

IMPLEMENTASI ARSITEKTUR MICROSERVICE PADA APLIKASI BANK SAMPAH DIGITAL BERBASIS DONATJS

Wawan Sismadi¹, Besar Agung Martono², Estuti Fitri Hartini³, Dika Wahyudanti Putri⁴

Universitas IPWIJA^{1,2,3}, Universitas Budi Luhur⁴

e-mail: wawan@sismadi.com¹, agungmartono@gmail.com², estufitrihartini@gmail.com³,
dikawputri@gmail.com⁴

ABSTRAK

Pengelolaan sampah digital merupakan salah satu inovasi penting dalam mendukung pembangunan berkelanjutan dan ekonomi sirkular. Namun, sebagian besar aplikasi bank sampah digital masih menghadapi kendala serius pada aspek skalabilitas, kinerja, dan maintainability ketika jumlah pengguna meningkat. Penelitian ini bertujuan merancang dan mengimplementasikan aplikasi bank sampah digital berbasis microservice architecture dengan menggunakan DonatJS framework. Metode penelitian menggunakan Extreme Programming (XP) melalui tahapan planning, design, coding, dan testing. Evaluasi performa dilakukan menggunakan enam metrik Core Web Vitals, yaitu FCP, LCP, SI, TBT, CLS, dan Responsiveness. Hasil penelitian menunjukkan bahwa DonatJS mampu menghasilkan waktu rendering konten lebih cepat (FCP dan LCP), menurunkan Speed Index, serta meningkatkan stabilitas layout (CLS) dibandingkan framework MVC konvensional seperti Laravel dan CodeIgniter. Pembahasan menunjukkan bahwa integrasi microservice dengan DonatJS mendukung fleksibilitas pengembangan, efisiensi pemrosesan transaksi, dan keberlanjutan layanan digital komunitas. Dengan demikian, penelitian ini berkontribusi pada literatur dengan menyediakan model implementasi yang dapat direplikasi pada sistem berbasis teknologi ramah lingkungan.

Kata Kunci: *bank sampah digital, microservice architecture, DonatJS, skalabilitas, web application*

ABSTRACT

Digital waste management has emerged as an important innovation in supporting sustainable development and the circular economy. However, most digital waste bank applications still face critical challenges in terms of scalability, performance, and maintainability as the number of users increases. This study aims to design and implement a digital waste bank application based on microservice architecture using the DonatJS framework. The research method applies Extreme Programming (XP) through four stages: planning, design, coding, and testing. System performance was evaluated using six Core Web Vitals metrics: First Contentful Paint (FCP), Largest Contentful Paint (LCP), Speed Index (SI), Total Blocking Time (TBT), Cumulative Layout Shift (CLS), and Responsiveness. The results indicate that DonatJS achieves faster content rendering times (FCP and LCP), lowers the Speed Index, and improves layout stability (CLS) compared to conventional MVC frameworks such as Laravel and CodeIgniter. The discussion shows that integrating microservice architecture with DonatJS enhances development flexibility, transaction processing efficiency, and the sustainability of community-based digital services. Thus, this research contributes to the literature by providing an implementable model that can be replicated in environmentally friendly technology-based systems.

Keywords: *digital waste bank, microservice architecture, DonatJS, scalability, web application*

PENDAHULUAN

Dalam lanskap pembangunan berkelanjutan dan ekonomi sirkular, inovasi digital memegang peranan yang semakin krusial. Salah satu manifestasi dari inovasi ini adalah pengembangan aplikasi bank sampah digital, sebuah solusi teknologi yang dirancang untuk merevolusi cara masyarakat berinteraksi dengan pengelolaan sampah (Fakhrurrazi et al., 2022; Purnami, 2021; Pusmiati et al., 2025). Secara ideal, aplikasi ini berfungsi sebagai jembatan yang menghubungkan partisipasi masyarakat dengan sistem daur ulang, memberikan insentif ekonomi, serta menyediakan data yang berharga untuk kebijakan lingkungan. Aplikasi yang ideal harus mampu melayani ribuan pengguna secara serentak, memproses transaksi secara cepat, dan menyediakan antarmuka yang intuitif untuk mendorong adopsi yang luas. Kehadirannya diharapkan tidak hanya meningkatkan angka daur ulang, tetapi juga menumbuhkan kesadaran dan budaya peduli lingkungan di tengah masyarakat (Kumar et al., 2025; Rahman & Shien, 2015).

