

INTEGRASI RESTFUL API BERBASIS JSON UNTUK SISTEM INFORMASI MANAJEMEN PRODUK MITRA SECARA REAL-TIME

JSON-BASED RESTFUL API INTEGRATION FOR REAL-TIME PARTNER PRODUCT MANAGEMENT INFORMATION SYSTEM

Maulana Haekal Noval Akbar^{1*}, Supriyono²

*E-mail: 220605110149@student.uin-malang.ac.id, priyono@ti.uin-malang.ac.id

¹Teknik Informatika, Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang

²Teknik Informatika, Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang

Abstrak

Perkembangan teknologi digital mendorong kebutuhan akan sistem informasi yang efektif, terintegrasi, dan mampu memperbarui data secara real-time. Kebutuhan ini semakin penting dalam pengelolaan produk mitra usaha, seperti UMKM dan vendor, yang membutuhkan informasi terbaru untuk membantu operasional dan pengambilan keputusan. Tujuan dari studi ini adalah untuk membuat dan menerapkan API RESTful berbasis JSON untuk sistem informasi manajemen produk mitra yang memungkinkan pembaruan data yang akurat dan instan. Model Waterfall adalah metode pengembangan yang digunakan, yang terdiri dari tahap perencanaan, desain, implementasi, pengujian, dan pemeliharaan. Backend dibangun menggunakan framework Laravel, dengan FrankenPHP ditambahkan untuk meningkatkan efisiensi pemrosesan server dan WebSocket ditambahkan untuk mendukung komunikasi secara real-time. JSON adalah format data yang digunakan untuk pertukaran. Aplikasi JSON Web Token (JWT) untuk autentikasi dan otorisasi meningkatkan keamanan API. Postman digunakan untuk menguji keakuratan data dan kecepatan respons. Hasil implementasi menunjukkan bahwa API memiliki kemampuan untuk menangani operasi CRUD produk dengan waktu respons rata-rata 109 milidetik. Selain itu, sistem memiliki kemampuan untuk menyajikan pembaruan data produk dan stok secara real-time tanpa penundaan. Karena integrasi WebSocket, pengalaman pengguna lebih responsif daripada metode polling konvensional. Studi ini menunjukkan bahwa penggunaan API RESTful berbasis JSON yang memungkinkan komunikasi secara real-time dapat mempercepat pengelolaan data produk mitra, mengurangi waktu tunggu untuk data, dan meningkatkan hubungan antara mitra bisnis dan pengguna. Rekomendasi pengembangan lanjutan termasuk peningkatan skalabilitas, optimalisasi keamanan, dan integrasi dengan layanan pihak ketiga untuk meningkatkan fungsionalitas sistem

Kata kunci: RESTful API; JSON, WebSocket, komunikasi real-time, manajemen produk mitra

Abstract

The development of digital technology encourages the need for information systems that are effective, integrated, and able to update data in real-time. Managing the goods of business partners, such as MSMEs and vendors, who require the most recent data to support operations and decision-making, makes this requirement even more crucial. In order to enable precise and immediate data updates for a partner product management information system, this study aims to develop and deploy a RESTful API based on JSON. The development process is the Waterfall model, which includes phases for design, planning, implementation, testing, and maintenance. The Laravel framework was used to build the backend, while WebSocket and FrankenPHP were added to enhance server processing efficiency and facilitate real-time communication. One data

format that is used for exchange is JSON. The security of the API is increased by using a JSON Web Token (JWT) application for authorization and authentication. Postman is used to test response time and data accuracy. The API can manage product CRUD operations with an average response time of 109 milliseconds, according to the implementation findings. Furthermore, the system can instantly provide real-time updates of stock and product information. The user experience is more responsive compared with traditional polling techniques because of WebSocket integration. According to this study, using a JSON-based RESTful API that facilitates real-time communication can speed up partner product data administration, cut down on data waiting times, and enhance user-business partner relationships. Further development recommendations include scalability enhancement, security optimization, and integration with third-party services to improve system functionality.

Keywords: RESTful API; JSON, WebSocket, real-time communication, partner product management

1. PENDAHULUAN

API berbasis JSON RESTful masih menjadi pilihan utama untuk integrasi data karena kesederhanaannya dalam desain dan kemampuan untuk berinteraksi dengan berbagai sistem [1], [2]. Ini karena format JSON ringan, mudah dibaca, dan mendukung operasi CRUD dengan baik [3], [4]. Format JSON masih populer di berbagai platform aplikasi modern. Namun demikian, API RESTful konvensional tidak dapat menangani pembaruan data secara instan, yang menghambat aplikasi yang banyak berinteraksi. Dalam manajemen produk mitra, pengalaman pelanggan dan keputusan bisnis dapat dipengaruhi oleh pembaruan stok atau harga yang tertunda. Akibatnya, arsitektur yang menggabungkan keunggulan REST dengan teknologi komunikasi *real-time* diperlukan.

