E-ISSN : 2988-1986
https://ejournal.cahayailmubangsa.institute/index.php/kohesi

Kohesi: Jurnal Multidisiplin Saintek
Volume 10 No 7 Tahun 2025

INTEGRASI KEAMANAN API BERBASIS RBAC DAN AUDIT TRAIL PADA BACKEND SISTEM INFORMASI EKSEKUTIF REKRUTMEN UNTUK OPTIMALISASI PENGAMBILAN KEPUTUSAN STRATEGIS (STUDI KASUS: PT SOLOMON INDO GLOBAL)

Karina Catur Febriantika[1], Zikhaila Diva Priamita[2], Agung Brastama Putra[3], Mohamad Irwan Afandi[4]
[1,2,3,4] Program Studi Sistem Informasi, Universitas Pembangunan Nasional "Veteran" Jawa Timur
E-mail: 23082010024@student.upnjatim.ac.id[1], 23082010024@student.upnjatim.ac.id[2], agungbp.si@upnjatim.ac.id[3], mohamadafandi.si@upnjatim.ac.id[4]

ABSTRACT

Backend development for recruitment Executive Information Systems (SIE) faces the challenge of aligning fast statistical data presentation (low latency) with strict protection of sensitive data. This research aims to design a REST API architecture using NestJS and PostgreSQL which focuses on two main pillars. First, to support executive needs in making strategic decisions, optimization of data aggregation at the database level (database-level aggregation) is carried out. This enables responsive presentation of statistical metrics and drill-down visualization features without burdening server memory. Second, implementing layered security (Defense in Depth) through JWT authentication, role-based authorization (Role-Based Access Control/RBAC), and Bcrypt hashing. In addition, an Audit Trail mechanism is implemented on the history table to ensure accountability and the principle of Non-Repudiation. Test results show that this architecture is successful in providing accurate and fast SIE data, while mitigating the risk of illegal access and data manipulation through strict access controls.

Keywords: Executive Information Systems (SIE), Backend Security, Role-Based Access Control, Audit Trail, Drill-down, NestJS.

ABSTRAK

Pengembangan backend untuk Sistem Informasi Eksekutif (SIE) rekrutmen menghadapi tantangan dalam menyelaraskan penyajian data statistik yang cepat *(low latency)* dengan perlindungan ketat terhadap data sensitif. Penelitian ini bertujuan merancang arsitektur REST API menggunakan NestJS dan PostgreSQL yang difokuskan pada dua pilar utama. Pertama, untuk mendukung kebutuhan eksekutif dalam pengambilan keputusan strategis, dilakukan optimalisasi agregasi data di level basis data (*database-level aggregation*). Hal ini memungkinkan penyajian metrik statistik dan fitur visualisasi *drill-down* secara responsif tanpa membebani memori server. Kedua, penerapan keamanan berlapis (*Defense in Depth*) melalui autentikasi JWT, otorisasi berbasis peran (*Role-Based Access Control*/RBAC), dan *hashing Bcrypt*. Selain itu, mekanisme *Audit Trail* diimplementasikan pada tabel histori untuk menjamin akuntabilitas dan prinsip *Non-Repudiation*. Hasil pengujian menunjukkan bahwa arsitektur

Article History
Received: Desember 2025
Reviewed:Desember 2025
Published: Desember 2025

Plagiarism Checker No 235
Prefix DOI : 10.8734/Kohesi.v1i2.365
Copyright : Author
Publish by : Kohesi

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

ini berhasil menyediakan data SIE yang akurat dan cepat, sekaligus memitigasi risiko akses ilegal serta manipulasi data melalui kontrol akses yang ketat.

Kata Kunci: ***Sistem Informasi Eksekutif (SIE), Backend Security, Role-Based Access Control, Audit Trail, Drill-down, NestJS.***

1. PENDAHULUAN

Tantangan utama dalam pengembangan backend untuk Sistem Informasi Eksekutif (SIE) terletak pada upaya menyeimbangkan kebutuhan akan performa akses data yang tinggi dengan tuntutan perlindungan informasi yang ketat. Pada konteks PT Solomon Indo Global, sistem rekrutmen mengelola data yang bersifat sensitif dan rahasia karena mengandung Informasi Identitas Pribadi (*Personally Identifiable Information*/PII), seperti data identitas kandidat, riwayat pekerjaan, serta informasi kontak. Data semacam ini memiliki risiko tinggi apabila diakses oleh pihak yang tidak berwenang, sehingga sistem backend tidak hanya dituntut mampu menyajikan data secara cepat, tetapi juga harus menjamin keamanan, kerahasiaan, dan akuntabilitas penggunaan data. Di sisi lain, kebutuhan eksekutif menuntut akses yang instan terhadap laporan statistik dan visualisasi interaktif untuk mendukung pengambilan keputusan yang bersifat strategis dan tepat waktu. Kondisi ini menjadikan perancangan arsitektur backend sebagai aspek krusial yang harus dibangun di atas fondasi keamanan informasi yang mencakup prinsip kerahasiaan (*confidentiality*), integritas (*integrity*), dan ketersediaan (*availability*). Sejumlah penelitian terdahulu telah mengkaji penerapan sistem informasi dalam konteks manajemen sumber daya manusia. Penelitian oleh (Ferdian et al., 2024) menunjukkan bahwa digitalisasi sistem kepegawaian mampu meningkatkan efisiensi pengelolaan data karyawan serta meminimalkan risiko kehilangan data akibat proses manual. Meskipun demikian, fokus utama penelitian tersebut masih terbatas pada pengelolaan data administratif umum, seperti pencatatan data pegawai dan proses kepegawaian dasar.