Namun, realitas di lapangan menunjukkan adanya kesenjangan yang signifikan antara visi ideal tersebut dengan performa teknis dari banyak aplikasi bank sampah digital yang ada. Sebagian besar aplikasi ini pada awalnya dibangun menggunakan arsitektur monolitik, di mana seluruh komponen perangkat lunak digabungkan menjadi satu kesatuan yang erat. Meskipun pendekatan ini memadai pada tahap awal pengembangan, ia mulai menunjukkan kelemahan fatal seiring dengan bertambahnya jumlah pengguna dan volume transaksi. Kesenjangan ini termanifestasi dalam bentuk penurunan kinerja yang drastis, waktu respons yang lambat, kesulitan dalam pemeliharaan, dan kegagalan sistem saat menghadapi beban puncak (Dincă et al., 2022; Heimbach et al., 2023). Akibatnya, pengalaman pengguna menjadi buruk, yang pada akhirnya dapat menurunkan tingkat partisipasi dan menghambat keberhasilan program bank sampah itu sendiri.

Akar dari permasalahan teknis ini terletak pada keterbatasan inheren dari arsitektur monolitik itu sendiri. Dalam sebuah sistem monolitik, setiap komponen saling bergantung satu sama lain, sehingga pembaruan pada satu fitur kecil sekalipun berisiko memengaruhi keseluruhan sistem (Rovanda et al., 2024; Suwito & Manao, 2024). Lebih penting lagi, arsitektur ini tidak memiliki *skalabilitas* yang fleksibel; ketika satu bagian dari aplikasi (misalnya, layanan transaksi) mengalami lonjakan trafik, seluruh aplikasi harus ditingkatkan skalanya, sebuah proses yang tidak efisien dan memakan banyak sumber daya (Lenarduzzi et al., 2020). Kurangnya *maintainability* dan fleksibilitas ini membuat aplikasi sulit untuk beradaptasi dengan kebutuhan baru, menghambat inovasi, dan secara bertahap menjadi usang secara teknis.

Sebagai jawaban atas tantangan tersebut, paradigma arsitektur *microservice* hadir sebagai sebuah solusi yang lebih modern dan tangguh. *Microservice architecture* adalah sebuah pendekatan di mana sebuah aplikasi besar dipecah menjadi serangkaian layanan kecil yang independen dan saling terhubung melalui *API* (Blinowski et al., 2022). Setiap layanan memiliki fungsi spesifiknya sendiri—misalnya, satu layanan untuk otentikasi pengguna, satu untuk transaksi, dan satu lagi untuk pelaporan. Keunggulan utama dari pendekatan ini adalah modularitasnya yang tinggi. Setiap layanan dapat dikembangkan, diperbarui, dan ditingkatkan skalanya secara mandiri tanpa memengaruhi layanan lainnya (Sendika & Frinaldi, 2025; Septiani et al., 2022). Hal ini secara dramatis meningkatkan *skalabilitas*, ketahanan terhadap kegagalan, dan kemudahan dalam pemeliharaan sistem secara keseluruhan.

Untuk mengimplementasikan arsitektur *microservice* secara efektif, diperlukan pemilihan kerangka kerja (*framework*) pengembangan yang tepat. Penelitian ini mengajukan penggunaan *DonatJS* sebagai *framework* utama. *DonatJS* adalah sebuah *framework* yang ringan, modern, dan dirancang untuk mendukung pengembangan aplikasi yang modular dan

berperforma tinggi. Keunggulan *DonatJS* terletak pada kemampuannya untuk merender konten dengan cepat dan menjaga interaktivitas aplikasi tetap tinggi, yang dapat diukur secara objektif melalui metrik *Core Web Vitals* dari Google. Kombinasi antara fleksibilitas arsitektur *microservice* dengan efisiensi yang ditawarkan oleh *DonatJS* menjadi inovasi inti yang diusulkan dalam penelitian ini untuk membangun aplikasi bank sampah digital generasi baru.

Meskipun beberapa penelitian sebelumnya telah menyoroti pentingnya kebijakan inovasi bank sampah (Fatmawati et al., 2022) dan kebutuhan akan digitalisasi dalam pengelolaan limbah (Brasika et al., 2023), masih terdapat celah dalam literatur yang menghubungkan secara langsung antara aspek teknis arsitektur perangkat lunak dengan keberhasilan program lingkungan. Nilai kebaruan dari penelitian ini terletak pada sintesisnya yang unik: ia tidak hanya membahas pentingnya bank sampah digital, tetapi juga menyediakan dan menguji sebuah cetak biru teknis yang konkret untuk membangunnya. Dengan secara spesifik mengimplementasikan arsitektur *microservice* menggunakan *framework DonatJS* dan mengevaluasi kinerjanya, penelitian ini mengisi kesenjangan antara wacana kebijakan lingkungan dengan praktik rekayasa perangkat lunak yang aplikatif.