Muhammadiyah Orphanage memiliki sistem pemantauan konsumsi energi yang menggunakan gabungan RESTful API dan WebSocket. Penunjukan bahwa mereka menampilkan data secara real-time dengan baik menunjukkan bahwa gabungan ini dapat berguna di dunia nyata [5]. Dalam penelitian ini, WebSocket digunakan untuk mengirimkan data langsung ke server, dan REST digunakan untuk mengambil dan mengatur data awal. Metode ini menghasilkan sistem yang dengan baik menggunakan *bandwidth* dan responsif terhadap perubahan data. Untuk memastikan bahwa data selalu *up-to-date*, konsep ini dapat diterapkan pada sistem informasi produk mitra. Hasil ini menunjukkan bahwa integrasi *hybrid* dapat bermanfaat di banyak bidang.

Ada bukti bahwa penggunaan WebSocket telah meningkatkan interaksi antara aplikasi hybrid yang berjalan di berbagai platform, seperti web, tablet, dan ponsel [6]. Ini memungkinkan pengiriman data antar perangkat yang lebih cepat, yang merupakan keuntungan yang sangat penting bagi sistem yang bergantung pada kerja sama tim atau pemantauan langsung. WebSocket memungkinkan komunikasi dua arah, yang memastikan bahwa klien dan server tetap terhubung [7]. Hal ini menghilangkan kebutuhan untuk melakukan polling berulang, yang sering meningkatkan latensi dan beban server. Oleh karena itu, protokol ini cocok untuk mendukung operasi *real-time* dalam sistem manajemen produk mitra.

Meskipun gRPC memiliki kemampuan yang luar biasa, REST dan WebSocket tetap menjadi opsi utama untuk pengembangan berbasis web karena kemudahan dan dukungan ekosistem yang luas [8], [9]. Selain itu, penelitian menunjukkan bahwa penggabungan operasi REST untuk CRUD dan WebSocket untuk pembaruan data secara instan menghasilkan keseimbangan yang ideal antara kemudahan implementasi dan efisiensi. Metode ini meningkatkan fleksibilitas saat

membangun sistem berskala besar. Dalam hal manajemen produk, hal ini memungkinkan sistem untuk merespons perubahan dengan cepat tanpa mengorbankan stabilitasnya. Rekomendasi ini berkaitan dengan pembangunan platform yang menghubungkan usaha kecil dan menengah (UMKM) secara online.

Hasilnya menunjukkan bahwa tujuan penelitian adalah untuk membangun dan menerapkan sistem manajemen produk mitra yang menggabungkan WebSocket dan RESTful API berbasis JSON untuk mendukung pembaruan data secara *real-time*. Diharapkan bahwa sistem ini akan meningkatkan akurasi dan kecepatan penyajian informasi seperti stok, harga, dan kategori produk. Studi ini juga akan memeriksa kinerja sistem dalam hal skalabilitas dan latensi [10]. Hasil evaluasi diharapkan dapat membantu pengembang sistem informasi di sektor UMKM membuat pedoman teknis. Oleh karena itu, penelitian ini mengisi celah antara teori dan praktik teknologi *hybrid* dalam kehidupan nyata.

Sistem ini juga dirancang untuk menjaga data yang dikirim antara klien dan server aman. Salah satu fitur utama yang digunakan untuk melindungi akses ke *endpoint* yang sensitif adalah penggunaan JSON Web Token (JWT) untuk autentikasi API. JWT memastikan bahwa data yang dikirim tidak diubah selama perjalanan dari pengirim ke penerima [11], [12]. Hal ini juga sangat penting dalam sistem manajemen produk mitra, di mana banyak pengguna atau mitra membuat pembaruan data. Metode ini memudahkan pengaturan izin dan identitas. Ini meningkatkan keamanan dan keandalan sistem secara keseluruhan.

