



Evaluasi Kualitas Respons Api Backend pada Aplikasi *Essay Analytic Online* (ESAO) menggunakan *Bruno*

Akhmam Fahmi¹, Navsa Urva^{2*}, Maulana Fakih Latief³

^{1,3} Dosen Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri, Indonesia

² Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri, Indonesia

Email: akhmam.fahmi@nurulfikri.ac.id¹, navs22085ti@student.nurulfikri.ac.id^{2*}, maulana.latief@nurulfikri.ac.id³

*Penulis Korespondensi: navs22085ti@student.nurulfikri.ac.id

Abstract. The Essay Analytic Online (ESAO) application is a smart essay evaluation system based on Artificial Intelligence (AI) that processes large text volumes. The operational reliance on the backend REST API architecture poses challenges related to performance stability and functional integrity, risking server downtime during massive access if left untested. This research aims to evaluate the backend REST API response quality of the ESAO application using the Bruno automation software. The evaluation is focused on fulfilling the Functional Suitability and Performance Efficiency characteristics based on the ISO/IEC 25010 international standard framework. A quantitative experimental approach was applied through the Automated Black Box Testing method by executing six endpoint samples representing various computational workloads for 30 test iterations. The results indicate a 100% functional success rate without any failures in returning HTTP 200 OK status codes and JSON response body integrity. In performance efficiency metrics, the server demonstrated stable response time resilience with an average of under 200 ms for light loads, 693.7 ms for medium loads, and reaching a maximum time of 6,600 ms for extreme loads without encountering any errors. In conclusion, the ESAO backend REST API architecture is proven to be robust, stable, and fully compliant with ISO/IEC 25010 standards, whereby Bruno is also proven to be highly effective as a local file-system-based automated testing instrument.

Keywords: Black Box Testing; Bruno; ESAO; ISO/IEC 25010; REST API.

Abstrak. Aplikasi *Essay Analytic Online* (ESAO) merupakan sistem evaluasi esai cerdas berbasis *Artificial Intelligence* (AI) yang memproses volume teks besar. Ketergantungan operasional pada arsitektur REST API *backend* memunculkan tantangan terkait stabilitas performa dan keutuhan fungsionalitas, yang berisiko menyebabkan kelumpuhan server (*downtime*) saat diakses secara massal jika tidak diuji. Penelitian ini bertujuan untuk mengevaluasi kualitas respons REST API *backend* pada aplikasi ESAO menggunakan perangkat lunak otomasi Bruno. Evaluasi difokuskan pada pemenuhan karakteristik *Functional Suitability* dan *Performance Efficiency* berdasarkan kerangka kerja standar internasional ISO/IEC 25010. Pendekatan eksperimental kuantitatif diterapkan melalui metode *Automated Black Box Testing*, dengan mengeksekusi enam sampel *endpoint* yang mewakili berbagai beban komputasi sebanyak 30 iterasi pengujian. Hasil penelitian menunjukkan tingkat keberhasilan fungsional mencapai 100% tanpa adanya kegagalan pada pengembalian kode status HTTP 200 OK serta keutuhan struktur *response body* JSON. Pada metrik efisiensi kinerja, server menunjukkan ketahanan waktu respons (*response time*) yang stabil dengan rata-rata di bawah 200 ms untuk beban ringan, 693,7 ms untuk beban menengah, dan menyentuh waktu maksimum 6.600 ms pada beban ekstrem tanpa mengalami *error*. Kesimpulannya, arsitektur REST API *backend* ESAO terbukti sangat tangguh, stabil, dan sepenuhnya mematuhi standar ISO/IEC 25010, di mana Bruno juga terbukti sangat efektif sebagai instrumen pengujian otomatis berbasis *file-system* lokal.

Kata Kunci: Bruno; ESAO; ISO/IEC 25010; Pengujian *Black Box*; REST API.

1. LATAR BELAKANG

Perkembangan teknologi kecerdasan buatan atau Artificial Intelligence (AI) di sektor pendidikan telah memicu lahirnya sistem evaluasi cerdas seperti Automated Essay Scoring (AES) yang mampu menilai esai secara otomatis dan objektif (Agisni Bepi Rosadi et al., 2026; Jiyan Suhada et al., 2026). Pemanfaatan model bahasa skala besar (LLM) seperti GPT terbukti

sangat memengaruhi tingkat akurasi penilaian terhadap kompleksitas linguistik teks secara efisien layaknya pakar manusia (Liu, Qi, & Lu, 2025). Aplikasi Essay Analytic Online (ESAO) merupakan wujud platform yang dirancang untuk menangani pemrosesan analitik esai semacam ini. Mengingat ESAO harus memproses format pertukaran data JavaScript Object Notation (JSON) yang berukuran besar dan kompleks, infrastruktur server (backend) harus dirancang dengan sangat tangguh guna menghindari beban memori yang berlebihan (Guna, Rahmatulloh, & Rachman, 2023).