Berdasarkan latar belakang masalah mengenai tantangan skalabilitas pada aplikasi bank sampah digital, adanya kesenjangan antara arsitektur monolitik dengan tuntutan performa, serta potensi arsitektur *microservice* yang dipadukan dengan *framework DonatJS* sebagai solusi, maka tujuan dari penelitian ini menjadi sangat jelas. Studi ini bertujuan untuk merancang, mengimplementasikan, dan mengevaluasi performa sebuah aplikasi bank sampah digital yang dibangun di atas arsitektur *microservice* dan *framework DonatJS*. Hasil dari penelitian ini diharapkan dapat memberikan bukti empiris mengenai keunggulan pendekatan ini serta menghasilkan sebuah model implementasi yang dapat direplikasi, sehingga dapat memberikan kontribusi nyata bagi pengembangan solusi teknologi yang andal dan berkelanjutan untuk mendukung gerakan ekonomi sirkular.

METODE PENELITIAN

Penelitian ini dirancang dengan menggunakan desain eksperimen terapan yang menerapkan metodologi pengembangan perangkat lunak *Extreme Programming* (XP). Pendekatan ini dipilih karena sifatnya yang iteratif dan fleksibel, yang memungkinkan pengembangan aplikasi secara bertahap melalui empat fase utama: perencanaan, desain, pengkodean, dan pengujian. Pada tahap perencanaan, dilakukan identifikasi kebutuhan pengguna dari berbagai peran, seperti nasabah, pengelola, dan operator bank sampah. Tahap desain berfokus pada perancangan sistem dengan menggunakan *CRC Card* untuk identifikasi objek dan menerapkan arsitektur *microservice*. Arsitektur ini secara spesifik memisahkan layanan-layanan inti seperti autentikasi, transaksi, dan pelaporan ke dalam modul-modul yang independen. Basis data yang digunakan dalam rancangan ini adalah MariaDB.

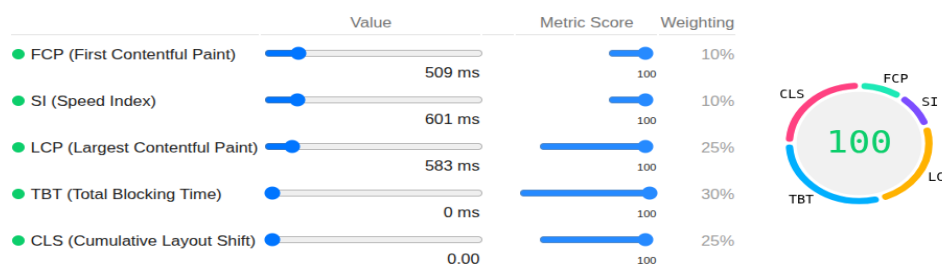
Prosedur implementasi dan pengkodean aplikasi bank sampah digital ini berpusat pada penggunaan *framework DonatJS*. Pada tahap pengkodean, setiap layanan yang telah dirancang dalam arsitektur *microservice* diimplementasikan secara modular dan independen. Pendekatan ini bertujuan untuk mendukung skalabilitas dan kemudahan pemeliharaan sistem di masa depan. Pengelolaan kode sumber dan kontrol versi selama proses pengembangan dilakukan dengan memanfaatkan platform GitHub, yang juga memfasilitasi proses *containerization* untuk penyiapan lingkungan produksi. Objek utama dalam penelitian ini adalah aplikasi bank sampah digital yang dihasilkan dari proses ini. Kinerja dari aplikasi berbasis *DonatJS* tersebut kemudian dibandingkan secara langsung dengan aplikasi serupa yang dibangun menggunakan dua *framework* MVC konvensional yang lebih umum, yaitu Laravel dan CodeIgniter, sebagai kelompok pembanding.