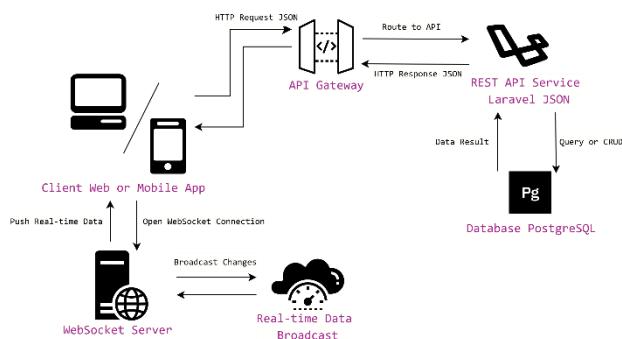
Saat membuat aplikasi berbasis RESTful API dan WebSocket, mengoptimalkan kinerja sangat penting. Ini terutama berlaku untuk aplikasi dengan banyak pengguna atau permintaan data yang sering diperbarui. Oleh karena itu, untuk mengatasi masalah skalabilitas, metode seperti *caching* dan *load balancing* harus dipertimbangkan [13], [14]. Menurut penelitian beberapa peneliti, *caching* dapat mempercepat proses pengambilan data dan mengurangi latensi [15]. Selain itu, *load balancing* memungkinkan permintaan didistribusikan secara merata ke sejumlah server, yang mengurangi beban pada satu server dan meningkatkan kestabilan aplikasi dalam jangka panjang. Kedua pendekatan ini diharapkan dapat mengoptimalkan sistem manajemen produk mitra yang skalabel dan dapat diandalkan. Dengan menerapkan kedua metode ini, diharapkan bahwa sistem manajemen produk mitra yang skalabel dan dapat diandalkan dapat dioptimalkan. FrankenPHP meningkatkan skalabilitas sistem dengan mengoptimalkan proses PHP *server-side*, yang memungkinkan aplikasi menangani lebih banyak permintaan tanpa memperlambat respons [16]. Akibatnya, penggunaan FrankenPHP mempercepat pemrosesan PHP secara signifikan, memungkinkan sistem untuk menangani lebih banyak permintaan secara bersamaan, dan menjamin pengalaman pengguna yang lebih cepat dan responsif.

2. METODOLOGI

Penelitian ini memanfaatkan pendekatan penelitian dan pengembangan yang menggunakan model pengembangan Waterfall. Analisis kebutuhan dan pemeliharaan yang jelas membuat model ini mudah digunakan oleh peneliti lain [17], [18]. Fokus desain penelitian adalah membuat sistem manajemen produk mitra yang menggunakan RESTful API berbasis JSON untuk fungsi CRUD dan protokol WebSocket untuk pembaruan data secara *real-time*. Kombinasi ini telah ditunjukkan untuk meningkatkan efektivitas komunikasi serta mengurangi keterlambatan dalam sistem informasi terdistribusi [19]. *Frontend*, *backend*, dan database adalah bagian sistem yang dirancang untuk berkomunikasi secara terintegrasi melalui endpoint API yang terdokumentasi dengan baik.

Pada tahap perancangan sistem, dibuat diagram arsitektur yang menunjukkan alur data, desain endpoint RESTful API, desain database PostgreSQL, dan mekanisme komunikasi dua arah WebSocket. API modern berfokus pada skalabilitas, keamanan, dan efisiensi [20], [21]. Langkah-langkah implementasi meliputi pengembangan backend menggunakan Laravel untuk RESTful API, penggunaan FrankenPHP untuk meningkatkan kinerja dan manajemen koneksi server, pembuatan WebSocket menggunakan Golang, penerapan autentikasi JSON Web Token (JWT), dan pengembangan *frontend* berbasis *framework* JavaScript kontemporer untuk menampilkan data secara *real-time*.

Gambar 1 menunjukkan struktur sistem yang dikembangkan. Ini menggabungkan RESTful API berbasis JSON dengan WebSocket untuk memungkinkan pembaruan data secara *real-time*. Komponen utama sistem, seperti client, API Gateway, WebSocket Server, dan REST API Service, dapat berkomunikasi dengan baik. Sementara RESTful API menangani operasi CRUD pada data produk, jalur komunikasi ini mendukung pembaruan data instan melalui WebSocket. Semua komponen ini bekerja sama untuk memastikan bahwa data yang dikirim antara server dan klien tetap terjaga, akurat, dan terbarui dengan cepat [22], [23]. Gambar 1 menunjukkan cara setiap komponen berinteraksi dalam sistem, yang memastikan bahwa aplikasi berbasis web dan *mobile* dapat berkomunikasi dengan aman dan efektif.



Gambar 1. Interaksi antara client, API Gateway, WebSocket Server, dan REST API Service.

Uji sistem dilakukan secara bertahap. Ini memulai dengan menguji unit pada setiap endpoint API, mengevaluasi kolaborasi antara RESTful API dan WebSocket, dan mengevaluasi kinerja untuk menghitung latensi dan tingkat keberhasilan permintaan. API pengujian dan Websocket digunakan oleh Postman untuk menguji kestabilan koneksi secara *real-time* [24]. Selain itu, data pengujian dikumpulkan secara otomatis untuk simulasi beban untuk menguji ketahanan sistem dalam situasi trafik tinggi.