Dalam pengembangan sistem berbasis web modern, Application Programming Interface (API) dengan arsitektur Representational State Transfer (REST) menjadi fondasi utama dalam menjembatani interaksi dan pertukaran antarmuka data backend secara real-time dan terproteksi (Shofiana, Mardiansyah, Parabi, & Syarif, 2024). Kinerja protokol API tersebut sangat krusial terhadap stabilitas sistem secara keseluruhan; parameter seperti waktu respons (response time) dan throughput menjadi indikator kritis dalam mengevaluasi efisiensi dan kehandalan layanan (Fhonna, Afrillia, Ilhadi, Arif, & Selian, 2024). Keterlambatan waktu respons sekecil apa pun, terutama di bawah tekanan lalu lintas pengiriman data operasional yang tinggi, dapat berdampak langsung pada terjadinya penumpukan request dan penurunan performa aplikasi (Azzahidi, Wijayanto, & Darmawan, 2025).

Untuk memastikan seluruh layanan REST API pada backend beroperasi dengan stabil, tahapan pengujian fungsionalitas dan ketahanan perangkat lunak (software testing) mutlak dilakukan secara berkala (Maria Rosario Borroek, et al., 2025). Pendekatan pengujian secara manual dianggap tidak lagi efisien karena memakan banyak waktu, sehingga otomatisasi pengujian (automation testing) kini sangat direkomendasikan untuk menjamin konsistensi proses identifikasi kesalahan (Rahayu, Wahyudin, Firmansyah, & Susanti, 2024). Metode Black Box Testing secara khusus sering diterapkan dalam otomatisasi ini karena berfokus murni pada evaluasi kesesuaian input dan output fungsionalitas sistem tanpa keharusan membedah logika kode sumber internalnya (Fadillah, 2024). Dalam konteks ini, penggunaan alat uji modern berarsitektur *file-system* lokal seperti Bruno menjadi pilihan yang sangat relevan dan efisien untuk mengatasi keterbatasan memori pada alat uji konvensional seperti Postman yang terlalu bergantung pada sinkronisasi cloud.

Evaluasi kehandalan API tersebut harus bersandar pada metrik pengukuran kualitas perangkat lunak internasional yang diakui secara global, yakni model ISO/IEC 25010 (Rojas et al., 2025). Dalam ruang lingkup pengujian API, terdapat dua karakteristik kunci yang paling relevan dari standar ini: Functional Suitability untuk memastikan setiap fungsi memberikan keluaran struktur JSON dan status kode HTTP yang akurat, serta Performance Efficiency untuk

menilai perilaku waktu (time behavior) sistem terhadap beban permintaan (Kurniasari & Rochimah, 2025).

Berdasarkan kajian pustaka terkini, mayoritas riset terkait AES lebih berfokus pada optimasi tingkat akurasi algoritma AI tanpa disertai evaluasi mendalam terhadap ketahanan infrastruktur arsitektur API backend-nya. Hingga saat ini, belum terdapat data empiris mengenai kualitas performa sistem ESAO dalam menangani beban operasional (request data esai) yang nyata. Oleh karena itu, penelitian ini bertujuan untuk secara spesifik mengisi kesenjangan tersebut dengan mengevaluasi kualitas respons REST API backend pada aplikasi ESAO secara otomatis menggunakan alat uji Bruno, dengan berlandaskan pada karakteristik fungsional dan efisiensi kinerja dari standar ISO/IEC 25010.

2. KAJIAN TEORITIS

Kajian teoritis ini menjadi landasan teori yang memuat konsep-konsep fundamental bagi penelitian dalam mengevaluasi kualitas fungsionalitas dan efisiensi kinerja REST API *backend* pada aplikasi *Essay Analytic Online* (ESAO). Pembahasan pada bab ini difokuskan pada arsitektur pertukaran data, metode otomatisasi pengujian, hingga parameter kualitas standar internasional yang digunakan sebagai instrumen pengukuran utama.

Arsitektur *Representational State Transfer* (REST) API dan Format JSON

Dalam arsitektur pengembangan perangkat lunak modern, sistem server (*backend*) memainkan peranan yang sangat esensial dalam memproses logika bisnis dan memfasilitasi kelancaran komunikasi data antarsistem (Fhonna et al., 2024). Sebagai jembatan komunikasi tersebut, *Application Programming Interface* (API) digunakan untuk menghubungkan berbagai aplikasi perangkat lunak yang berbeda platform tanpa mengharuskan pengguna mengetahui kompleksitas logika internal servernya (Guntara & Azkarin, 2023). Arsitektur *Representational State Transfer* (REST) adalah pendekatan yang paling umum digunakan dalam membangun API karena kemampuannya dalam memfasilitasi pertukaran data yang cepat, ringan, dan berskala besar melalui protokol standar *Hypertext Transfer Protocol* (HTTP) (Nurjaman, Utomo, & Hermanto, 2024). REST API beroperasi dengan mengeksplorasi metode instruksi standar seperti GET, POST, PUT, dan DELETE untuk menjalankan operasi basis data dari sisi klien ke server (Muhammad & Paputungan, 2024). Untuk meningkatkan efisiensi komputasi pada pertukaran data yang kompleks, REST API umumnya mendistribusikan balasan responsnya dalam format *JavaScript Object Notation* (JSON), yakni sebuah format penyimpanan data terstruktur yang ringan dan sangat mudah diuraikan oleh mesin (Muhammad & Paputungan, 2024).