Aspek keamanan *Application Programming Interface* (API) tingkat lanjut serta mekanisme optimalisasi agregasi data untuk kebutuhan dashboard eksekutif yang responsif belum menjadi perhatian utama. Padahal, sistem informasi eksekutif menuntut penyajian data yang telah diolah dan diringkas secara real-time agar dapat langsung dimanfaatkan dalam proses pengambilan keputusan. Penelitian lain yang relevan adalah kajian keamanan sistem oleh Mihajlovic (2023) yang membahas penggunaan JSON *Web Token* (JWT) sebagai metode autentikasi stateless dalam pengembangan aplikasi modern. JWT dinilai efektif dalam meningkatkan performa sistem karena tidak memerlukan penyimpanan sesi di sisi server.

Namun, penerapan JWT dalam banyak studi sering kali hanya berfokus pada aspek autentikasi dan otorisasi dasar, tanpa disertai mekanisme akuntabilitas internal yang memadai. Ketiadaan pencatatan jejak aktivitas pengguna atau *Audit Trail* dapat menjadi celah dalam pengawasan internal, khususnya pada sistem yang mengelola data strategis dan sensitif. Tanpa *Audit Trail*, aktivitas seperti akses data, perubahan informasi, atau mutasi data penting tidak dapat ditelusuri secara jelas apabila terjadi penyalahgunaan atau pelanggaran kebijakan. Sementara itu, (Ferraiolo et al., 2019) menegaskan bahwa *Role-Based Access Control* (RBAC) merupakan pendekatan yang efektif dalam membatasi hak akses pengguna berdasarkan peran dan tanggung jawabnya di dalam organisasi. RBAC mampu mencegah akses tidak sah dengan memastikan bahwa setiap pengguna hanya dapat mengakses sumber daya yang sesuai dengan perannya. Namun, penelitian tersebut belum mengintegrasikan konsep RBAC dengan kebutuhan penyajian data statistik hierarkis yang menjadi karakteristik utama Sistem Informasi Eksekutif, khususnya dalam mendukung fitur *drill-down*. Dalam konteks dashboard eksekutif, pembatasan akses tidak hanya berlaku pada level halaman atau modul, tetapi juga pada level data agregat dan detail, sehingga sistem harus mampu mengelola hak akses secara granular tanpa mengorbankan performa. Berdasarkan keterbatasan yang ditemukan pada penelitian-penelitian terdahulu, penelitian ini mengusulkan rancangan arsitektur *backend* terpusat yang secara khusus dikembangkan untuk mendukung kebutuhan Sistem Informasi Eksekutif di PT Solomon Indo Global. *Backend* sistem dibangun menggunakan framework NestJS yang mengadopsi arsitektur modular dan prinsip *dependency injection*, sehingga memudahkan pengelolaan kode, peningkatan skalabilitas, serta pemeliharaan sistem dalam jangka panjang. PostgreSQL digunakan sebagai basis data relasional karena kemampuannya dalam menangani *query* kompleks, transaksi yang konsisten, serta dukungan terhadap fungsi agregasi data yang efisien.

Optimalisasi performa sistem dicapai melalui penerapan agregasi data langsung di level basis data. Data rekrutmen diolah menggunakan *query* agregat yang dirancang khusus untuk menyajikan statistik sesuai kebutuhan *dashboard* eksekutif, seperti tren jumlah pelamar berdasarkan waktu, kategori pelamar, dan status rekrutmen. Pendekatan ini memungkinkan fitur visualisasi *drill-down* berjalan secara responsif, karena data yang dikirim ke frontend telah melalui proses peringkasan dan pengelompokan yang efisien (Kurniawan et al., 2025). Dengan demikian, beban pemrosesan di sisi aplikasi dapat dikurangi, sekaligus memastikan bahwa eksekutif memperoleh informasi secara instan tanpa penundaan yang signifikan. Dari sisi keamanan, sistem backend menerapkan strategi keamanan berlapis dengan mengintegrasikan RBAC dan *Audit Trail* secara terpadu. RBAC digunakan untuk mengatur hak akses pengguna berdasarkan peran, seperti administrator, manajer HR, dan eksekutif, sehingga setiap peran

hanya dapat mengakses data dan fungsi yang relevan. Sementara itu, *Audit Trail* diimplementasikan untuk mencatat setiap aktivitas penting dalam sistem, termasuk proses login, akses data strategis, serta perubahan atau mutasi data rekrutmen. Pencatatan ini tidak hanya berfungsi sebagai mekanisme pengawasan internal, tetapi juga sebagai bentuk pertanggungjawaban dan kepatuhan terhadap standar keamanan informasi korporasi. Sinergi antara agregasi data yang responsif dan penerapan keamanan berlapis menjadikan arsitektur backend yang diusulkan mampu menjawab dua kebutuhan utama secara bersamaan, yaitu performa dan keamanan. Sistem tidak hanya mampu menyajikan data statistik secara cepat untuk mendukung pengambilan keputusan eksekutif, tetapi juga memastikan bahwa setiap akses dan perubahan data tercatat serta terlindungi dari penyalahgunaan. Dengan pendekatan ini, backend Sistem Informasi Eksekutif di PT Solomon Indo Global dinilai memiliki tingkat keandalan, akuntabilitas, dan keamanan yang sesuai dengan kebutuhan organisasi modern yang mengelola data strategis dan sensitif (Supriyanto et al., 2022).