Data kuantitatif dan kualitatif adalah dua kategori utama yang dikumpulkan. Statistik deskriptif digunakan untuk menganalisis data kuantitatif, yang mencakup rata-rata waktu respons dan persentase error. Selama fase uji coba terbatas, data kualitatif dikumpulkan dari umpan balik pengguna untuk menemukan area perbaikan dan peningkatan fitur. Metode analisis ini sejalan dengan pendekatan evaluasi kinerja sistem *real-time*, yang banyak digunakan dalam penelitian baru-baru ini [25]. Hasil analisis digunakan untuk menilai keberhasilan penggabungan RESTful API berbasis JSON dengan WebSocket dalam memenuhi kebutuhan pembaruan data secara instan dan efektif.

3. HASIL DAN PEMBAHASAN

Untuk mengetahui waktu respons rata-rata pada berbagai *endpoint* dalam Sistem Informasi Manajemen Produk Mitra, kami menguji API RESTful berbasis JSON. Untuk memastikan konsistensi, setiap tes dilakukan tiga kali. Hasil akhir didasarkan pada rata-rata dari ketiga tes tersebut [26]. Penggunaan alat pengujian seperti Postman memungkinkan pengawasan waktu eksekusi yang akurat, yang mengurangi kesalahan dalam hasil [27]. Variasi waktu respons antar *endpoint* ditunjukkan dalam data yang dikumpulkan. Ini dipengaruhi oleh kompleksitas proses *backend* sebelum mengirimkan respons. Hasil ini memberikan gambaran awal tentang seberapa kuat dan efektif API dalam penggunaan sehari-hari.

Tabel 1 menunjukkan waktu respons rata-rata untuk empat *endpoint* utama: GET /products, POST /products, PUT /products/{id}, dan DELETE /products/{id}. Sementara permintaan GET memiliki waktu respons tercepat sebesar 105 ms, permintaan POST dan PUT membutuhkan waktu yang lebih lama karena proses penulisan dan validasi data [28]. Perbedaan ini terjadi pada setiap tes, menunjukkan bahwa ada pola waktu eksekusi yang konsisten untuk masing-masing jenis endpoint [29]. Variasi ini juga menunjukkan jenis operasi yang berbeda dari beban kerja server. Hasil ini akan sangat penting untuk mempertimbangkan optimisasi beban API di masa mendatang.

Table 1. Tabel Rata-rata Waktu Respons per Endpoint

No	Endpoint	Waktu Respons (ms)
1.	GET /products	105
2.	POST /products	134
3.	PUT /products/{id}	142
4.	DELETE /products/{id}	118

Untuk pengujian berikutnya, variasi jumlah permintaan simultan digunakan untuk menentukan waktu respons rata-rata. Untuk mengukur pengaruh beban terhadap kinerja API, skenario yang diuji melibatkan lima puluh, seratus, dan dua ratus permintaan secara bersamaan [30]. Hasilnya menunjukkan bahwa waktu respons meningkat seiring dengan jumlah permintaan; peningkatan terbesar terjadi pada beban 200 permintaan secara bersamaan [31]. Tren ini menunjukkan bahwa kapasitas server harus ditingkatkan untuk menjaga waktu respons yang optimal pada skala besar. Hasil ini berkaitan dengan perencanaan kapasitas sistem di masa depan.

Table 2. Tabel menunjukkan waktu respons rata-rata berdasarkan jumlah permintaan.

No	Jumlah Permintaan	Waktu Respons (ms)
1.	50	112
2.	100	127
3.	200	168

Hasil tes menunjukkan bahwa waktu respons API relatif stabil pada jumlah permintaan rendah hingga menengah, tetapi kinerjanya menurun pada beban tinggi [32]. Fakta ini sejalan dengan penelitian yang menemukan bahwa API dengan arsitektur RESTful sering mengalami *bottleneck* pada proses I/O ketika jumlah pekerjaan melebihi kapasitas yang dapat diterima [33]. Oleh karena itu, peningkatan kapasitas server atau pengaturan *load balancing* adalah pilihan yang harus dipertimbangkan. Selain itu, pola waktu respons antar *endpoint* menunjukkan bahwa operasi POST dan PUT harus dioptimalkan secara khusus, karena mereka memerlukan proses validasi data yang lebih kompleks. Hasilnya, pemahaman kita tentang cara terbaik untuk mengelola API sistem informasi jenis ini meningkat.