Pengujian *Black Box* dan Otomatisasi Pengujian

Untuk menjamin bahwa setiap fungsi layanan REST API tersebut beroperasi sesuai dengan spesifikasi dan terbebas dari kesalahan program (*bug*), tahapan pengujian perangkat lunak (*software testing*) mutlak untuk dilakukan secara berkala. Pengujian dengan pendekatan *Black Box Testing* secara ekstensif diterapkan karena berfokus murni pada evaluasi fungsionalitas eksternal, yakni memverifikasi kesesuaian antara instruksi masukan (*request*) dan keluaran (*response*) tanpa perlu menganalisis struktur kode pemrogramannya secara internal (Shofiana et al., 2024). Mengingat pengujian fungsional secara manual sangat rentan terhadap inefisiensi waktu dan kesalahan penguji manusia (*human error*), pergeseran menuju otomatisasi pengujian (*automation testing*) muncul sebagai solusi modern untuk menjalankan skenario uji secara konsisten, berulang, dan komprehensif (Fadillah, 2024). Penggunaan alat otomatisasi khusus memungkinkan proses identifikasi dan pelaporan kesalahan pada setiap *endpoint* API menjadi jauh lebih efektif serta akurat dalam waktu yang sangat singkat dibandingkan pengujian konvensional (Rahayu et al., 2024). Implementasi instrumen uji seperti Bruno sangat merepresentasikan kebutuhan pengujian otomatis modern karena memfasilitasi eksekusi *script* yang terpusat.

Standar Kualitas Perangkat Lunak ISO/IEC 25010

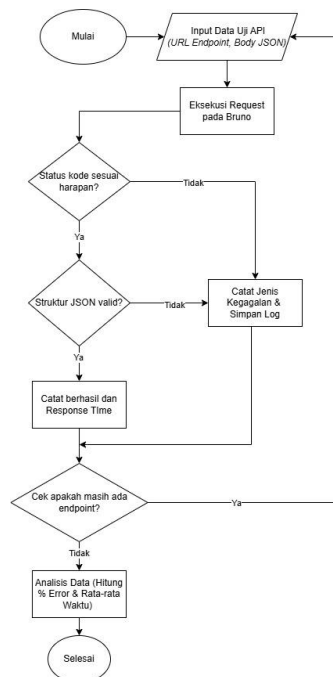
Evaluasi performa terhadap layanan *backend* API tersebut harus diukur secara objektif dengan menggunakan kerangka kerja metrik kualitas perangkat lunak internasional, yaitu standar ISO/IEC 25010 (Anang Novida Harinuridin et al., 2025 Kurniasari & Rochimah, 2025). Model standar ini mendefinisikan karakteristik kualitas secara terstruktur, di mana untuk evaluasi kehandalan sebuah API, fokus utama selalu diarahkan pada aspek *Functional Suitability* dan *Performance Efficiency* (Swari, Suryawan, Sugiarta, Ma-hadewi, & Darmawan, 2023). Karakteristik *Functional Suitability* (Kesesuaian Fungsional) mengevaluasi sejauh mana fitur-fitur pada sistem mampu memberikan hasil keluaran kode status yang benar dan menjalankan tugas sesuai dengan kebutuhan operasional penggunanya (Nursahiba & Enda, 2025). Di sisi lain, *Performance Efficiency* (Efisiensi Kinerja) menilai seberapa efisien daya tanggap dan ketahanan sistem saat memproses volume beban permintaan tertentu (Kurniasari & Rochimah, 2025). Parameter pengukuran paling fundamental pada karakteristik efisiensi ini adalah perilaku waktu (*time behavior*), yakni durasi waktu respons (*response time*) yang diberikan oleh server sejak menerima permintaan hingga merespons, yang sangat krusial agar pengguna akhir tidak mengalami keterlambatan operasional (*delay*) (Swari et al., 2023).

3. METODE PENELITIAN

Bab metode penelitian ini menguraikan secara sistematis pendekatan dan langkah-langkah teknis yang dilakukan untuk mengevaluasi kualitas REST API *backend* pada aplikasi ESAO. Rangkaian tahapan ini mencakup penentuan objek dan sampel uji, penyiapan instrumen serta lingkungan pengujian, hingga penetapan metrik evaluasi yang digunakan untuk menarik kesimpulan yang empiris dan objektif.

Jenis dan Pendekatan Penelitian.