2. METODE PENELITIAN

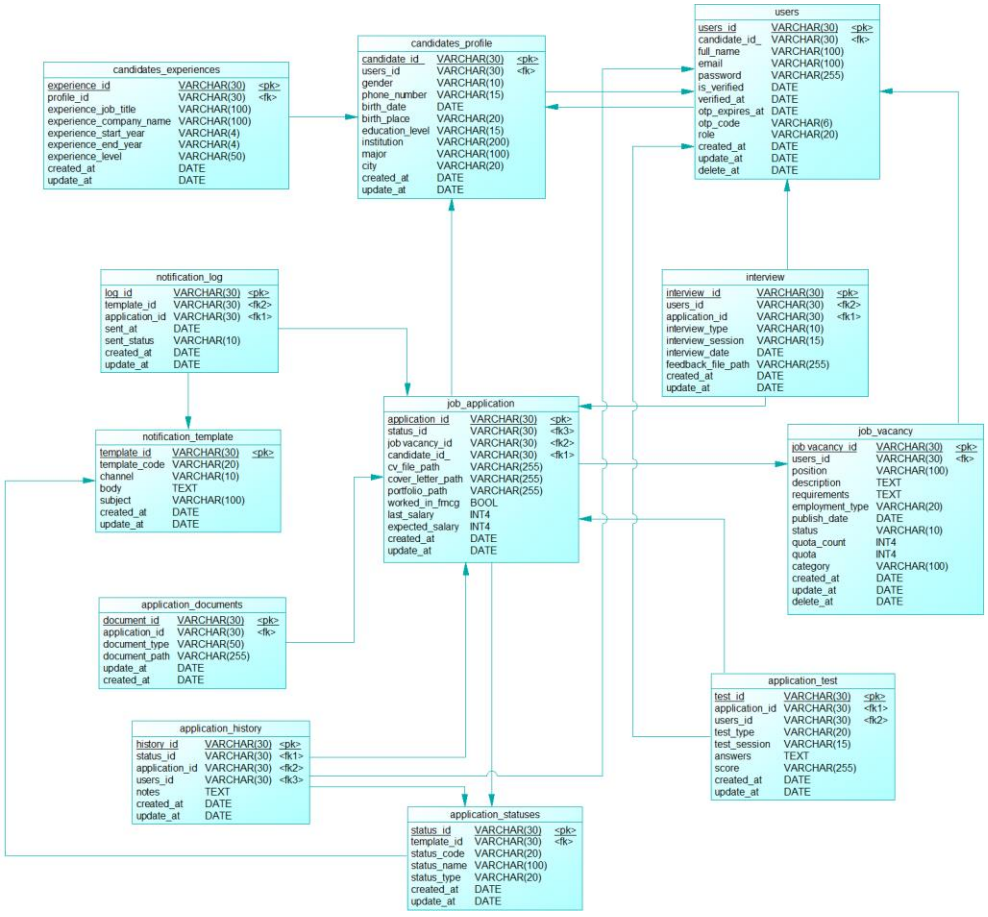
Metode yang digunakan dalam penelitian ini berfokus pada perancangan basis data dan arsitektur keamanan *backend* untuk mendukung Sistem Informasi Eksekutif (SIE) yang responsif dan aman. Basis data dirancang menggunakan Prisma ORM dengan PostgreSQL sebagai sistem manajemen basis data relasional, di mana perancangan Physical Data Model (PDM) menitikberatkan pada normalisasi tabel *job_application* dan *application_history*. Pendekatan ini bertujuan untuk mendukung operasi agregasi data seperti COUNT dan GROUP BY secara efisien, sehingga proses pengambilan data statistik untuk *dashboard* eksekutif dapat dilakukan dengan latensi minimal. Di sisi keamanan, sistem menerapkan arsitektur keamanan bertingkat (*layered security*) yang mencakup beberapa mekanisme proteksi utama. Perlindungan kredensial pengguna dilakukan dengan menerapkan algoritma *hashing Bcrypt* menggunakan *salt rounds* sebesar 10 untuk memastikan kata sandi tersimpan dalam bentuk terenkripsi dan aman dari upaya pembobolan basis data. Selanjutnya, mekanisme autentikasi dan otorisasi diimplementasikan menggunakan pustaka *Passport*-JWT untuk mendukung manajemen sesi secara *stateless*, yang dipadukan dengan *guard* kustom guna memvalidasi penerapan *Role-Based Access Control* (RBAC). Pendekatan ini memastikan bahwa setiap pengguna hanya dapat mengakses data dan fungsi sistem sesuai dengan peran yang dimilikinya. Selain itu, aspek akuntabilitas sistem diperkuat melalui implementasi audit trail dengan memanfaatkan trigger basis data atau logika layanan pada sisi backend untuk mencatat setiap aktivitas dan mutasi data secara sistematis. Pencatatan ini bertujuan untuk memenuhi prinsip *non-repudiation*,

sehingga setiap perubahan data dapat ditelusuri kembali secara jelas dan dapat dipertanggungjawabkan.

3. HASIL DAN PEMBAHASAN

Implementasi Arsitektur *Database*

Struktur basis data (Gambar 1) menempatkan tabel job_application sebagai entitas pusat transaksi.



