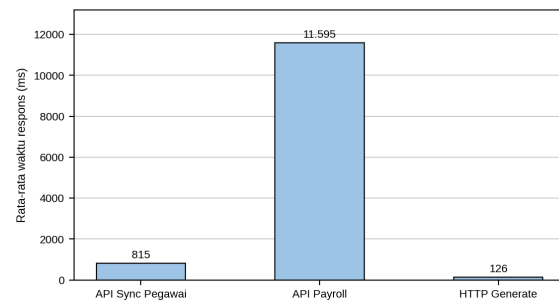
Endpoint API payroll memiliki waktu respons paling tinggi karena melakukan proses komputasi yang lebih kompleks, termasuk *assembling* komponen gaji, perhitungan PPh 21 TER, penyusunan histori, dan penyimpanan data payroll. Rata-rata waktu respons API payroll pada tiga pengujian berada pada kisaran 11,420 sampai 11,625 detik. Meskipun lebih lambat dibanding *endpoint* lain, *error rate* tetap 0% dan variasi waktu masih dapat diterima karena proses payroll dilakukan secara periodik, bukan sebagai transaksi *real-time* untuk pengguna akhir.

HTTP request generate payroll menunjukkan waktu respons sangat rendah, yaitu sekitar 120 sampai 126 ms. Hal ini mengindikasikan bahwa pemanggilan proses dari sisi antarmuka tidak menjadi beban utama. Bottleneck utama berada pada proses komputasi payroll, bukan pada request antarmuka. Hasil ini memberikan arah optimasi yang jelas, yaitu penggunaan *queue worker*, *batch processing*, *optimasi query*, atau pemisahan proses berat ke *background job*. Perbandingan rata-rata waktu respons ketiga endpoint divisualisasikan pada Gambar 4.

Tabel 5. Ringkasan hasil pengujian JMeter

Endpoint	Avg (ms)	Min (ms)	Max (ms)	Error
API Sync Pegawai	815	679	1.197	0%
API Payroll	11.595	11.266	12.924	0%
HTTP Generate	126	100	167	0%
Total	4.044	100	12.924	0%

Tabel 5 menunjukkan bahwa seluruh *endpoint* berjalan dengan *error rate* 0% meskipun besaran waktu responsnya berbeda sesuai beban komputasi masing-masing. Perbedaan tersebut tampak jelas pada Gambar 4, di mana *endpoint API payroll* memiliki waktu respons tertinggi karena menjalankan proses komputasi paling kompleks.



Gambar 4. Grafik rata-rata waktu respons endpoint

3.4. Keamanan, Audit Trail, dan Transparansi

Keamanan integrasi menjadi aspek penting karena data payroll merupakan data sensitif. Sistem menggunakan beberapa lapisan pengamanan, yaitu HTTPS untuk melindungi komunikasi, API *key* untuk mengidentifikasi *client*, HMAC untuk memverifikasi *integritas request*, *timestamp* untuk mengurangi risiko replay request, dan IP allowlist untuk membatasi sumber permintaan. Jika request tidak memiliki signature valid atau berasal dari IP yang tidak terdaftar, sistem menolak permintaan tersebut. Rangkuman kontrol keamanan dan tata kelola sistem disajikan pada Tabel 6.

Selain keamanan integrasi, sistem menerapkan *role-based access control* untuk membatasi hak akses pengguna. Admin kepegawaian dapat mengelola data pegawai, admin keuangan dapat memproses payroll dan laporan, pimpinan dapat melakukan verifikasi sesuai kewenangan, administrator mengatur hak akses, dan pegawai hanya dapat melihat slip gaji miliknya sendiri. Pembatasan hak akses ini penting agar kerahasiaan informasi penggajian tetap terjaga.

Audit trail diterapkan untuk mencatat aktivitas kritis seperti sinkronisasi data, perubahan komponen gaji, perubahan parameter, dan proses *generate payroll*. Log mencakup waktu aktivitas, identitas pengguna, dan jenis aktivitas. Dengan adanya audit trail, setiap perubahan dapat ditelusuri kembali ketika terjadi perbedaan atau kebutuhan pemeriksaan internal. Fitur ini memperkuat akuntabilitas dan memudahkan proses audit.