Penelitian ini menerapkan jenis penelitian kuantitatif dengan pendekatan eksperimental melalui metode Black Box Testing yang diotomatisasi (Automated API Testing). Pendekatan black box digunakan karena evaluasi difokuskan sepenuhnya pada validasi fungsionalitas eksternal yakni mengukur kesesuaian antara masukan (request) dan keluaran (response) serta efisiensi kinerja tanpa melakukan intervensi terhadap struktur logika kode internal (source code) aplikasi. Agar penelitian berjalan sistematis, tahapan pelaksanaan disusun berdasarkan alur kerja pengujian perangkat lunak yang terstandarisasi, yang dapat dilihat secara visual pada Gambar 1.



Gambar 1. Diagram Alur Tahapan Penelitian.

Objek dan Sampel Uji.

Objek utama dalam penelitian ini adalah REST API backend pada aplikasi Essay Analytic Online (ESAO). Evaluasi difokuskan pada enam endpoint krusial yang mewakili

variasi beban komputasi server, mulai dari otentikasi, transaksi data, unggah berkas, hingga integrasi kecerdasan buatan (AI). Daftar sampel uji tersebut dirincikan pada Tabel 1 berikut.

Tabel 1. Daftar Sampel Endpoint Uji ESAO.

No	Modul	Method	Endpoint (URI)	Fungsi Utama
1	<i>Auth</i>	<i>POST</i>	/auth/login	Proses <i>login</i> dan pembuatan token JWT.
2	<i>Subject</i>	<i>GET</i>	/subjects/	Mengambil daftar mata kuliah (<i>read data</i>).
3	<i>Exam</i>	<i>POST</i>	/subjects/.../exams	Membuat ujian baru dengan fitur unggah <i>file</i> .
4	<i>Student</i>	<i>GET</i>	/student/jawaban-exam-detail	Mengambil dan menampilkan detail jawaban ujian siswa.
5	<i>Reports</i>	<i>GET</i>	/reports/.../download	Mengunduh laporan nilai dalam format Excel (beban memori).
6	<i>Utils</i>	<i>POST</i>	/utils/check-consistency	Pengecekan konsistensi jawaban menggunakan AI.

Lingkungan dan Instrumen Pengujian.

Proses pengumpulan data dilaksanakan pada lingkungan pengujian yang terkendali menggunakan perangkat keras berspesifikasi prosesor Intel Core i5-1235U (12th Gen), RAM 8 GB, sistem operasi Windows 11 64-bit, dan koneksi internet stabil pada kecepatan ± 20 Mbps. Sebagai instrumen pengujian, penelitian ini memanfaatkan perangkat lunak Bruno yang berarsitektur file-system lokal untuk mengeksekusi skenario uji secara terpusat dan berulang.

Skenario Pelaksanaan Uji.

Setiap endpoint API pada Tabel 1 dieksekusi secara otomatis sebanyak 30 kali iterasi ($n=30$) menggunakan fitur Collection Runner pada Bruno. Pemilihan 30 iterasi ini bertujuan untuk memenuhi kaidah sampel statistik minimum guna mendapatkan nilai rata-rata komputasi yang stabil serta mengurangi bias yang diakibatkan oleh fluktuasi latensi jaringan internet.

Evaluasi dan Metrik Pengukuran.

Data metrik yang terekam dari eksekusi pengujian tersebut kemudian dievaluasi secara objektif menggunakan metrik standar kualitas produk perangkat lunak ISO/IEC 25010. Penilaian berfokus pada dua karakteristik utama. Pertama, Functional Suitability (Kesesuaian Fungsional) dievaluasi dengan memvalidasi tingkat keberhasilan setiap iterasi berdasarkan dua parameter mutlak: pengembalian Kode Status HTTP yang sesuai (seperti 200 OK) dan keutuhan struktur response body JSON menggunakan fitur Assertions. Hasilnya dikalkulasikan ke dalam persentase tingkat keberhasilan fungsi (success rate) dari total keseluruhan iterasi

pengujian. Kedua, Performance Efficiency (Efisiensi Kinerja) dianalisis berdasarkan sub-karakteristik perilaku waktu (time behavior). Data waktu respons (response time) yang diukur dalam satuan milidetik (ms) pada setiap iterasi direkapitulasi untuk mendapatkan nilai waktu minimum, maksimum, dan rata-rata atau mean. Nilai rata-rata waktu respons tersebut kemudian diklasifikasikan berdasarkan standar kelayakan performa server untuk menyimpulkan tingkat ketahanan dan responsivitas infrastruktur backend ESAO secara empiris.

4. HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil dari serangkaian pengujian yang telah dieksekusi secara otomatis menggunakan instrumen Bruno terhadap REST API *backend* pada aplikasi *Essay Analytic Online* (ESAO). Pembahasan pada bab ini difokuskan pada pemaparan data empiris terkait tingkat keberhasilan fungsionalitas dan metrik efisiensi waktu respons sistem, yang kemudian dilanjutkan dengan analisis evaluatif untuk mengukur kelayakan infrastruktur *backend* tersebut berdasarkan standar parameter internasional ISO/IEC 25010.