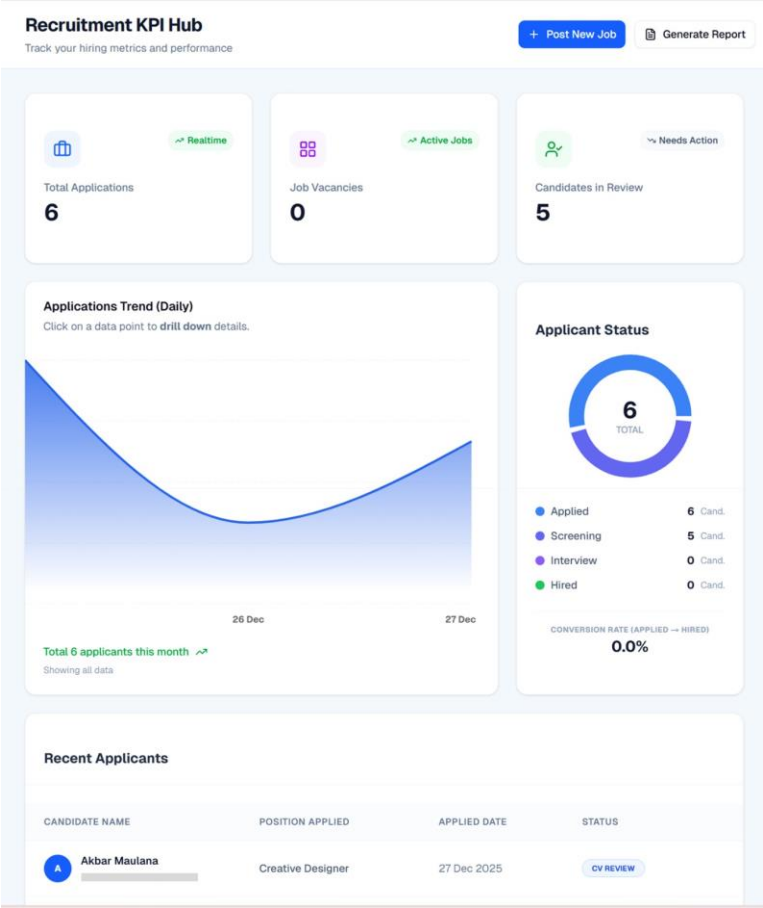
Gambar 1. Physical Data Model (PDM) Sistem Rekrutmen

Untuk mendukung kebutuhan SIE, backend menyediakan endpoint khusus /dashboard/status. Alih-alih mengambil ribuan baris data mentah ke aplikasi (*fetching all rows*), sistem melakukan kalkulasi statistik langsung di level query database. Pendekatan ini secara signifikan mengurangi beban memori server dan mempercepat waktu respons API (*Response Time*), sehingga visualisasi *drill-down* di *frontend* terasa responsif meskipun volume data pelamar meningkat.

Optimalisasi Agregasi Data dan Visualisasi Drill-down

Kebutuhan utama dalam Sistem Informasi Eksekutif (SIE) adalah penyajian informasi yang interaktif. Dalam standar Sistem Informasi Eksekutif (SIE), kemampuan *drill-down* sangat vital bagi eksekutif untuk melakukan analisis mendalam dari data makro ke mikro.

Untuk mendukung hal tersebut, backend tidak sekadar mengirimkan data mentah ke frontend. Sistem mengimplementasikan *Database-level* Aggregation menggunakan perintah GROUP BY dan COUNT langsung pada level query PostgreSQL. Hasilnya, visualisasi pada dashboard dapat merespons permintaan data secara instan tanpa latensi yang berarti.



Gambar 2. Implementasi Dashboard SIE dengan Fitur Drill-down

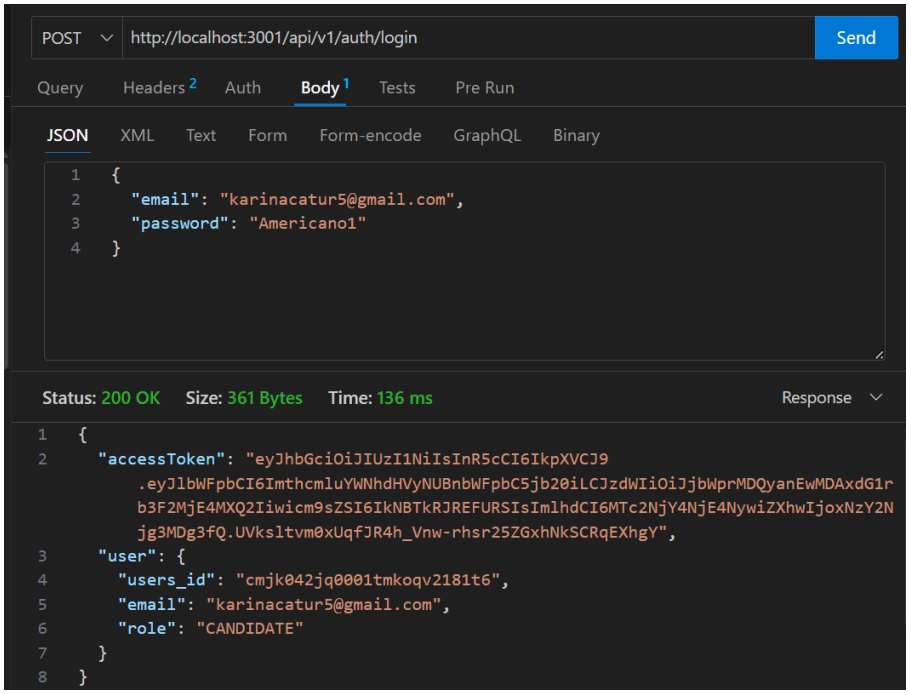
Gambar 2 menunjukkan visualisasi grafik tren rekrutmen yang memiliki fitur drill-down. Efisiensi tampilan ini merupakan hasil dari desain endpoint API yang responsif. Dengan memproses kalkulasi di server (*backend*), beban memori pada sisi klien (*frontend*) berkurang drastis. Hal ini sejalan dengan penelitian (Ferdian et al., 2024) bahwa pengolahan data yang optimal di sisi server sangat mempengaruhi kecepatan akses informasi bagi pihak manajemen.