Transparansi meningkat melalui penyediaan slip gaji digital. Setelah *payroll* selesai diverifikasi dan dipublikasikan, pegawai dapat mengakses slip gaji melalui akun masing-masing. Hal ini mengurangi ketergantungan pada dokumen cetak dan mempercepat layanan administrasi. Transparansi tidak berarti membuka seluruh data *payroll* kepada semua pihak, tetapi memberikan akses yang tepat kepada pengguna yang berwenang.

Tabel 6. Kontrol keamanan dan tata kelola sistem

Kontrol	Tujuan	Dampak
HTTPS	Melindungi transmisi	Kerahasiaan data
HMAC	Validasi integritas request	Mencegah request palsu

IP allowlist	Membatasi sumber akses	Mengurangi akses ilegal
RBAC	Mengatur hak pengguna	Kerahasiaan slip gaji
Audit trail	Mencatat aktivitas	Akuntabilitas

3.5. Pembahasan

Hasil penelitian menunjukkan bahwa integrasi API memberikan manfaat nyata terhadap proses payroll di lingkungan perguruan tinggi. Pertama, integrasi mengurangi duplikasi data karena data master pegawai bersumber dari SMART UMRI. Kedua, proses sinkronisasi otomatis mempercepat persiapan data payroll. Ketiga, validasi manual menunjukkan bahwa hasil perhitungan sistem dapat dipercaya untuk menghasilkan slip gaji. Keempat, pengujian performa memperlihatkan bahwa sistem mampu berjalan stabil pada skenario operasional normal. Dengan demikian, keempat permasalahan yang melatarbelakangi penelitian terbukti teratasi, yaitu duplikasi entri hilang karena data master bersumber tunggal, ketidaksesuaian data tereliminasi dengan selisih perhitungan nol, keterlambatan proses berkurang melalui sinkronisasi terjadwal dengan *error rate* 0%, dan kesulitan audit teratasi melalui log aktivitas yang dapat ditelusuri.

Dari sisi arsitektur, pola *pull-based synchronization* memberikan kendali kepada sistem payroll untuk menentukan kapan data perlu ditarik. Pola ini lebih sesuai untuk proses *payroll* karena integrasi tidak harus berlangsung *real-time* setiap detik, tetapi harus konsisten sebelum periode penggajian dijalankan. Scheduler harian pada pukul 02.00 menjadi pilihan yang relevan karena mengurangi beban pada jam kerja dan memastikan data sudah tersedia ketika bagian keuangan melakukan proses payroll.

Dari sisi akurasi, selisih nol pada dua sampel menunjukkan bahwa sistem telah menerapkan formula payroll sesuai parameter yang diuji. Walaupun jumlah sampel validasi manual terbatas, sampel tersebut telah mencakup variasi komponen gaji dan kategori TER yang berbeda. Untuk penerapan lebih luas, validasi dapat diperluas pada seluruh pegawai atau minimal menggunakan sampel stratifikasi berdasarkan status pegawai, status pajak, jabatan, dan kepesertaan BPJS.

Dari sisi performa, *endpoint API payroll* merupakan bagian yang paling berat. Hal ini wajar karena proses payroll melakukan agregasi data dan perhitungan banyak komponen. Karena *payroll* merupakan proses periodik, waktu sekitar 11 detik pada endpoint komputasi masih dapat diterima untuk kebutuhan operasional normal. Akan tetapi, jika jumlah pegawai bertambah atau proses dilakukan secara simultan, optimasi *query*, *job batching*, dan *queue worker* perlu diprioritaskan.

Dari sisi tata kelola, penerapan HMAC, IP *allowlist*, RBAC, dan *audit trail* menunjukkan bahwa sistem tidak hanya menyelesaikan masalah teknis penghitungan gaji, tetapi juga mendukung prinsip keamanan dan akuntabilitas. Hal ini menjadi kontribusi

penting karena banyak pengembangan sistem payroll hanya menekankan fungsi perhitungan, sementara aspek audit dan pengendalian akses sering dibahas secara terbatas.

3.6. Posisi Penelitian terhadap Penelitian Terkait

Hasil penelitian ini memperluas temuan penelitian terdahulu yang menyatakan bahwa sistem penggajian berbasis web dapat meningkatkan efisiensi proses dan aksesibilitas data [3,4,7,8]. Perbedaan utama penelitian ini terletak pada fokus integrasi langsung dengan sistem utama universitas melalui API, bukan hanya pengembangan aplikasi payroll mandiri. Pada sistem yang dievaluasi, data pegawai tidak lagi diperlakukan sebagai data salinan yang harus selalu diperbarui secara manual, tetapi sebagai data master yang berasal dari SMART UMRI. Dengan demikian, manfaat integrasi tidak hanya terlihat pada kecepatan proses, tetapi juga pada konsistensi sumber data dan kemampuan audit.