Hasil Pengujian Functional Suitability

Pengujian kesesuaian fungsional (Functional Suitability) bertujuan untuk memvalidasi apakah setiap endpoint API mampu memproses request dan memberikan keluaran yang utuh. Pengujian ini dieksekusi menggunakan perangkat lunak Bruno dengan mengirimkan beban pengujian secara otomatis sebanyak 30 kali iterasi pada masing-masing endpoint. Berdasarkan log eksekusi, seluruh 180 permintaan (6 endpoint dikali 30 iterasi) berhasil diproses oleh server tanpa mengalami kendala.

Hasil validasi sistem pada Bruno mencatat bahwa setiap balasan (response) sukses mengembalikan Kode Status HTTP 200 (OK). Lebih lanjut, fitur Assertions pada Bruno juga memvalidasi bahwa format JavaScript Object Notation (JSON) yang dikembalikan oleh server memiliki struktur yang utuh, tidak rusak (corrupt), dan memuat field data yang sesuai dengan spesifikasi kontrak API aplikasi ESAO. Tingkat keberhasilan fungsional yang menyentuh angka 100% ini secara empiris membuktikan bahwa mekanisme pemrosesan payload, integrasi parameter basis data, serta penanganan kesalahan (error handling) pada backend telah terimplementasi dengan sempurna dan memenuhi karakteristik operasional dari ISO/IEC 25010.

Hasil Pengujian Performance Efficiency

Setelah seluruh endpoint dinyatakan valid secara fungsional, evaluasi dilanjutkan dengan mengukur efisiensi kinerja (Performance Efficiency) yang berfokus pada perilaku

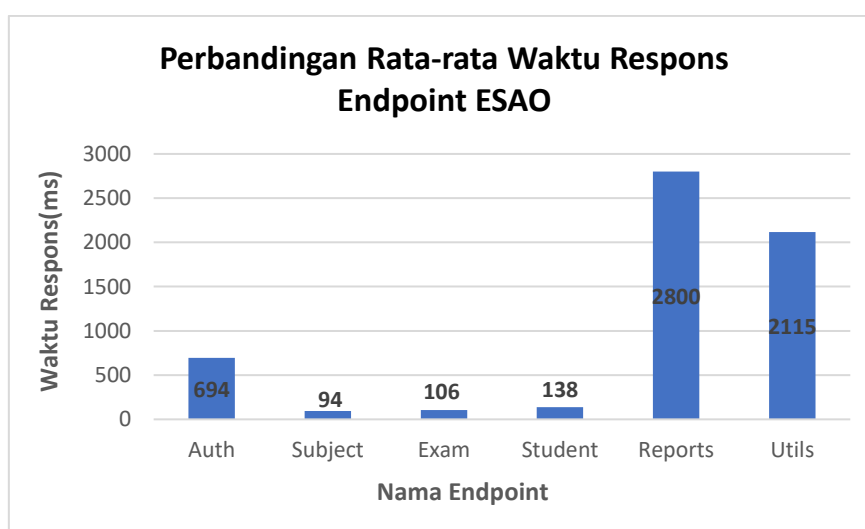
waktu (time behavior). Parameter yang diukur adalah waktu respons (response time) dalam satuan milidetik (ms) selama 30 iterasi. Rekapitulasi nilai minimum, maksimum, dan rata-rata (mean) dari waktu respons tersebut disajikan pada Tabel 2.

Tabel 2. Rekapitulasi Rata-Rata Waktu Respons API ESAO (30 Iterasi).

Iterasi	Auth (ms)	Subject (ms)	Exam (ms)	Student (ms)	Reports (ms)	Utils (ms)
1	475	119	80	88	2100	2300
2	409	113	94	128	1780	1990
3	469	99	84	167	1820	2330
4	413	100	152	77	2110	1710
5	508	73	81	83	1770	2900
6	421	78	77	132	1750	1720
7	708	117	116	111	2900	2110
8	784	92	87	132	2790	1810
9	723	75	98	124	1910	1880
10	717	88	128	125	3180	2300
11	719	97	81	84	3430	1470
12	807	70	93	184	2970	2680
13	806	101	115	125	3490	1860
14	899	94	90	194	2400	3440
15	800	77	119	101	5100	2100
16	757	78	113	176	3830	1680
17	680	78	68	109	3920	2370
18	718	116	92	156	3900	1510
19	726	134	133	75	6600	1500
20	794	77	100	157	4570	1860
21	789	76	142	80	2400	1770
22	761	107	110	103	2130	1860
23	769	56	73	105	1920	1940
24	712	108	160	192	1970	2680
25	684	156	72	111	1750	1760
26	810	91	157	154	2900	2200
27	727	94	107	82	1750	1900
28	713	106	128	565	1790	2500
29	757	92	105	103	1810	1720
30	756	71	111	118	3260	3600
Rata-rata (Mean)	693.7	94.43	105.53	138.03	2800	2115

Waktu Minimum	409	56	68	75	1750	1470
Waktu Maksimum	899	156	160	565	6600	3600

Berdasarkan Tabel 2, terlihat adanya variasi waktu respons yang signifikan antar-endpoint. Untuk memberikan visualisasi yang lebih jelas terkait stabilitas fluktuasi kinerja server dalam menangani beban tersebut, pergerakan waktu respons pada masing-masing iterasi dapat dilihat pada Gambar 2.