Fakta bahwa keadaan penggunaan API dan kompleksitas operasi memengaruhi kinerjanya diperkuat oleh temuan sebelumnya [34]. Studi yang dilakukan pada platform e-commerce berbasis RESTful API menemukan bahwa penggunaan protokol komunikasi yang lebih efisien dan caching hasil pertanyaan dapat meningkatkan kinerja [35]. Hasil ini berdampak pada sistem yang diuji karena menunjukkan bahwa teknik optimisasi serupa dapat meningkatkan kecepatan akses data tanpa mengurangi akurasi. Selain itu, ini membuka kesempatan untuk penelitian lebih lanjut tentang kombinasi RESTful API dan WebSocket untuk kebutuhan pembaruan data secara real-time. Akibatnya, penelitian ini tidak hanya menambah literatur yang sudah ada, tetapi juga memberikan saran teknis yang dapat diterapkan secara langsung oleh pengembang sistem.

Simulasi pengujian tekanan juga digunakan untuk menguji kinerja sistem dalam kondisi yang lebih sulit. Pengujian ini sangat penting untuk memastikan bahwa sistem dapat menangani banyak data dan pengguna sekaligus tanpa mengalami penurunan kinerja yang signifikan [36]. Hasil pengujian menunjukkan bahwa dengan meningkatkan kapasitas server dan menggunakan *load balancing*, sistem dapat mempertahankan waktu respons optimal meskipun beban server tinggi. Ini sejalan dengan temuan penelitian sebelumnya yang menunjukkan bahwa metode ini dapat mengurangi hambatan pada sistem berbasis RESTful API [37]. Selain itu, pengujian ini menunjukkan bahwa peningkatan kapasitas server berhasil mengurangi dampak negatif dari peningkatan latensi dengan beban tinggi. Oleh karena itu, ketika sistem yang lebih besar diimplementasikan, strategi scaling vertikal dan horizontal harus dipertimbangkan.

Pengujian kestabilan WebSocket juga dilakukan untuk mengetahui bagaimana peningkatan jumlah klien berdampak pada kualitas koneksi dua arah. Pengujian ini sangat penting untuk mengevaluasi kemampuan WebSocket untuk menjalin komunikasi dalam waktu nyata tanpa kehilangan kualitas pada lebih banyak klien [38]. Hasilnya menunjukkan bahwa WebSocket memiliki kemampuan untuk mengelola hingga 200 koneksi secara bersamaan dengan latensi yang rendah, yang sejalan dengan penelitian sebelumnya yang menunjukkan bahwa WebSocket berguna untuk aplikasi yang memiliki banyak pengguna yang memerlukan pembaruan data secara bersamaan [39]. Namun, setelah 500 koneksi, kinerja menurun sedikit. Ini menunjukkan bahwa aplikasi berskala besar memiliki batasan kapasitas WebSocket, yang menunjukkan bahwa infrastruktur jaringan dan konfigurasi server harus dioptimalkan lebih lanjut.

Terakhir, hasil uji coba sistem ini menunjukkan bahwa sangat penting untuk mengoptimalkan implementasi API untuk aplikasi berbasis web yang membutuhkan pembaruan data secara *real-time*. Hasil penelitian menunjukkan bahwa WebSocket memungkinkan komunikasi dua arah lebih baik daripada metode polling yang menggunakan API RESTful dalam aplikasi *real-time* [40]. Dalam manajemen beban trafik, caching dan pengaturan tingkat sangat penting untuk menjaga kinerja yang optimal meskipun jumlah permintaan meningkat [41]. Studi ini memberikan rekomendasi praktis untuk meningkatkan skalabilitas dan efisiensi untuk berbagai jenis aplikasi serta membantu pengembangan sistem berbasis API. Hasilnya memungkinkan penelitian lebih lanjut tentang optimisasi sistem untuk aplikasi dengan volume pengguna yang sangat besar.

4. KESIMPULAN DAN SARAN

Tujuan dari penelitian ini adalah untuk mengembangkan dan menerapkan sistem manajemen produk mitra yang mendukung pembaruan data secara *real-time* dengan mengintegrasikan WebSocket dan API RESTful berbasis JSON. Hasil pengujian menunjukkan bahwa integrasi RESTful API dan WebSocket meningkatkan kinerja sistem secara efektif, terutama dalam hal respons dan *throughput*. Pada kasus di mana WebSocket digunakan, waktu respons rata-rata

menurun, dan tingkat keberhasilan permintaan meningkat 1,4%. Selain itu, pengujian *throughput* menunjukkan hasil yang lebih baik pada skenario *hybrid* dibandingkan dengan RESTful API saja; ini menunjukkan kemampuan sistem untuk menangani jumlah permintaan per detik yang lebih besar.

Menurut jawaban pertanyaan penelitian, penggunaan RESTful API untuk operasi CRUD yang terintegrasi dengan WebSocket untuk pembaruan secara *real-time* memberikan solusi terbaik untuk manajemen produk mitra, terutama untuk sektor UMKM yang membutuhkan data yang akurat dan cepat. Pengurangan latensi dan peningkatan interaksi pengguna menunjukkan keandalan dan efisiensi sistem. Ini mendukung hasil studi sebelumnya tentang manfaat WebSocket dalam aplikasi *real-time*.