Studi mengenai integrasi sistem payroll dengan sistem keuangan maupun sistem sumber daya manusia umumnya menekankan peningkatan koordinasi antarunit [9-11]. Penelitian ini menunjukkan bahwa koordinasi tersebut dapat diwujudkan melalui mekanisme teknis yang lebih terukur, yaitu *endpoint API*, *scheduler*, identitas pegawai unik, dan log sinkronisasi. Setiap perubahan data master dapat dilacak melalui proses sinkronisasi, sedangkan proses generate payroll dapat diverifikasi oleh admin keuangan sebelum slip gaji dipublikasikan kepada pegawai. Mekanisme ini menjembatani kebutuhan teknis dan kebutuhan tata kelola.

Kontribusi lain penelitian ini adalah penggabungan aspek keamanan dan performa dalam evaluasi sistem payroll. Banyak implementasi sistem administratif hanya menampilkan fitur antarmuka dan keluaran laporan, tetapi tidak menilai bagaimana pertukaran data diamankan atau bagaimana sistem merespons ketika *endpoint* diuji. Pada penelitian ini, HMAC, *timestamp*, IP *allowlist*, dan RBAC digunakan sebagai lapisan kontrol, sedangkan *Apache JMeter* digunakan untuk mengukur perilaku endpoint. Kombinasi ini menjadikan evaluasi sistem lebih lengkap karena tidak hanya menjawab apakah sistem dapat berjalan, tetapi juga apakah sistem cukup aman dan stabil untuk digunakan.

3.7. Formula Payroll dan Kesesuaian Proses Bisnis

Proses *payroll* yang dievaluasi tidak hanya menghitung gaji pokok, tetapi juga melakukan agregasi banyak komponen penghasilan dan potongan. Total penghasilan diperoleh dari penjumlahan gaji pokok, tunjangan jabatan, tunjangan fungsional, tunjangan keluarga, tunjangan transport, tunjangan kinerja, tunjangan khusus, tunjangan natura, BAD, dan honor lainnya. Total potongan mencakup BPJS Kesehatan, BPJS Ketenagakerjaan, PPh 21, koperasi, pinjaman, zakat, majalah, wakaf, qurban, denda kehadiran, dan

potongan lainnya. Gaji bersih kemudian diperoleh dari total penghasilan dikurangi total potongan.

Kesesuaian sistem dengan proses bisnis UMRI terlihat dari kemampuan sistem menyimpan parameter yang dapat berubah, seperti status nikah, status pajak, tunjangan, potongan, dan nilai BPJS. Parameter tersebut dikelola oleh pengguna berwenang sehingga perubahan kebijakan tidak selalu membutuhkan perubahan kode program. Pendekatan ini penting karena kebijakan penggajian dapat berubah mengikuti regulasi nasional dan ketentuan internal universitas. Sistem yang terlalu kaku akan sulit dipelihara, sedangkan sistem yang menggunakan parameter dapat lebih mudah dikembangkan.

Perhitungan PPh 21 menggunakan pendekatan Tarif Efektif Rata-rata sesuai ketentuan perpajakan yang berlaku [12,13]. Sistem menentukan kategori TER berdasarkan status perkawinan dan tanggungan, kemudian mengambil persentase potongan sesuai rentang penghasilan. Pada sampel Pegawai X, status dan penghasilan bruto menempatkan pegawai pada kategori dengan tarif 0%, sehingga PPh 21 bulanan bernilai nol. Pada sampel Pegawai Y, penghasilan bruto berada pada rentang kategori A dengan tarif 1,5%, sehingga sistem menghasilkan PPh 21 sebesar Rp113.304. Hasil tersebut sama dengan perhitungan manual sehingga menunjukkan kesesuaian algoritma dengan skenario validasi.