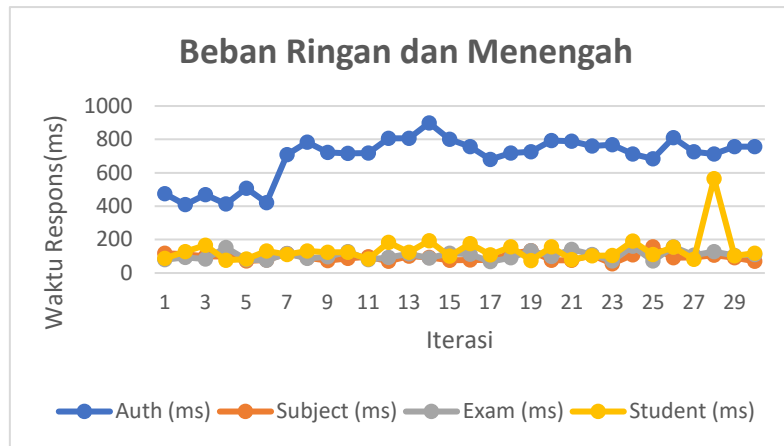


Gambar 2. Grafik Stabilitas dan Rata-rata Waktu Respons Endpoint ESAO.

Pembahasan Evaluasi Kinerja Sistem

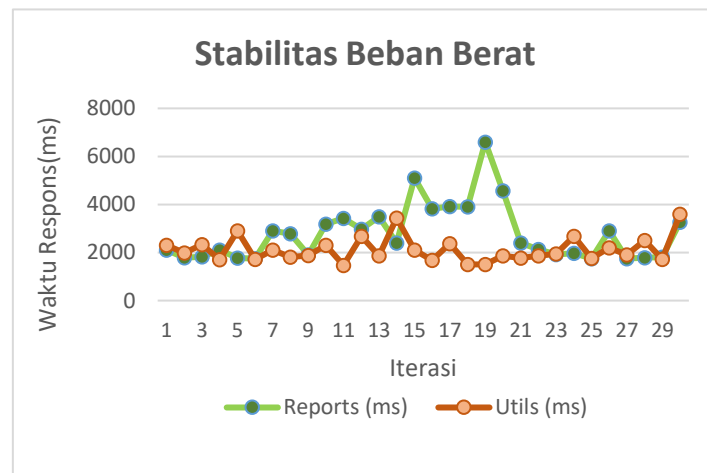
Hasil pengumpulan data empiris pada Gambar 2 dan Tabel 2 diatas menunjukkan bahwa kinerja waktu respons API pada aplikasi ESAO sangat bergantung pada kompleksitas beban komputasi di sisi backend. Performa rata-rata ini dapat diklasifikasikan ke dalam tiga kategori utama: beban ringan, menengah, dan berat.

Pada kelompok beban ringan (Subject, Exam, dan Student), server mampu mempertahankan stabilitas waktu respons secara konsisten di bawah rentang 200 ms. Performa ini secara teknis membuktikan bahwa arsitektur server telah teroptimasi dengan sangat baik untuk menangani operasi penarikan data relasional dasar (query) tanpa menimbulkan jeda pada antarmuka pengguna akhir. Dapat dilihat pada Gambar 3 dibawah ini.



Gambar 3. Stabilitas Waktu Respons API pada Kelompok Beban Ringan dan Menengah.

Pada kelompok beban menengah, endpoint otentikasi (Auth) mencatatkan waktu rata-rata 693,7 ms. Waktu tunggu menengah ini merupakan konsekuensi logis dari beban kerja kriptografi yang harus diproses server untuk memvalidasi kredensial dan mengeksekusi algoritma otentikasi. Penggunaan enkripsi data semacam JSON Web Token (JWT) pada arsitektur API terbukti mengonsumsi waktu komputasi ekstra, namun sangat krusial dan sepadan untuk diterapkan guna menjamin lalu lintas data sistem terproteksi dari akses pihak ketiga yang bersifat ilegal (Shofiana et al., 2024).



Gambar 4. Stabilitas Waktu Respons API pada Kelompok Beban Berat.

Sementara itu, pada Gambar 4 yaitu kelompok beban berat (Reports dan Utils), waktu respons melonjak secara signifikan dengan rata-rata masing-masing 2.800 ms dan 2.115 ms, serta sempat menyentuh waktu maksimum 6.600 ms pada iterasi tertentu. Fenomena ini sangat wajar dan dapat ditoleransi mengingat kedua endpoint tersebut harus memproses ekstraksi basis data menjadi berkas fisik berekstensi Excel dan mengeksekusi sinkronisasi layanan berbasis Artificial Intelligence (AI) yang mengurus memori server secara intensif. Meskipun

grafiknya menunjukkan fluktuasi yang dinamis pada beban berat, server ESAO terbukti sangat tangguh karena mampu merespons 100% request yang masuk secara simultan tanpa mengalami satu pun kelumpuhan (downtime).