Tahap pengujian dalam penelitian ini dilaksanakan untuk mengevaluasi dan membandingkan performa aplikasi secara kuantitatif. Proses pengumpulan data dilakukan melalui simulasi beban pengguna secara paralel pada ketiga aplikasi (*framework* DonatJS, Laravel, dan CodeIgniter). Kinerja setiap aplikasi diukur dengan menggunakan alat Google Measure yang berfokus pada enam metrik utama *Core Web Vitals*. Metrik-metrik tersebut meliputi *First Contentful Paint* (FCP), *Largest Contentful Paint* (LCP), *Speed Index* (SI), *Total Blocking Time* (TBT), *Cumulative Layout Shift* (CLS), dan *Responsiveness*. Data kuantitatif yang diperoleh dari hasil pengukuran ini kemudian dianalisis secara deskriptif. Hasil analisis disajikan dalam bentuk tabel dan grafik untuk membandingkan secara langsung nilai-nilai metrik dari ketiga *framework* tersebut dan menarik kesimpulan mengenai efektivitas implementasi arsitektur *microservice* dengan DonatJS.

HASIL DAN PEMBAHASAN

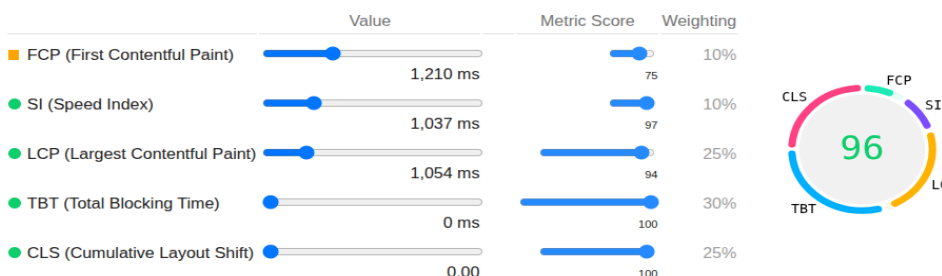
Hasil

Visualisasi hasil uji juga ditampilkan pada Gambar 1 untuk memperjelas perbedaan performa.



Gambar 1. Kinerja Framework DonatJS Pada Metrik FCP, LCP, SI, TBT, dan CLS

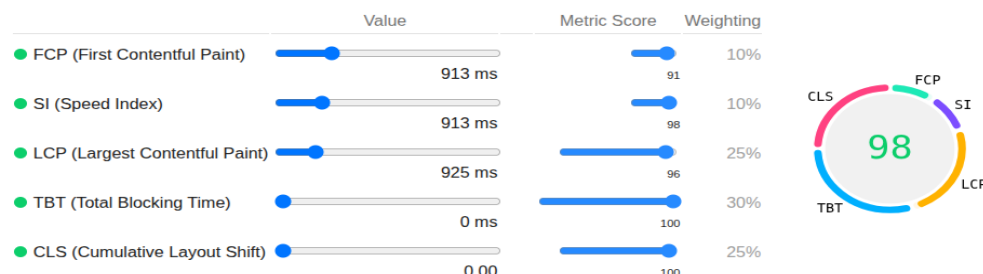
Gambar 1 menampilkan hasil evaluasi kinerja framework DonatJS berdasarkan lima metrik inti performa web. Data menunjukkan performa yang sangat optimal, dimana framework ini berhasil mencapai skor metrik 100 untuk kelima kategori yang diuji. Nilai FCP (First Contentful Paint) tercatat sebesar 509 ms, SI (Speed Index) sebesar 601 ms, dan LCP (Largest Contentful Paint) sebesar 583 ms. Selain itu, TBT (Total Blocking Time) adalah 0 ms dan CLS (Cumulative Layout Shift) adalah 0.00. Dengan skor sempurna di semua metrik yang memiliki bobot berbeda, DonatJS berhasil meraih skor performa keseluruhan 100.



Gambar 2. Kinerja Framework Laravel Pada Metrik FCP, LCP, SI, TBT, dan CLS

Gambar 2 menyajikan data kinerja untuk framework Laravel pada metrik yang sama, yang menghasilkan skor performa keseluruhan 98. Secara rinci, metrik TBT (Total Blocking Time) dan CLS (Cumulative Layout Shift) berhasil mencatatkan skor sempurna 100, dengan nilai masing-masing 0 ms dan 0.00. Namun, metrik yang berkaitan dengan waktu muat menunjukkan sedikit kelambatan. FCP (First Contentful Paint) tercatat 913 ms dengan skor 91,

SI (Speed Index) 913 ms dengan skor 98, dan LCP (Largest Contentful Paint) 925 ms dengan skor 96. Kinerja ini tetap tergolong sangat baik.