Implementasi Role-Based Access Control (RBAC)

Keamanan akses pada arsitektur backend ini dikendalikan melalui mekanisme RBAC yang ketat untuk memastikan integritas data eksekutif. Merujuk pada penelitian Kurniawan et al. (2025), evaluasi keamanan sistem informasi harus mempertimbangkan risiko eksploitasi pada kontrol akses. Oleh karena itu, pengujian dilakukan dengan mengacu pada standar OWASP *Application Security Verification Standard* (ASVS) untuk memastikan tidak terdapat kerentanan *Broken Access Control*. Dalam implementasinya, *middleware* pada backend akan mencegat

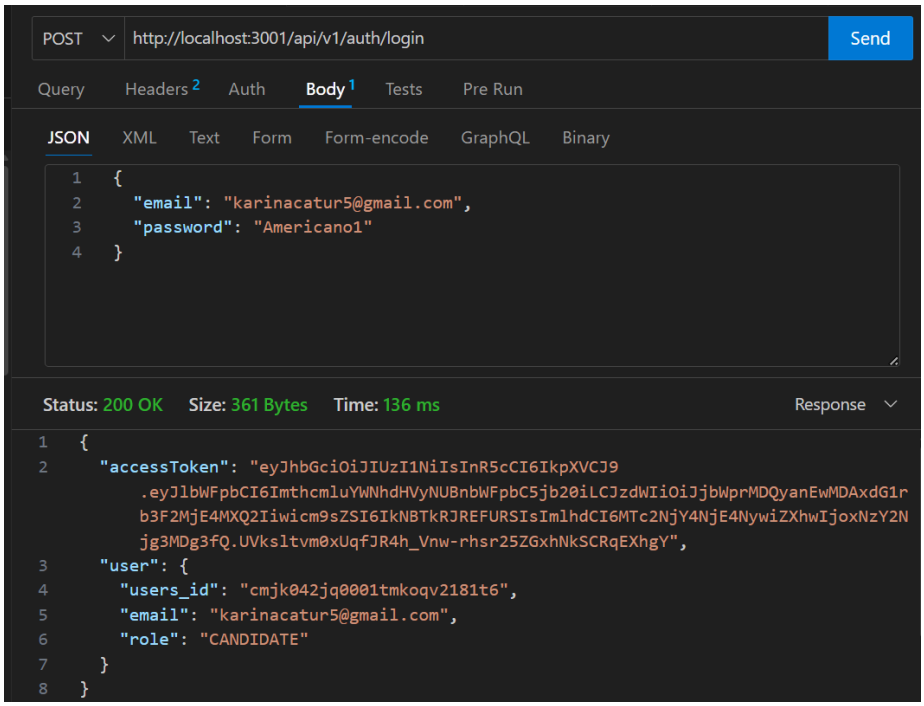
setiap *request* untuk memvalidasi peran pengguna sebelum memberikan akses ke *endpoint*. Mekanisme kontrol akses ini dibagi menjadi dua kategori utama:

1. Role ADMIN: Diberikan akses penuh ke seluruh endpoint manajemen pelamar, pengelolaan lowongan, dan *dashboard* SIE.
2. Role *CANDIDATE*: Akses dibatasi secara ketat hanya pada data milik pribadi (Row-Level Security logic) seperti pengunggahan CV dan pemantauan riwayat lamaran.



Gambar 3. *Response* JSON *Success* Login sebagai *Role CANDIDATE*.

Gambar 3 menampilkan respons login untuk pengguna dengan level Kandidat. Objek pengguna ini memiliki atribut role bernilai '*CANDIDATE*' yang secara otomatis membatasi hak aksesnya di dalam sistem.



Gambar 4. *Response* JSON *Success Login* sebagai *Role* ADMIN..

Gambar 4 menunjukkan respons sistem saat pengguna dengan level Admin melakukan login. Backend mengembalikan status 200 OK disertai dengan *accessToken* (JWT) yang mengandung klaim otoritas administratif ('ADMIN') untuk membuka akses penuh ke dashboard manajemen. Berdasarkan hasil pengujian pada Gambar 3 dan Gambar 4, sistem secara konsisten membedakan respons token berdasarkan klaim peran *(role claim)* pengguna. Pengujian lebih lanjut menunjukkan bahwa upaya akses ke endpoint administratif menggunakan token kandidat menghasilkan *respons* 403 *Forbidden*. Sebagaimana ditekankan oleh (Kurniawan et al., 2025), validasi peran yang dilakukan di sisi server *(server-side validation)* adalah pertahanan utama dalam memitigasi serangan eskalasi hak akses *(privilege escalation)*. Hal ini membuktikan bahwa mekanisme otorisasi yang dibangun telah memenuhi standar keamanan untuk melindungi data sensitif rekrutmen di PT Solomon Indo Global.