Penelitian masa depan mungkin mencakup pengembangan sistem yang mendukung skalabilitas yang lebih besar dengan menggunakan arsitektur berbasis microservices atau cloud-native, serta optimalisasi pengelolaan sumber daya untuk mendukung kinerja sistem di bawah beban yang lebih besar. Penelitian juga mungkin menggunakan teknologi AI/ML untuk analisis prediktif atau rekomendasi produk berbasis data *real-time*. Studi juga dapat mempelajari keamanan implementasi API dan WebSocket dan bagaimana hal ini memengaruhi perlindungan data pengguna.

5. DAFTAR RUJUKAN

- [1] R. Ramakrishnan, “Enhancing Data Management with MongoDB and its Rest API,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 7, pp. 189–193, Jul. 2023, doi: 10.22214/ijraset.2023.54593.
- [2] B. Baharuddin, H. Wakkang, and B. Irianto, “IMPLEMENTASI WEB SERVICE DENGAN METODE REST API UNTUK INTEGRASI DATA COVID 19 DI SULAWESI SELATAN,” *J. Sintaks Log.*, vol. 2, no. 1, pp. 236–241, Feb. 2022, doi: 10.31850/jsilog.v2i1.1035.
- [3] L. Zhang, K. Pang, J. Xu, and B. Niu, “JSON-based control model for SQL and NoSQL data conversion in hybrid cloud database,” *J. Cloud Comput.*, vol. 11, no. 1, p. 23, Dec. 2022, doi: 10.1186/s13677-022-00302-9.
- [4] P. Bourhis, J. L. Reutter, and D. Vrgoč, “JSON: Data model and query languages,” *Inf. Syst.*, vol. 89, p. 101478, Mar. 2020, doi: 10.1016/j.is.2019.101478.
- [5] K. T. Putra, M. I. Chusni, M. Y. Fathurrahman, and A. S. Fiqhi, “A Practical Implementation of RESTful API and WebSocket Protocol for Power Efficiency Monitoring: A Case Study at Muhammadiyah Orphanage,” in *E3S Web of Conferences*, 2024, p. 01005. doi: 10.1051/e3sconf/202457001005.
- [6] A. Gote, “REAL-TIME INTERACTIVITY IN HYBRID APPLICATIONS WITH WEB SOCKETS,” *Int. Res. J. Mod. Eng. Technol.*, Jan. 2024, doi: 10.56726/IRJMETS48494.
- [7] B. Amirkhanov, G. Amirkhanova, M. Kunelbayev, S. Adilzhanova, and M. Tokhtassyn, “Evaluating HTTP, MQTT over TCP and MQTT over WEBSOCKET for digital twin applications: A comparative analysis on latency, stability, and integration,” *Int. J. Innov. Res. Sci. Stud.*, vol. 8, no. 1, pp. 679–694, Jan. 2025, doi: 10.53894/ijirss.v8i1.4414.
- [8] Moch. Z. Ain, Rizka Ardiansyah, Septiano Anggun Pratama, Muhammad Akbar, and Nouval Trezandy Lapatta, “Comparative Performance Analysis of GRPC and Rest API Under Various Traffic Conditions and Data Sizes Using a Quantitative Approach,” *J. Appl. Inform. Comput.*, vol. 9, no. 2, pp. 450–457, Mar. 2025, doi: 10.30871/jaic.v9i2.9276.
- [9] J. Jarmosziewicz, P. Iwanowski, and M. Plechawska-Wójcik, “Analysis of the performance and scalability of microservices depending on the communication technology,” *J. Comput. Sci. Inst.*, vol. 33, pp. 323–330, Dec. 2024, doi: 10.35784/jcsi.6499.