Secara keseluruhan, arsitektur REST API backend ESAO terbukti layak, stabil, dan sepenuhnya mematuhi standar parameter Performance Efficiency internasional ISO/IEC 25010, khususnya dalam menyajikan perilaku waktu (time behavior) yang sangat responsif terhadap lonjakan permintaan klien (Kurniasari & Rochimah, 2025; Swari et al., 2023). Di samping itu, keberhasilan pengujian ini juga membuktikan bahwa penerapan otomatisasi testing berbasis eksekusi script mampu mengukur parameter beban server secara cepat, berulang, dan lebih akurat jika dibandingkan dengan pengujian konvensional secara manual (Fadillah, 2024; Rahayu et al., 2024).

5. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian dan analisis yang telah dilakukan terhadap kualitas respons REST API *backend* aplikasi *Essay Analytic Online* (ESAO), dapat ditarik beberapa kesimpulan utama. Pertama, pada karakteristik *Functional Suitability* (Kesesuaian Fungsional), arsitektur *backend* ESAO terbukti beroperasi dengan sangat baik dengan tingkat keberhasilan 100%. Keseluruhan iterasi pengujian otomatis menggunakan Bruno pada keenam *endpoint* sampel berhasil memvalidasi pengembalian Status Kode HTTP 200 OK serta menghasilkan struktur data *JavaScript Object Notation* (JSON) yang utuh tanpa kerusakan (*corrupt*).

Kedua, pada karakteristik *Performance Efficiency* (Efisiensi Kinerja), server ESAO menunjukkan ketahanan dan responsivitas yang sangat stabil dalam menangani berbagai variasi beban komputasi. *Endpoint* dengan beban ringan (penarikan data relasional) hingga menengah (kriptografi otentikasi) mampu memproses data dengan waktu respons rata-rata di bawah 700 milidetik (ms). Meskipun *endpoint* dengan beban komputasi berat seperti ekstraksi dokumen Excel dan integrasi *Artificial Intelligence* (AI) mencatatkan waktu respons maksimum hingga 6.600 ms, sistem tetap beroperasi tanpa mengalami kelumpuhan (*downtime*). Hal ini membuktikan bahwa arsitektur API ESAO telah teroptimasi dengan baik dan layak digunakan secara massal sesuai spesifikasi kualitas internasional ISO/IEC 25010. Selain itu, penggunaan Bruno sebagai perangkat uji otomatis berbasis *file-system* lokal terbukti sangat efektif dan efisien dalam mengeksekusi skenario uji secara berulang tanpa membebani memori komputasi secara berlebih.

Meskipun infrastruktur REST API ESAO telah memenuhi standar fungsional dan efisiensi waktu, terdapat beberapa saran untuk pengembangan penelitian selanjutnya.

Diharapkan peneliti selanjutnya dapat memperluas cakupan evaluasi menggunakan parameter ISO/IEC 25010 lainnya, seperti *Security* (Keamanan) untuk mendeteksi kerentanan celah lapisan aplikasi, serta *Reliability* (Keandalan) untuk mengukur tingkat stabilitas server dalam jangka waktu yang lebih lama. Selain itu, sangat disarankan untuk melakukan pengujian beban ekstrem (*Stress Testing*) menggunakan jumlah permintaan maya (*virtual users*) yang masif guna mengetahui batas maksimal (*bottleneck*) kapasitas server ESAO sebelum mengalami kegagalan sistem (Rochimah, Yuhana, & Hidayat, 2023).