Audit Trail dan Prinsip Non-Repudiation

Salah satu pilar utama keamanan informasi adalah *Non-Repudiation* (nirsangkal). Penelitian ini menjawab tantangan tersebut melalui implementasi tabel *application_history* yang mencatat setiap aktivitas mutasi data. Berdasarkan standar yang dikemukakan oleh Sahyudi & Susanto (2025), mekanisme Audit Trail sangat vital untuk menjamin akuntabilitas dan kepatuhan regulasi pada sistem enterprise.

```
1. FUNGSI VALIDATE USER
  async validateUser(email: string, pass: string): Promise<any> {
  const user = await (this.prisma as any).users.findUnique({ where: { email } });

    if (user) {

      if (user.delete_at) {
        throw new UnauthorizedException('Akun ini telah dihapus. Silakan daftar lagi jika Anda ingin bergabung kembali');
      }

      const isPasswordValid = await bcrypt.compare(pass, user.password);

      // Cek verifikasi (Admin sudah kita abaikan di sini, tetapi kandidat wajib verif)
      if (user && isPasswordValid) {
        if (user.role !== 'ADMIN' && !user.is_verified) {
          throw new UnauthorizedException('Akun Anda belum terverifikasi. Silakan periksa email Anda atau selesaikan verifikasi OTP
        }

        const { password, otp_code, otp_expires_at, ...result } = user;
        return result;
      }

      return null;
    }
}
```

Gambar 5. Implementasi Logika Validasi dan Normalisasi Role di Backend.

Sebelum data dicatat ke dalam histori, sistem memastikan konsistensi role melalui logika validasi di sisi *backend*. Seperti terlihat pada Gambar 5, potongan kode pada auth.service.ts menunjukkan proses pengecekan role secara spesifik sebelum token diterbitkan, guna menjamin bahwa aktor yang melakukan perubahan data adalah pihak yang terotorisasi.

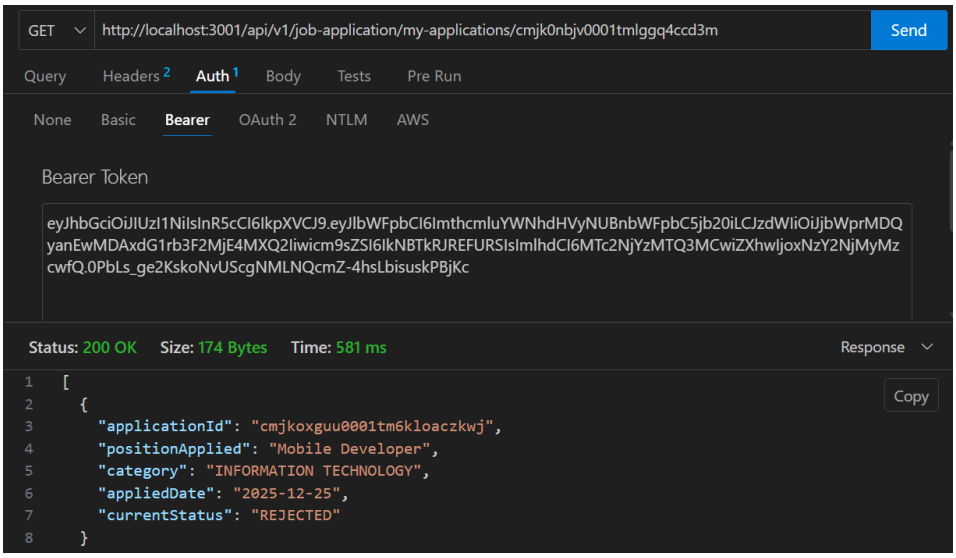
	status_id	application_id	users_id	notes	created_at	update_at
	search status_id...	search application_id...	search users_id...	search notes...	search created_at...	search update_at
‹1yi2h7…	cmjk6qh4u0000tmkovvvv00…	cmjkoxguu0001tm6kloaczk…	cmidtestadmin12345678	Lolos sortir berkas	2025-12-25T01:48:42.389Z	2025-12-25T01
‹ixes59…	cmjk6qh4u0000tmkovvvv00…	cmjkoxguu0001tm6kloaczk…	cmidtestadmin12345678	Lolos sortir berkas	2025-12-25T01:59:51.479Z	2025-12-25T01
4mobapj…	cmjkiststat0001tmkoabc8…	cmjkoxguu0001tm6kloaczk…	cmidtestadmin12345678	Kandidat sangat kompete…	2025-12-25T02:28:24.981Z	2025-12-25T02
43muuac…	cmjk6qh4u0000tmkovvvv00…	cmjkoxguu0001tm6kloaczk…	cmidtestadmin12345678	Mohon maaf, kualifikasi…	2025-12-25T02:32:04.261Z	2025-12-25T02

Open in new tab Skip to unconnected records

Gambar 6. Pencatatan Histori Lamaran (Audit Trail) pada Tabel application_history

Hasil dari pencatatan tersebut dapat dilihat pada Gambar 6, di mana setiap perubahan status pelamar (misal: dari '*Applied*' ke '*Rejected*') terekam secara permanen dengan atribut sebagai berikut:

1. Timestamp (created_at): Menunjukkan waktu presisi saat aksi dilakukan.
2. Aktor (users_id): Mengidentifikasi Admin yang bertanggung jawab atas perubahan tersebut.
3. Konteks (notes): Mencatat alasan keputusan secara transparan (misal: "Kualifikasi tidak sesuai").



Gambar 7. Implementasi Mekanisme Keamanan Bearer Token JWT.

Sebagai bukti pendukung integritas sistem, Gambar 7 menampilkan respons sukses login yang menyertakan Bearer Token JWT. Token ini membawa klaim identitas aktor yang nantinya akan dicatat oleh sistem ke dalam tabel histori setiap kali aktor tersebut melakukan perubahan data. Sinergi antara JWT dan Audit Trail ini memungkinkan perusahaan melakukan audit forensik jika terjadi sengketa hasil rekrutmen, karena seluruh jejak digital tersimpan secara permanen dan tidak dapat dimanipulasi.