3.8. Implikasi Implementasi terhadap Tata Kelola

Implementasi sistem payroll terintegrasi memiliki implikasi langsung terhadap tata kelola administrasi sumber daya manusia, sejalan dengan temuan penerapan ERP pada pengelolaan gaji di perguruan tinggi [14]. Dari sisi operasional, bagian kepegawaian tidak perlu mengirim ulang data dasar pegawai secara terpisah kepada bagian keuangan apabila data telah tersedia pada SMART UMRI. Dari sisi keuangan, proses generate payroll dapat dilakukan setelah data sinkron dan komponen gaji diverifikasi. Dari sisi pegawai, slip gaji digital dapat diakses secara mandiri setelah dipublikasikan. Alur ini memperpendek rantai administrasi dan mengurangi pekerjaan repetitif.

Dari sisi akuntabilitas, audit trail memberikan kemampuan telusur terhadap perubahan data dan aktivitas sistem. Setiap aktivitas penting yang berkaitan dengan sinkronisasi, perubahan komponen gaji, dan generate payroll dapat dicatat bersama identitas pengguna dan waktu aktivitas. Apabila terjadi perbedaan nominal atau pertanyaan dari pegawai, admin dapat menelusuri kembali proses yang menghasilkan nilai tersebut. Kemampuan telusur ini merupakan elemen penting dalam sistem penggajian karena penggajian berkaitan dengan hak pegawai dan laporan keuangan institusi.

Dari sisi keamanan, pembatasan akses berdasarkan peran memastikan bahwa data payroll tidak dapat diakses oleh semua pengguna. Pegawai hanya melihat

slip gaji miliknya sendiri, sedangkan admin dan pimpinan memperoleh akses sesuai tugasnya. Dalam konteks perguruan tinggi, pembatasan ini penting karena data gaji bersifat rahasia dan dapat menimbulkan risiko privasi apabila tersebar. Implementasi RBAC dan verifikasi publikasi slip gaji membantu menjaga keseimbangan antara transparansi layanan dan kerahasiaan informasi.

3.9. Keterbatasan dan Arah Pengembangan

Meskipun hasil validasi menunjukkan akurasi 100% pada dua sampel, jumlah sampel tersebut masih terbatas. Untuk penerapan jangka panjang, pengujian akurasi sebaiknya diperluas menggunakan dataset yang lebih besar dan mencakup seluruh variasi status pegawai, status pajak, jabatan, kepesertaan BPJS, serta komponen potongan. Pengujian dengan dataset yang lebih besar akan memberikan keyakinan lebih tinggi terhadap keandalan algoritma pada kondisi nyata yang lebih beragam.

Pengujian performa juga menunjukkan bahwa proses API *payroll* merupakan bagian yang paling berat. Walaupun waktu sekitar sebelas detik masih dapat diterima untuk proses periodik bulanan, peningkatan jumlah pegawai atau penambahan komponen payroll dapat meningkatkan beban komputasi. Oleh karena itu, pengembangan selanjutnya perlu mempertimbangkan *queue worker*, *job batching*, *optimasi query database*, indeks pada kolom yang sering digunakan, dan pemrosesan paralel untuk mengurangi waktu *generate payroll*.

Pengembangan berikutnya juga dapat diarahkan pada integrasi dengan sistem keuangan dan *e-budgeting* universitas. Integrasi tersebut akan mempercepat proses rekonsiliasi, pelaporan, dan audit. Selain itu, sistem dapat dilengkapi dashboard analitik untuk memantau tren biaya pegawai, distribusi tunjangan, beban BPJS, serta estimasi kewajiban pajak. Dengan demikian, data *payroll* tidak hanya digunakan untuk pembayaran gaji, tetapi juga untuk mendukung pengambilan keputusan strategis dalam pengelolaan sumber daya manusia.

3.10. Analisis Risiko Operasional

Sistem penggajian terintegrasi mengurangi beberapa risiko operasional yang umum terjadi pada proses payroll manual atau semi-manual. Risiko pertama adalah risiko duplikasi data. Risiko ini muncul ketika data pegawai disalin ke aplikasi yang berbeda dan diperbarui oleh unit yang berbeda. Dengan integrasi API, data pegawai dipusatkan pada SMART UMRI dan sistem payroll hanya menggunakan hasil sinkronisasi. Risiko kedua adalah risiko keterlambatan proses. Scheduler harian membuat data dasar tersedia sebelum periode penggajian sehingga bagian keuangan tidak harus menunggu pengiriman data manual. Risiko ketiga adalah risiko kesalahan perhitungan. Penggunaan formula terpusat dan parameter yang dikendalikan

membantu memastikan perhitungan dilakukan secara konsisten pada setiap periode.