DAFTAR REFERENSI

- Agisni Bepi Rosadi, et al. (2026). Implementasi media digital dan artificial intelligence dalam pembelajaran pendidikan agama Islam. *Jurnal Manajemen dan Pendidikan Agama Islam*, 4(3). <https://doi.org/10.61132/jmpai.v4i3.1970>
- Anang Novida Harinurdin, Primasanti, Y., & Indriastiningsih, E. (2025). Penerapan ISO 45001:2018 terhadap peningkatan produktivitas. *Manufaktur: Publikasi Sub Rumpun Ilmu Keteknikan Industri*, 3(1), 39–55. <https://doi.org/10.61132/manufaktur.v3i1.775>
- Azzahidi, A. S., Wijayanto, B., & Darmawan, A. (2025). Performance evaluation of backend frameworks for REST API: A comparative study of Spring Boot, Flask, Express.js, Laravel FrankenPHP, and Gin. *Jurnal Teknik Informatika (JUTIF)*, 6(4), 2405–2419. <https://doi.org/10.52436/1.jutif.2025.6.4.4811>
- Fadillah, Y. (2024). *Analisis efektivitas penggunaan Katalon Studio untuk automasi pengujian application programming interface (API) pada sistem informasi telekomunikasi [Skripsi/Tesis]*. Bina Sarana Informatika, Jakarta.
- Fhonna, R. P., Afrillia, Y., Ilhadi, V., Arif, A. H., & Selian, R. A. (2024). Performance analysis of API protocol models as recommendations for developers in application development. *JINAV: Journal of Information and Visualization*, 5(2). <https://doi.org/10.35877/454RI.jinav3041>
- Guna, N. S., Rahmatulloh, A., & Rachman, A. N. (2023). Implementasi JSON parsing untuk pertukaran data pada aplikasi VPN client berbasis mobile. *Jurnal Ilmiah NERO*, 8(1). <https://doi.org/10.21107/nero.v8i1>
- Guntara, R. G., & Azkarin, V. (2023). Implementasi dan pengujian REST API sistem reservasi ruang rapat dengan metode black box testing. *Jurnal Minfo Polgan*, 12(1), 1229–1238. <https://doi.org/10.33395/jmp.v12i1.12691>
- Jiyan Suhada, et al. (2026). Peningkatan pemahaman teknologi kecerdasan buatan melalui pelatihan AI generatif pada SMK Al-Muhajirin Depok. *Jurnal Pengabdian dan Keberlanjutan Masyarakat*, 3(2). <https://doi.org/10.62383/bersama.v3i2.3210>
- Kurniasari, D. T., & Rochimah, S. (2025). An evaluation model of website testing framework based on ISO 25010 performance efficiency. *Indonesian Journal of Electrical Engineering and Computer Science*, 37(2), 1130–1139. <https://doi.org/10.11591/ijeecs.v37.i2.pp1130-1139>
- Liu, Y., Qi, H., & Lu, X. (2025). Enhancing GPT-based automated essay scoring: The impact of fine-tuning and linguistic complexity measures. *Computer Assisted Language Learning*. <https://doi.org/10.1080/09588221.2025.2518430>

- Maria Rosario Borroek, et al. (2025). Optimasi software effort estimation menggunakan random forest. *Prosiding Seminar Nasional Ilmu Teknik*, 2(2). <https://doi.org/10.61132/prosemnasproit.v2i2.156>
- Muhammad, I. R. D., & Paputungan, I. V. (2024). Development of backend server based on REST API architecture in e-wallet transfer system. *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, 3(2), 79–87. <https://doi.org/10.20885/snati.v3.i2.35>
- Nurjaman, I., Utomo, F. S., & Hermanto, N. (2024). Penerapan REST API Laravel sebagai fondasi back-end aplikasi G-MOOC 4D. *Journal of Informatics and Interactive Technology (JIITE)*, 1, 9–18. <https://doi.org/10.63547/jiite.v1i1.4>
- Nursahiba, & Enda, D. (2025). Pengukuran kualitas perangkat lunak dengan standar ISO/IEC 25010 pada website Jurusan Teknik Informatika Politeknik Negeri Bengkalis. *JEKIN: Jurnal Teknik Informatika*, 5(1), 13–23. <https://doi.org/10.58794/jekin.v5i1.839>
- Rahayu, S. S., Wahyudin, Firmansyah, D. A., & Susanti, S. (2024). Analisis penggunaan tools automation testing pada aplikasi: Systematic literature review. *REMIK: Riset dan E-Jurnal Manajemen Informatika Komputer*, 8(1). <https://doi.org/10.33395/remik.v8i1.13241>
- Rochimah, S., Yuhana, U. L., & Hidayat, S. B. (2023). Software quality measurement for functional suitability, performance efficiency, and reliability characteristics using analytical hierarchy process. *International Journal on Informatics Visualization*, 7(4), 2421–2426. <https://doi.org/10.62527/joiv.7.4.2441>
- Rojas, H., Renteria, R., Duran, V. M., Gutiérrez, Y. T., Cabrera, M. J. I., & Aminuddin, A. (2025). Mapping the evolution and future directions of ISO/IEC 25010: A bibliometric and thematic analysis. *Engineering, Technology & Applied Science Research*, 15(5), 27530–27541. <https://doi.org/10.48084/etasr.11772>
- Shofiana, D. A., Mardiansyah, S. S., Parabi, M. I., & Syarif, A. (2024). Implementasi REST API menggunakan JSON Web Token (JWT) pada sistem monitoring KPI berbasis mobile di PT Industri Kereta Api (Persero). *Jurnal Komputasi*, 12(2). <https://doi.org/10.23960/komputasi.v12i2.272>
- Swari, D. A. A. I., Suryawan, I. K. D., Sugiartha, I. K., Mahadewi, P. J., & Darmawan, I. M. P. (2023). Analisis kualitas aplikasi Talocraft menggunakan standar ISO/IEC 25010. *TEKNIMEDIA*, 4(2), 228–234. <https://doi.org/10.46764/teknimedia.v4i2.161>