PEMBAHASAN

Pembahasan hasil implementasi arsitektur database dan keamanan backend pada Sistem Informasi Eksekutif (SIE) menunjukkan bahwa rancangan sistem telah mampu memenuhi kebutuhan utama organisasi, yaitu penyajian data rekrutmen yang cepat, interaktif, dan aman. Struktur basis data yang menempatkan tabel *job_application* sebagai entitas pusat transaksi mencerminkan pendekatan desain yang tepat, mengingat tabel ini menjadi sumber utama seluruh aktivitas rekrutmen. Normalisasi tabel pendukung seperti *application_history* memungkinkan pemisahan antara data operasional dan data histori, sehingga sistem dapat mengelola transaksi aktif sekaligus menjaga rekam jejak perubahan data secara sistematis. *Desain Physical Data Model* (PDM) ini berkontribusi langsung terhadap *efisiensi query*, terutama dalam mendukung kebutuhan agregasi data pada dashboard eksekutif. Implementasi endpoint khusus */dashboard/status* menjadi bukti bahwa backend dirancang dengan mempertimbangkan karakteristik Sistem Informasi Eksekutif yang membutuhkan respons cepat terhadap permintaan data statistik. Alih-alih mengambil seluruh data mentah dalam jumlah besar ke lapisan aplikasi, sistem memproses kalkulasi statistik langsung di tingkat basis data PostgreSQL. Pendekatan ini secara signifikan mengurangi beban memori server dan meminimalkan waktu respons API. Dengan melakukan agregasi data di level database menggunakan perintah GROUP BY dan

COUNT, backend mampu mengirimkan data yang sudah terstruktur dan siap divisualisasikan oleh *frontend*. Hasilnya, fitur *drill-down* pada dashboard dapat berjalan secara responsif meskipun jumlah data pelamar terus bertambah, tanpa menurunkan performa sistem secara keseluruhan. Optimalisasi agregasi data ini sejalan dengan karakteristik utama Sistem Informasi Eksekutif yang menekankan penyajian informasi ringkas namun mendalam. Fitur *drill-down* memungkinkan eksekutif melakukan analisis bertingkat dari data makro ke data mikro secara intuitif. Efisiensi visualisasi yang ditampilkan pada *dashboard* merupakan hasil dari desain endpoint API yang berorientasi pada kebutuhan analisis manajerial, bukan sekadar penyajian data mentah (Panuntun et al., 2015).

Hal ini mendukung temuan (Ferdian et al., 2024) yang menyatakan bahwa pengolahan data yang optimal di sisi server berpengaruh besar terhadap kecepatan akses informasi dan kualitas pengambilan keputusan di tingkat manajemen. Dari sisi keamanan, implementasi *Role-Based Access Control* (RBAC) menjadi komponen krusial dalam menjaga integritas dan kerahasiaan data rekrutmen. Sistem secara konsisten membedakan hak akses pengguna berdasarkan peran yang dimiliki, sehingga setiap permintaan ke *endpoint backend* divalidasi terlebih dahulu oleh *middleware*. Pendekatan ini memastikan bahwa hanya pengguna dengan otorisasi yang sesuai yang dapat mengakses data strategis, terutama data yang ditujukan bagi eksekutif. Pengujian keamanan dilakukan dengan mengacu pada standar OWASP *Application Security Verification Standard* (ASVS) untuk memitigasi risiko *Broken Access Control*, sebagaimana direkomendasikan oleh (Kurniawan et al., 2025). Hasil pengujian menunjukkan bahwa backend mampu menolak upaya akses tidak sah secara konsisten melalui mekanisme validasi peran di sisi *server*. Penerapan RBAC juga terlihat jelas pada perbedaan respons sistem saat proses autentikasi antara pengguna dengan peran CANDIDATE dan ADMIN. Respons JSON login yang dihasilkan backend secara eksplisit memuat klaim peran *(role claim)* di dalam token JWT. Token dengan klaim CANDIDATE membatasi akses pengguna hanya pada data milik pribadi melalui penerapan logika *row-level security*, seperti pengunggahan CV dan pemantauan status lamaran. Sebaliknya, token dengan klaim ADMIN memberikan akses penuh terhadap endpoint manajemen pelamar, pengelolaan lowongan, serta dashboard SIE. Pengujian lanjutan membuktikan bahwa penggunaan token kandidat untuk mengakses endpoint administratif akan menghasilkan respons 403 *Forbidden*, yang menegaskan bahwa mekanisme otorisasi telah berjalan sesuai prinsip least *privilege* (Ginting, 2018).

Selain kontrol akses, penelitian ini juga menekankan pentingnya akuntabilitas sistem melalui penerapan *Audit Trail* untuk memenuhi prinsip *non-repudiation*. Tabel *application_history* berfungsi sebagai repositori histori yang mencatat setiap mutasi data

rekrutmen secara permanen. Setiap perubahan status pelamar direkam lengkap dengan atribut waktu (*timestamp*), aktor yang melakukan aksi (*users_id)*, serta konteks keputusan dalam bentuk catatan (*notes*). Mekanisme ini memungkinkan perusahaan menelusuri kembali seluruh proses pengambilan keputusan secara transparan dan terstruktur. Pendekatan ini sejalan dengan standar keamanan informasi enterprise yang dikemukakan oleh (Sahyudi & Susanto, 2025), di mana *Audit Trail* menjadi elemen penting dalam menjamin akuntabilitas dan kepatuhan regulasi. Integrasi antara JWT dan *Audit Trail* memperkuat aspek keamanan sistem secara menyeluruh. Klaim identitas yang dibawa oleh *Bearer* Token JWT tidak hanya berfungsi untuk otorisasi akses, tetapi juga digunakan sebagai referensi aktor dalam pencatatan histori perubahan data. Dengan demikian, setiap aksi yang dilakukan oleh pengguna dapat dipetakan secara jelas ke identitas dan perannya dalam sistem. Sinergi ini memungkinkan dilakukannya audit forensik apabila terjadi sengketa atau permasalahan terkait hasil rekrutmen, karena seluruh jejak digital tersimpan secara konsisten dan tidak dapat dimanipulasi. Secara keseluruhan, hasil implementasi dan pengujian menunjukkan bahwa arsitektur database dan keamanan backend yang dibangun telah memenuhi kebutuhan performa, keamanan, dan akuntabilitas Sistem Informasi Eksekutif di PT Solomon Indo Global (Nasution, 2018).