Gambar 3. Kinerja Framework CodeIgniter Pada Metrik FCP, LCP, SI, TBT, dan CLS

Gambar 3 mengilustrasikan hasil pengujian kinerja untuk framework CodeIgniter yang mencatatkan skor performa keseluruhan 96. Sama seperti framework Laravel, CodeIgniter menunjukkan kinerja sempurna pada metrik TBT (0 ms) dan CLS (0.00), yang keduanya mendapat skor 100. Metrik SI (1,037 ms) dan LCP (1,054 ms) juga masih sangat baik dengan skor 97 dan 94. Namun, metrik FCP (First Contentful Paint) menunjukkan waktu yang lebih lama, yaitu 1,210 ms, yang menghasilkan skor metrik 75. Meskipun demikian, skor akhir CodeIgniter tetap menunjukkan performa yang tinggi.

Pembahasan

Hasil penelitian ini memperlihatkan bahwa framework DonatJS yang dipadukan dengan arsitektur microservice memberikan keunggulan performa yang signifikan dibandingkan dengan framework MVC konvensional. Data pengujian (Gambar 1) menunjukkan DonatJS meraih skor performa keseluruhan 100, sementara Laravel (Gambar 2) dan CodeIgniter (Gambar 3) mencatatkan skor 98 dan 96. Keunggulan ini terutama terlihat pada metrik waktu muat. First Contentful Paint (FCP) DonatJS tercatat 509 ms, jauh lebih cepat dibandingkan Laravel (913 ms) dan CodeIgniter (1.210 ms). Begitu pula pada Largest Contentful Paint (LCP), DonatJS (583 ms) mengungguli Laravel (925 ms) dan CodeIgniter (1.054 ms). Kecepatan rendering konten awal dan konten terbesar ini mendukung teori yang menyatakan bahwa framework yang ringan mampu mempercepat pemrosesan antarmuka pengguna (Blinowski et al., 2022; Putra et al., 2025; Sinlae et al., 2024). Speed Index (SI) yang rendah pada DonatJS (601 ms) juga mengonfirmasi bahwa pengguna memperoleh tampilan visual yang utuh lebih cepat, meningkatkan kualitas pengalaman penggunaan aplikasi.

Selain kecepatan pemuatan, metrik interaktivitas dan stabilitas visual juga menunjukkan kinerja optimal DonatJS. Nilai Total Blocking Time (TBT) yang esensial untuk responsiveness aplikasi, tercatat 0 ms (skor 100) untuk DonatJS. Meskipun Laravel dan CodeIgniter juga mencatatkan TBT 0 ms, keunggulan DonatJS pada metrik pemuatan memberikan nilai keseluruhan yang lebih superior. TBT 0 ms mengindikasikan bahwa main thread tidak mengalami pemblokiran, sehingga aplikasi mampu merespons input pengguna secara instan. Hal ini sangat penting dalam konteks aplikasi bank sampah digital, di mana sistem harus mampu memproses transaksi paralel seperti penyetoran, verifikasi saldo, dan penarikan insentif secara real time. Stabilitas tata letak, yang diukur dengan Cumulative Layout Shift (CLS), juga mencatatkan nilai sempurna 0.00. Temuan ini konsisten dengan standar Core Web Vitals yang ditetapkan Google sebagai indikator utama kualitas pengalaman pengguna di web.

Keunggulan performa DonatJS tidak dapat dipisahkan dari implementasi arsitektur microservice yang menjadi inti desain penelitian ini. Framework MVC konvensional seperti Laravel dan CodeIgniter cenderung bersifat monolitik. Sebaliknya, DonatJS dalam penelitian

ini dirancang untuk memanfaatkan modularitas microservice secara penuh, memisahkan layanan inti (autentikasi, transaksi, pelaporan) menjadi modul-modul independen. Temuan ini sejalan dengan hasil penelitian Lenarduzzi et al. (2020) yang menyatakan bahwa migrasi dari arsitektur monolitik ke microservice terbukti mampu mengurangi beban teknis dan meningkatkan kinerja sistem. Dengan hanya memuat layanan yang diperlukan untuk permintaan spesifik, efisiensi rendering DonatJS meningkat drastis. Keberhasilan ini juga mendukung pandangan Meijer et al. (2024), yang menekankan bahwa pemisahan layanan pada microservice merupakan kunci untuk mendukung skalabilitas sistem ketika jumlah pengguna aplikasi meningkat.