Risiko lain yang perlu diperhatikan adalah kegagalan sinkronisasi atau ketidaksesuaian data karena perubahan struktur API. Untuk mengurangi risiko ini, kontrak API harus terdokumentasi dengan baik, termasuk definisi endpoint, parameter, skema JSON, dan kode respons. Setiap perubahan pada SMART UMRI yang mempengaruhi struktur data harus diuji pada lingkungan pengujian sebelum diterapkan ke produksi. Penggunaan log sinkronisasi juga membantu administrator mengetahui apakah data berhasil diperbarui atau gagal diproses. Dengan demikian, risiko integrasi tidak dihilangkan sepenuhnya, tetapi dapat dikendalikan melalui prosedur teknis yang jelas.

Risiko keamanan juga harus dipandang sebagai bagian dari risiko operasional. Data *payroll* mengandung informasi pribadi dan finansial pegawai. Apabila *endpoint* API terbuka tanpa autentikasi yang kuat, data dapat diakses oleh pihak yang tidak berwenang. Oleh karena itu, penggunaan HTTPS, API key, HMAC, *timestamp*, dan IP *allowlist* menjadi kontrol minimal yang harus dipertahankan. Pada sisi aplikasi, RBAC perlu ditinjau secara berkala agar tidak ada pengguna yang memiliki hak akses lebih luas dari tugasnya. Audit hak akses penting dilakukan terutama ketika terjadi mutasi jabatan atau perubahan personel pada unit keuangan dan kepegawaian.

3.11. Evaluasi Kelayakan Implementasi

Kelayakan implementasi sistem dinilai berdasarkan tiga indikator utama, yaitu kelayakan fungsional, kelayakan akurasi, dan kelayakan teknis. Kelayakan fungsional ditunjukkan oleh keberhasilan seluruh fungsi inti yang diuji. Sistem dapat menyinkronkan data, menghitung *payroll*, menghasilkan slip gaji, dan menyediakan laporan. Kelayakan akurasi ditunjukkan oleh hasil validasi manual yang menghasilkan selisih nol pada sampel pengujian. Kelayakan teknis ditunjukkan oleh hasil pengujian performa dengan *error rate* 0% pada pengujian *endpoint* utama.

Dari sisi pengguna, sistem juga layak digunakan karena alur kerja yang dihasilkan sesuai dengan pembagian tugas antara kepegawaian, keuangan, pimpinan, administrator, dan pegawai. Bagian kepegawaian berperan menyediakan dan memastikan kebenaran data master. Bagian keuangan berperan mengatur komponen gaji, melakukan *generate payroll*, dan memeriksa hasil perhitungan. Pimpinan atau pejabat berwenang dapat melakukan verifikasi. Pegawai memperoleh akses terhadap slip gaji digital setelah proses dipublikasikan. Pembagian peran ini menunjukkan bahwa sistem tidak hanya memenuhi kebutuhan teknis, tetapi juga sesuai dengan struktur kerja organisasi.

Berdasarkan evaluasi tersebut, sistem dapat diterapkan pada lingkungan operasional UMRI dengan catatan adanya prosedur pemeliharaan berkala. Pemeliharaan

mencakup pengecekan log sinkronisasi, pembaruan parameter pajak dan BPJS, pencadangan database, pemantauan performa server, dan audit hak akses. Prosedur tersebut diperlukan agar manfaat integrasi tetap terjaga setelah sistem digunakan dalam jangka panjang. Dengan adanya pemeliharaan yang baik, sistem *payroll* terintegrasi dapat menjadi fondasi pengembangan layanan administrasi SDM yang lebih luas.

4. Kesimpulan

Penelitian ini membuktikan bahwa Sistem Penggajian Terintegrasi berbasis API pada SMART UMRI mampu mengatasi permasalahan duplikasi entri, ketidaksesuaian data pegawai, keterlambatan proses *payroll*, dan kesulitan audit yang melatarbelakangi penelitian. Sistem berhasil menyinkronkan data pegawai secara otomatis, menghitung seluruh komponen *payroll*, dan menghasilkan slip gaji digital sesuai kebutuhan operasional. Validasi terhadap dua pegawai sampel menghasilkan selisih nol pada seluruh komponen yang diuji sehingga akurasi mencapai 100%, sedangkan pengujian JMeter menunjukkan *error rate* 0% dengan waktu respons yang masih layak untuk proses penggajian periodik.