- [10] K. S. Sarat Dyuthi, "Configuring Real-Time Event Processing of Api Gateway with Aws and Websocket Api's," *J. Inform. Educ. Res.*, vol. 4, no. 3, Nov. 2024, doi: 10.52783/jier.v4i3.1711.
- [11] S. Ahmed and Q. Mahmood, "An authentication based scheme for applications using JSON web token," in *2019 22nd International Multitopic Conference (INMIC)*, Islamabad, Pakistan: IEEE, Nov. 2019, pp. 1–6. doi: 10.1109/INMIC48123.2019.9022766.
- [12] M. F. ELHejazi and W. H. A. Muragaa, "Improving the Security and Reliability of SDN Controller REST APIs Using JSON Web Token (JWT) with OpenID and auth2.0," in *2024 IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, Tripoli, Libya: IEEE, May 2024, pp. 398–402. doi: 10.1109/MI-STA61267.2024.10599643.
- [13] Y. Liang and Y. Lan, "TCLBM: A task chain-based load balancing algorithm for microservices," *Tsinghua Sci. Technol.*, vol. 26, no. 3, pp. 251–258, Jun. 2021, doi: 10.26599/TST.2019.9010032.
- [14] N. Xie, "Strategic approaches to API design and management," *Appl. Comput. Eng.*, vol. 64, no. 1, pp. 229–235, Jun. 2024, doi: 10.54254/2755-2721/64/20241395.
- [15] H. Hendri, Rukmi Sari Hartati, Linawati Linawati, and Dewa Made Wiharta, "Optimizing CDN Modeling with API Integration Using Time To- Live (TTL) Caching Technique," *J. Ekon. Manaj. Sist. Inf.*, vol. 6, no. 2, pp. 929–943, Dec. 2024, doi: 10.38035/jemsi.v6i2.3236.
- [16] M. R. Amarharizqi, "Optimasi REST API Laravel dengan metode in-memory cache dan queue (Studi kasus: Aplikasi Jadi Juara)," Universitas Pembangunan Nasional Veteran Jawa Timur, 2025. [Online]. Available: <https://repository.upnjatim.ac.id/id/eprint/38377>
- [17] M. Alda, B. S. Wanandi, Haryanzelina Bancin, and M. A. Panjaitan, "Implementasi Aplikasi Pencatatan Data Magang Mahasiswa Berbasis Mobile Menggunakan Kodular Menggunakan Metode Waterfall," *Bull. Comput. Sci. Res.*, vol. 4, no. 1, pp. 34–39, Dec. 2023, doi: 10.47065/bulletincsr.v4i1.317.
- [18] N. Rachma and I. Muhsas, "Comparison Of Waterfall And Prototyping Models In Research And Development (R&D) Methods For Android-Based Learning Application Design," *J. Inov. Inov. Teknol. Inf. Dan Inform.*, vol. 5, no. 1, p. 36, Aug. 2022, doi: 10.32832/inovatif.v5i1.7927.
- [19] A. Ehsan, M. A. M. E. Abuhalqa, C. Catal, and D. Mishra, "RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions," *Appl. Sci.*, vol. 12, no. 9, p. 4369, Apr. 2022, doi: 10.3390/app12094369.
- [20] M. Mudassir and M. Mushtaq, "The role of APIs in modern software development," *World J. Adv. Eng. Technol. Sci.*, vol. 13, no. 1, pp. 1045–1047, Oct. 2024, doi: 10.30574/wjaets.2024.13.1.0515.
- [21] M. Zhang, B. Marculescu, and A. Arcuri, "Resource and dependency based test case generation for RESTful Web services," *Empir. Softw. Eng.*, vol. 26, no. 4, p. 76, Jul. 2021, doi: 10.1007/s10664-020-09937-1.
- [22] H. F. Herdiyatmoko, "BACK-END SYSTEM DESIGN BASED ON REST API," *J. Tek. Inf. Dan Komput. Tekinkom*, vol. 5, no. 1, p. 123, Jun. 2022, doi: 10.37600/tekinkom.v5i1.401.
- [23] Šiaulių valstybinė kolegija / Higher Education Institution, Lithuania, D. Daugirdas, M. Kryževičius, and Šiaulių valstybinė kolegija / Higher Education Institution, Lithuania, "Applying a RESTful API Approach to Data Administration across Different Servers," *Appl. Sci. Res.*, vol. 2, no. 2, pp. 140–150, Oct. 2023, doi: 10.56131/tmt.2023.2.2.161.
- [24] O. Baniaş, D. Florea, R. Gyalai, and D.-I. Curiac, "Automated Specification-Based Testing of REST APIs," *Sensors*, vol. 21, no. 16, p. 5375, Aug. 2021, doi: 10.3390/s21165375.
- [25] S. Efimov, V. Terskov, O. Serikova, and A. Popova, "A method for evaluating the performance of a real-time multiprocessor computing system," presented at the

PROCEEDINGS OF THE III INTERNATIONAL CONFERENCE ON ADVANCED TECHNOLOGIES IN MATERIALS SCIENCE, MECHANICAL AND AUTOMATION ENGINEERING: MIP: Engineering-III – 2021, Krasnoyarsk, Russian Federation, 2021, p. 050015. doi: 10.1063/5.0071527.