Dari perspektif aplikasi lingkungan, hasil penelitian ini memiliki implikasi praktis yang penting. Keberlanjutan program bank sampah digital sangat bergantung pada adopsi teknologi yang andal oleh nasabah dan pengelola. Studi Brasika et al. (2023) di Bali menyoroti perlunya ketersediaan sistem digital yang reliabel untuk mendukung pengelolaan sampah plastik. Senada dengan itu, Fatmawati et al. (2022) menegaskan bahwa inovasi kebijakan terkait bank sampah perlu didukung oleh infrastruktur teknologi yang stabil dan responsif. Penelitian ini secara langsung menjawab kebutuhan tersebut. Dengan menunjukkan kinerja superior pada metrik-metrik Core Web Vitals, aplikasi bank sampah berbasis DonatJS dan microservice membuktikan dirinya sebagai alternatif teknologi yang sangat efektif untuk mendukung keberlanjutan digital dalam praktik ekonomi sirkular di tingkat komunitas, memastikan pengguna mendapatkan pengalaman yang cepat dan tidak frustratif.

Analisis komparatif dalam studi ini juga menyentuh aspek efisiensi pengembangan. Framework MVC yang lebih matang seperti Laravel dan CodeIgniter, meskipun kaya fitur, seringkali membawa overhead fungsionalitas yang relatif kompleks dan mungkin tidak seluruhnya diperlukan untuk aplikasi skala komunitas. Implementasi DonatJS, yang dikembangkan menggunakan metodologi Extreme Programming (XP), menunjukkan bahwa framework yang lebih ringan dan modular lebih mudah diintegrasikan dalam konteks microservice (Cheripurathu & Kulkarni, 2024; Hussein et al., 2025; Wyciślik et al., 2023). Fleksibilitas ini sangat bermanfaat selama proses pengkodean iteratif dan pemeliharaan. Temuan ini memperkuat gagasan yang dikemukakan oleh Černý et al. (2023), yang berargumen bahwa framework yang sederhana namun modular seringkali lebih sesuai untuk aplikasi berbasis komunitas yang menuntut efisiensi sumber daya dan keberlanjutan jangka panjang dibandingkan kelengkapan fitur enterprise yang masif.

Meskipun demikian, penelitian ini memiliki keterbatasan metodologis yang perlu diakui. Pengujian kinerja dilakukan pada skala simulasi yang terkontrol. Walaupun simulasi beban pengguna paralel menggunakan Google Measure memberikan data kuantitatif yang objektif, skenario ini belum tentu sepenuhnya mereplikasi kompleksitas dan variabilitas penggunaan di dunia nyata. Validitas performa aplikasi, terutama dalam hal skalabilitas saat menangani volume transaksi yang fluktuatif dari pengguna massal di lapangan, perlu diuji lebih lanjut. Skor performa 100 yang dicapai DonatJS dalam pengujian ini merupakan baseline kinerja optimal dalam kondisi ideal, namun kinerjanya di bawah tekanan operasional sesungguhnya masih memerlukan validasi empiris lebih lanjut untuk memastikan keandalannya dalam implementasi skala besar.

Keterbatasan lain dari penelitian ini adalah fokus analisis yang terbatas pada metrik Core Web Vitals dan mengabaikan aspek krusial lainnya seperti keamanan siber dan integrasi multi-platform. Arsitektur microservice dikenal memiliki tantangan keamanan yang unik, seperti yang ditekankan oleh Berardi et al. (2022) dan Nasab et al., (2021), terutama terkait otentikasi antar layanan dan keamanan endpoint API. Mengingat aplikasi bank sampah digital mengelola data finansial nasabah, aspek keamanan menjadi tidak dapat dinegosiasikan. Oleh

karena itu, penelitian selanjutnya sangat disarankan untuk memperluas lingkup pengujian. Agenda riset ke depan harus mencakup *penetration testing* yang rigor, analisis keamanan arsitektur, serta pengujian keandalan sistem dalam skenario penggunaan real-world dengan volume pengguna yang jauh lebih besar dan beragam.