Penerapan HMAC, HTTPS, IP *allowlist*, RBAC, dan *audit trail* meningkatkan keamanan serta akuntabilitas pengelolaan *payroll*, dan slip gaji digital meningkatkan transparansi layanan tanpa mengurangi kerahasiaan data. Secara keseluruhan, sistem layak digunakan pada lingkungan operasional Universitas Muhammadiyah Riau. Pengembangan selanjutnya dapat diarahkan pada integrasi dengan sistem keuangan, perluasan validasi pada seluruh populasi pegawai, serta optimasi performa melalui *queue worker*, *job batching*, dan pemrosesan paralel.

Daftar Rujukan

- [1] Achsan, S. A., and Susetyo, Y. A., 2022. RESTful web service implementation using Spring framework in room assets management system. *Jurnal Teknik Informatika (JUTIF)*, 3(2). doi: <https://doi.org/10.20884/1.jutif.2022.3.2.213>.
- [2] Setiawan, R., Ikhwana, A., and Muhidin, A. M., 2017. Rancang bangun sistem informasi penggajian di perguruan tinggi swasta. *Jurnal Algoritma*, 14(2), pp. 373-383.
- [3] Pratama, F. R., and Kadafi, M., 2022. Sistem informasi pengelolaan gaji karyawan Bank Sumsel Babel Kantor Cabang Pembantu Lemabang. *Jurnal FASILKOM*, 12(1), pp. 1-7. doi: <https://doi.org/10.37859/jf.v12i1.3528>.
- [4] Amir, A. Y., and Devi, P. A. R., 2022. Sistem informasi penggajian karyawan berbasis framework CodeIgniter di Toko Citra Mandiri Gresik. *Jurnal FASILKOM*, 12(1), pp. 35-42. doi: <https://doi.org/10.37859/jf.v12i1.3472>.
- [5] Pressman, R. S., 2010. *Software Engineering: A Practitioner's Approach*. 7th ed. New York: McGraw-Hill.
- [6] ISO/IEC, 2011. ISO/IEC 25010: Systems and software engineering - systems and software quality requirements and evaluation. Geneva: International Organization for Standardization.
- [7] Dinka, S. P., Salsabilah, Z. P., and Nilawati, L., 2022. Penerapan metode waterfall dalam rancang bangun sistem

-
- informasi penggajian berbasis web. Artikel Ilmiah Sistem Informasi Akuntansi (AKASIA), 2(2), pp. 156-166. doi: <https://doi.org/10.31294/akasia.v2i2.1431>.
- [8] Setiadi, M. R., Nugroho, R. A., and Abdussalaam, F., 2022. Perancangan sistem informasi penggajian berbasis web di Kantor Pos Bandung. JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika), 7(3), pp. 639-650. doi: <https://doi.org/10.29100/jipi.v7i3.2883>.
- [9] Zhao, L., and Rabiei, M., 2023. Integration of payroll systems with human resource management systems. International Journal of Computer Applications, 185(12), pp. 10-18.
- [10] Gaffar, A., and Gaffar, H., 2024. Integrasi sistem informasi penggajian dengan sistem keuangan. Jurnal Teknologi dan Sistem Informasi, 11(2), pp. 66-75.
- [11] Judijanto, L., et al., 2025. Evaluasi sistem penggajian terintegrasi dengan sistem HR di institusi pendidikan. Jurnal Informatika Terapan, 10(1), pp. 1-10.
- [12] Kementerian Keuangan Republik Indonesia, 2023. Peraturan Menteri Keuangan Nomor 168 Tahun 2023 tentang Petunjuk Pelaksanaan Pemotongan Pajak atas Penghasilan sehubungan dengan Pekerjaan, Jasa, atau Kegiatan Orang Pribadi. Jakarta: Kementerian Keuangan.
- [13] Republik Indonesia, 2021. Undang-Undang Nomor 7 Tahun 2021 tentang Harmonisasi Peraturan Perpajakan. Jakarta: Sekretariat Negara.
- [14] Judijanto, L., 2024. Pengaruh implementasi ERP pada pengelolaan gaji di perguruan tinggi. Jurnal Manajemen Informatika, 8(3), pp. 144-153.