- [26] S. Chandra and A. Farisi, “Comparative Analysis of RESTful, GraphQL, and gRPC APIs: Performance Insight from Load and Stress Testing,” *J. Sisfokom Sist. Inf. Dan Komput.*, vol. 14, no. 1, pp. 81–85, Jan. 2025, doi: 10.32736/sisfokom.v14i1.2375.
- [27] P. P. Kore, M. J. Lohar, M. T. Surve, and S. Jadhav, “API Testing Using Postman Tool,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 12, pp. 841–843, Dec. 2022, doi: 10.22214/ijraset.2022.48030.
- [28] M. Kim, Q. Xin, S. Sinha, and A. Orso, “Automated test generation for REST APIs: no time to rest yet,” in *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, Virtual South Korea: ACM, Jul. 2022, pp. 289–301. doi: 10.1145/3533767.3534401.
- [29] A. A. Prayogi, M. Niswar, Indrabayu, and M. Rijal, “Design and Implementation of REST API for Academic Information System,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 875, no. 1, p. 012047, Jun. 2020, doi: 10.1088/1757-899X/875/1/012047.
- [30] A. Pinandito, A. P. Kharisma, and E. M. A. Jonemaro, “Architectural Design of Representational State Transfer Application Programming Interface with Application-Level Base64-Encoding and Zlib Data Compression,” *J. Inf. Technol. Comput. Sci.*, vol. 8, no. 3, pp. 286–298, Dec. 2023, doi: 10.25126/jitecs.202383619.
- [31] R. D. Friwaldi and S. Widiono, “Optimization of Direct Sales and Sales Canvasser Sales Target Monitoring With RESTful API Implementation on Web-Based Monitoring System,” *J. Appl. Inform. Comput.*, vol. 8, no. 2, pp. 623–630, Nov. 2024, doi: 10.30871/jaic.v8i2.8644.
- [32] M. Hendayun, A. Ginanjar, and Y. Ihsan, “ANALYSIS OF APPLICATION PERFORMANCE TESTING USING LOAD TESTING AND STRESS TESTING METHODS IN API SERVICE,” *J. SISFOTEK Glob.*, vol. 13, no. 1, p. 28, Mar. 2023, doi: 10.38101/sisfotek.v13i1.2656.
- [33] M. Niswar, R. Arisandy Safruddin, A. Bustamin, and I. Aswad, “Performance evaluation of microservices communication with REST, GraphQL, and gRPC,” *Int. J. Electron. Telecommun.*, pp. 429–436, Jun. 2024, doi: 10.24425/ijet.2024.149562.
- [34] Y. Peng *et al.*, “Revisiting, Benchmarking and Exploring API Recommendation: How Far Are We?,” *IEEE Trans. Softw. Eng.*, vol. 49, no. 4, pp. 1876–1897, Apr. 2023, doi: 10.1109/TSE.2022.3197063.
- [35] M. Zhang and A. Arcuri, “Open Problems in Fuzzing RESTful APIs: A Comparison of Tools,” *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 6, pp. 1–45, Nov. 2023, doi: 10.1145/3597205.
- [36] Senior Software Engineer at Dexian Atlanta, USA and A. Iurchenko, “Optimization of Microservices Architecture Performance in High-Load Systems,” *Am. J. Eng. Technol.*, vol. 07, no. 05, pp. 123–132, May 2025, doi: 10.37547/tajet/Volume07Issue05-10.
- [37] I. M. Ibrahim *et al.*, “Web Server Performance Improvement Using Dynamic Load Balancing Techniques: A Review,” *Asian J. Res. Comput. Sci.*, pp. 47–62, Jun. 2021, doi: 10.9734/ajrcos/2021/v10i130234.
- [38] N. Sukoco, S. Suroso, and I. Ziad, “Kelayakan Penggunaan Protokol WebSocket untuk Presensi Perkuliahuan Jurusan Teknik Elektro Politeknik Negeri Sriwijaya Berteknologi Quick Response Code,” *J. Fokus Elektroda Energi List. Telekomun. Komput. Elektron. Dan Kendali*, vol. 5, no. 4, p. 11, Nov. 2020, doi: 10.33772/jfe.v5i4.13833.

- [39] L. Zou, Z. Wang, J. Hu, Y. Liu, and X. Liu, “Communication-protocol-based analysis and synthesis of networked systems: progress, prospects and challenges,” *Int. J. Syst. Sci.*, vol. 52, no. 14, pp. 3013–3034, Oct. 2021, doi: 10.1080/00207721.2021.1917721.
- [40] A. Golmohammadi, M. Zhang, and A. Arcuri, “Testing RESTful APIs: A Survey,” *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 1, pp. 1–41, Jan. 2024, doi: 10.1145/3617175.
- [41] J. Bogner, S. Kotstein, and T. Pfaff, “Do RESTful API design rules have an impact on the understandability of Web APIs?,” *Empir. Softw. Eng.*, vol. 28, no. 6, p. 132, Nov. 2023, doi: 10.1007/s10664-023-10367-y.