KESIMPULAN

Berdasarkan hasil pembahasan, dapat disimpulkan bahwa implementasi arsitektur database pada Sistem Informasi Eksekutif (SIE) rekrutmen telah dirancang secara terstruktur dan efisien untuk mendukung kebutuhan analisis manajerial. Penempatan tabel *job_application* sebagai entitas pusat transaksi, yang didukung oleh tabel *application_history*, memungkinkan sistem mengelola data operasional dan histori secara terpisah namun saling terintegrasi. Penerapan agregasi data langsung di tingkat basis data melalui *endpoint* khusus dashboard terbukti mampu meningkatkan performa sistem dengan menurunkan beban memori server dan mempercepat waktu respons API. Pendekatan ini menjadikan visualisasi data dan fitur *drill-down* pada *dashboard* eksekutif tetap responsif meskipun volume data pelamar meningkat, sehingga informasi strategis dapat disajikan secara tepat waktu dan relevan bagi pengambilan keputusan. Selain aspek performa, sistem juga menunjukkan tingkat keamanan dan akuntabilitas yang tinggi melalui penerapan mekanisme keamanan bertingkat. Implementasi *Role-Based Access Control* (RBAC) yang divalidasi di sisi server, pemanfaatan *Bearer* Token JWT, serta pencatatan *Audit Trail* pada tabel histori mampu menjamin kerahasiaan, integritas, dan keterlacakan setiap aktivitas dalam sistem. Perbedaan hak akses antara peran ADMIN dan CANDIDATE diterapkan secara konsisten, sehingga risiko eskalasi hak akses dan penyalahgunaan

data dapat diminimalkan. Dengan terpenuhinya prinsip *non-repudiation* dan standar keamanan aplikasi, arsitektur backend yang dikembangkan dinilai telah memenuhi kebutuhan PT Solomon Indo Global dalam mengelola data rekrutmen yang sensitif secara aman, transparan, dan dapat dipertanggungjawabkan.

DAFTAR PUSTAKA

Sahyudi, M., & Susanto, E. R. (2025). Analisis Implementasi Sistem Keamanan Basis Data Berbasis Role-Based Access Control (RBAC) pada Aplikasi Enterprise Resource Planning. SATESI: Jurnal Sains Teknologi dan Sistem Informasi, 5(1), 105-116.

Gamayanto, I., Kurniawan, M. C., & Sanyoto, G. K. (2025). Security Evaluation of Keycloak-Based Role-Based Access Control in Microservice Architectures Using the OWASP ASVS Framework. Journal of Applied Informatics and Computing, 9(6), 3964-3973.

Yuricha, Y., & Phan, I. K. (2023). Penerapan Role Based Access Control dalam Sistem Supply Chain Management Berbasis Cloud: The Implementation of Role Based Access Control in a Cloud-Based Supply Chain Management System. MALCOM: Indonesian Journal of Machine Learning and Computer Science, 3(2), 339-348.

ARAVINDA, A. K., & Divya, T. L. (2024). Security measures implemented in RESTful API Development. OPEN ACCESS RESEARCH JOURNAL OF ENGINEERING AND TECHNOLOGY Учредители: Open Access Research Journals Publication, 7(1), 105-112.

Tanveer, F., Iradat, F., Iqbal, W., & Ahmad, A. (2025). Towards Secure APIs: A Survey on RESTful API Vulnerability Detection. Computers, Materials & Continua, 84(3).

Ojha, P. R. (2024). Securing the digital frontier: Best practices for RESTful API protection in modern web applications. International Journal of Engineering Trends and Technology Research (IJETR), 9(2), 481–491. https://doi.org/10.5281/zenodo.13860934

Ferdian, F., Fransen, L. A., & Hermawan, H. (2024). Rancang Bangun Sistem Informasi Kepegawaian Berbasis Website Pada PT. XYZ. MSC (MDP Student Conference), 3(1), 917–925.

Putra, I. N. D. P., Astawa, I. G. S., & Mogi, I. K. A. (2024). Aplikasi Manajemen Karyawan Berbasis Website Sebagai Pengelola Karyawan di CV Tridatu Solution. Jurnal Pengabdian Informatika, 2(4), 617–622.

Supriyanto, H., Nurhadi, M., Prasetya, M. S., Hermansyah, D., & Puspitaningrum, A. C. (2022). Pembuatan media informasi digital sebagai sarana informasi dan promosi sekolah. JMM (Jurnal Masyarakat Mandiri), 6(5).

Panuntun, R., Rochim, A. F., & Martono, K. T. (2015). Perancangan papan informasi digital berbasis web pada Raspberry Pi. Jurnal Teknologi dan Sistem Komputer, 3(2), 192–197.

Ginting, R. (2020). Kemampuan literasi media pada era informasi digital di kalangan mahasiswa Kota Medan. Talenta Conference Series: Local Wisdom, Social, and Arts (LWSA), 3(1).