KESIMPULAN

Penelitian ini secara konklusif menunjukkan bahwa *framework* DonatJS yang dipadukan dengan arsitektur *microservice* memberikan keunggulan performa yang signifikan dibandingkan *framework MVC* konvensional seperti Laravel dan CodeIgniter. Keunggulan ini terbukti secara kuantitatif pada seluruh metrik *Core Web Vitals*, di mana DonatJS mencapai skor performa 100 dengan waktu *First Contentful Paint (FCP)* dan *Largest Contentful Paint (LCP)* yang jauh lebih cepat. Selain itu, *framework* ini mencatatkan *Total Blocking Time (TBT)* 0 ms dan *Cumulative Layout Shift (CLS)* 0.00, yang mengindikasikan *responsiveness* instan dan stabilitas visual yang sempurna. Keunggulan performa ini tidak terlepas dari desain arsitektur *microservice* yang *modular* dan ringan, yang mampu memisahkan layanan inti secara independen. Berbeda dengan pendekatan *monolithic MVC* yang cenderung membawa *overhead* fungsionalitas, arsitektur ini memungkinkan *rendering* yang lebih efisien dan skalabilitas yang lebih baik, menjawab kebutuhan akan sistem bank sampah digital yang andal dan memiliki *user experience* berkualitas tinggi.

Meskipun demikian, penting untuk mengakui bahwa temuan ini didasarkan pada *simulasi* pengujian yang terkontrol dan belum sepenuhnya mereplikasi kompleksitas penggunaan di dunia nyata. Kinerja superior DonatJS dalam *skenario* ideal ini perlu divalidasi lebih lanjut untuk memastikan *skalabilitas* dan keandalannya di bawah tekanan volume transaksi massal yang fluktuatif. Keterbatasan utama lainnya adalah fokus yang sempit pada metrik *Core Web Vitals*, dengan mengabaikan aspek krusial seperti keamanan *siber*. Oleh karena itu, penelitian di masa depan sangat disarankan untuk memperluas lingkup pengujian. Agenda riset ke depan harus mencakup *penetration testing* yang rigor untuk menganalisis keamanan arsitektur *microservice*, terutama pada *endpoint API* yang mengelola data finansial nasabah. Selain itu, perlu dilakukan pengujian *real-world load testing* dengan volume pengguna yang jauh lebih besar untuk memvalidasi performa sistem dalam implementasi skala penuh.

DAFTAR PUSTAKA

- Berardi, D., et al. (2022). Smells And Refactorings For Microservices Security: A Multivocal Literature Review. *PeerJ Computer Science*, 8, e779. <https://doi.org/10.7717/peerj-cs.779>
- Blinowski, G., et al. (2022). Monolithic Vs. Microservice Architecture: A Performance And Scalability Evaluation. *IEEE Access*, 10, 23985–23999. <https://doi.org/10.1109/ACCESS.2022.3152803>
- Brasika, I. B. M., et al. (2023). Evaluating The Collection And Composition Of Plastic Waste: A Study Of Digital Waste Banks In Bali. *Waste Management & Research*, 41(12), 1805–1814. <https://doi.org/10.1177/0734242X221123490>
- Černý, T., et al. (2023). Catalog And Detection Techniques Of Microservice Anti-Patterns And Bad Smells. *Journal Of Systems And Software*, 201, 111829. <https://doi.org/10.1016/j.jss.2023.111829>
- Fatmawati, F., et al. (2022). Waste Bank Innovation Policy: A Study Of Makassar And Bantaeng. *Sustainability*, 14(13), 7974. <https://doi.org/10.3390/su14137974>

- Lenarduzzi, V., et al. (2020). Does Migrating A Monolithic System To Microservices Decrease The Technical Debt? *Journal Of Systems And Software*, 169, 110710. <https://doi.org/10.1016/j.jss.2020.110710>
- Meijer, W., et al. (2024). Experimental Evaluation Of Architectural Software Performance Design Patterns In Microservices. *Journal Of Systems And Software*, 206, 112183. <https://doi.org/10.1016/j.jss.2024.112183>
- Sismadi, W., et al. (2024). Implementasi Arsitektur Microservices Pada Web Aplikasi Penerimaan Mahasiswa Baru. *EDUTECH: Jurnal Inovasi Pendidikan Berbantuan Teknologi*, 4(2), 105–114. <https://doi.org/10.51878/edutech.v4i2.3062>
- Dincă, A.-M., et al. (2022). Performance Enhancements For Database Transactions. *International Journal Of Information Security And Cybercrime*, 11(2), 29. <https://doi.org/10.19107/ijisc.2022.02.02>
- Fakhrurrazi, F., et al. (2022). Pengembangan Desa Digital Dalam Pelayanan Publik Dan Kearsipan Berbasis Teknologi Informasi Di Gampong Reulet Timur. *RAMBIDEUN : Jurnal Pengabdian Kepada Masyarakat*, 5, 252. <https://doi.org/10.51179/pkm.v5i3.1468>
- Heimbach, L., et al. (2023). DeFi And NFTs Hinder Blockchain Scalability. In *Lecture Notes In Computer Science* (p. 291). Springer Science+Business Media. https://doi.org/10.1007/978-3-031-47751-5_17
- Kumar, C. S., et al. (2025). A Dynamic Trading Approach Based On Walrasian Equilibrium In A Blockchain-Based NFT Framework For Sustainable Waste Management. *Mathematics*, 13(3), 521. <https://doi.org/10.3390/math13030521>
- Purnami, W. (2021). Pengelolaan Sampah Di Lingkungan Sekolah Untuk Meningkatkan Kesadaran Ekologi Siswa. *INKUIRI Jurnal Pendidikan IPA*, 9(2), 119. <https://doi.org/10.20961/inkui.v9i2.50083>
- Pusmiati, P., et al. (2025). Hubungan Tingkat Pengetahuan Dengan Perilaku Pengelolaan Sampah Di Kampung Yaturaharja Arso X. *SCIENCE Jurnal Inovasi Pendidikan Matematika Dan IPA*, 5(2), 493. <https://doi.org/10.51878/science.v5i2.4875>
- Rahman, W. N. W. Ab., & Shien, J. N. Y. (2015). Facilitate Recycle Activities Via ICT And Physical Approach. *Applied Mechanics And Materials*, 747, 273. <https://doi.org/10.4028/www.scientific.net/amm.747.273>
- ROVANDA, I. G., et al. (2024). Hubungan Tingkat Pengetahuan Dan Dukungan Tenaga Kesehatan Lingkungan Terhadap Perilaku Pengelolaan Sampah Di Bank Sampah Ramli Graha Indah. *CENDEKIA Jurnal Ilmu Pengetahuan*, 4(4), 378. <https://doi.org/10.51878/cendekia.v4i4.3328>
- Sendika, M., & Frinaldi, A. (2025). Transformasi Budaya Organisasi Di Sektor Publik: Inovasi Menuju Pelayanan Publik Yang Lebih Responsif. *SOCIAL Jurnal Inovasi Pendidikan IPS*, 5(2), 371. <https://doi.org/10.51878/social.v5i2.5376>
- Septiani, A., et al. (2022). Peranan E-Government Dalam Pelayanan Publik. *JURNAL SYNTAX IMPERATIF Jurnal Ilmu Sosial Dan Pendidikan*, 3(5), 302. <https://doi.org/10.36418/syntax-imperatif.v3i5.183>
- SUWITO, A. S. H., & MANAO, A. H. I. (2024). Uji Teknis Berwawasan Lingkungan Alat Perajang Sampah Organik Berbahan Logam Dan Komposit Berlapiskan Fiber Glass. *CENDEKIA Jurnal Ilmu Pengetahuan*, 4(3), 279. <https://doi.org/10.51878/cendekia.v4i3.3115>
- Cheripurathu, K. G., & Kulkarni, S. (2024). Integrating Microservices And Microfrontends: A Comprehensive Literature Review On Architecture, Design Patterns, And

- Implementation Challenges. *Journal Of Scientific Research And Technology*, 1. <https://doi.org/10.61808/jsrt115>
- Hussein, S., et al. (2025). A Comparative Analysis Of Programming Languages Used In Microservices. *International Journal Of Computational And Experimental Science And Engineering*, 11(1). <https://doi.org/10.22399/ijcesen.977>
- Nasab, A. R., et al. (2021). An Empirical Study Of Security Practices For Microservices Systems. Zenodo (CERN European Organization for Nuclear Research). <https://doi.org/10.5281/zenodo.5803080>
- Putra, F. P. E., et al. (2025). Analysis Comparative Of Performance Optimization Techniques For PHP Framework Testing: Laravel, CodeIgniter, Symfony. *Brilliance Research Of Artificial Intelligence*, 5(1), 242. <https://doi.org/10.47709/brilliance.v5i1.5989>
- Sinlae, F., et al. (2024). Penggunaan Framework Laravel Dalam Membangun Aplikasi Website Berbasis PHP. *Jurnal Siber Multi Disiplin*, 2(2), 119. <https://doi.org/10.38035/jsmd.v2i2.186>
- Wycislik, Ł., et al. (2023). A Comparative Assessment Of JVM Frameworks To Develop Microservices. *Applied Sciences*, 13(3), 1343. <https://doi.org/10.3390/app13031343>