Jurnal Manajemen Teknologi dan Sistem Informasi (JMS)
Volume 6, Nomor 1, Maret 2026
ISSN 2808-5450 (media cetak), ISSN 2808-5019 (media online)
Available Online at https://ejournal.unama.ac.id/index.php/jms
DOI https://doi.org/ 10.33998/jms.v6i1

Evaluasi Kinerja Arsitektur Web Berbasis Single Page Application Pada Aplikasi Presensi Kuliah

Ali Sadikin[1*], Abdul Rahim[2], Muhammad Wardani[3], Irawan[4]
[1-4] Ilmu Komputer, Universitas Dinamika Bangsa, Jambi, Indonesia
Email: [1]alisadikin@unama.ac.id, [2.*]abdulrahim@unama.ac.id, [3]irawan@unama.ac.id, [4]ardan@unama.ac.id
Email Penulis Korespondensi: abdulrahim@unama.ac.id

Submitted :
23 February 2026

Revision :
11 Maret 2026

Accepted:
12 Maret 2026

Published:
31 Maret 2026

Abstrak– Perkembangan aplikasi web interaktif mendorong penggunaan pendekatan server-driven seperti Livewire sebagai alternatif pengembangan Single Page Application (SPA) tanpa ketergantungan JavaScript yang kompleks. Namun, implikasi performa dari pendekatan ini dibandingkan metode konvensional masih memerlukan evaluasi empiris. Penelitian ini membandingkan kinerja aplikasi berbasis Laravel Blade (Tradisional/AJAX) dengan Livewire pada skenario sistem presensi akademik. Pengujian dilakukan menggunakan k6 untuk simulasi beban serta pengujian melalui browser untuk mengamati pola komunikasi aktual. Hasil pengujian beban menunjukkan Livewire memiliki waktu respons rata-rata 2,7× lebih tinggi dan konsumsi bandwidth hingga 6× lebih besar akibat mekanisme snapshot. Namun, pengujian interaksi browser membuktikan keunggulan arsitektur SPA pada Livewire yang mampu mereduksi permintaan HTTP hingga 88,1% dan beban data navigasi sebesar 78,7% melalui eliminasi full page reload. Selain itu, Livewire menunjukkan stabilitas dengan tingkat kesalahan 0% dibandingkan Blade (0,69%). Temuan ini mengindikasikan bahwa meskipun terdapat overhead pada setiap permintaan, Livewire menawarkan efisiensi jaringan yang lebih baik pada penggunaan berkelanjutan serta stabilitas yang lebih konsisten untuk aplikasi interaktif.

Kata Kunci: Laravel Livewire; Load Testing; Performa Web; Web Arsitektur; Single Page Application

Abstract– *The evolution of interactive web applications has popularized server-driven approaches like Livewire as an alternative for developing Single Page Applications (SPA) without complex JavaScript dependencies. However, the performance implications of this approach compared to conventional methods require empirical evaluation. This study compares the performance of Laravel Blade (Traditional/AJAX) and Livewire within the context of an academic attendance system. Testing was conducted using k6 for load simulation and browser-based analysis to observe actual communication patterns. Load testing results indicate that Livewire has an average response time 2.7× higher and bandwidth consumption up to 6× greater due to the state snapshot mechanism. However, browser interaction tests reveal the architectural superiority of Livewire's SPA model, which reduced HTTP requests by 88.1% and navigation data load by 78.7% by eliminating full page reloads. Furthermore, Livewire demonstrated superior stability with a 0% error rate compared to Blade (0.69%). These findings indicate that while per-request overhead exists, Livewire offers better network resource efficiency during sustained use and more consistent stability for interactive applications.*

Keywords*: Laravel Livewire; Load Testing; Performa Web; Web Architecture; Single Page Application*

1. PENDAHULUAN

Laravel, sebagai kerangka kerja PHP dengan pangsa pasar 50% [1], menyediakan dua pendekatan utama: Blade dengan pemuatan ulang halaman penuh, dan Livewire yang menawarkan komponen reaktif dengan rendering sisi server [2]. Livewire menggunakan mekanisme snapshot untuk mempertahankan status komponen antar permintaan HTTP [3]. Meskipun menyederhanakan pengembangan, pertanyaan kritis muncul mengenai kompromi kinerja dalam aplikasi dengan interaksi sering dan operasi intensif data.

Penelitian terkait menunjukkan Laravel memiliki waktu respons lebih tinggi namun menawarkan fitur matang. Penelitian [4] membandingkan Laravel, Django, dan Node.js, menemukan Express.js memiliki execution speed tercepat sedangkan Laravel berada di posisi ketiga. Penelitian [5] menganalisis Flask, Laravel, dan Express.js menggunakan Postman, menemukan Express.js unggul dengan response time 53-58 ms dan error rate 0%, Laravel 556-683 ms dengan error rate 1,63-2,58%, dan Flask 723-1757 ms tanpa error. [6]membandingkan Laravel dengan Express.js menggunakan JMeter, menemukan Express.js 48,68 ms lebih cepat. Penelitian [7]

Volume 6, Nomor 1, Maret 2026
ISSN 2808-5450 (media cetak), ISSN 2808-5019 (media online)
Available Online at https://ejournal.unama.ac.id/index.php/jms
DOI https://doi.org/ 10.33998/jms.v6i1

menemukan CodeIgniter lebih cepat (1369 ms vs 1984 ms) namun Laravel unggul dalam maintainability. Penelitian [8] melakukan pengujian API dengan load testing, mengidentifikasi batasan kapasitas dan pola degradasi kinerja. Meskipun penelitian ada memberikan wawasan tentang performa framework, belum terdapat studi empiris yang mengevaluasi karakteristik kinerja Livewire menggunakan pengujian beban sistematis, khususnya pada snapshot dan implikasi terhadap penggunaan bandwidth.

Kesenjangan ini penting mengingat adopsi Livewire yang meningkat dalam aplikasi produksi. Sistem presensi dosen, dengan pembaruan sering dan kebutuhan umpan balik real-time, menjadi kasus ideal untuk evaluasi kinerja. Pengembang memerlukan data empiris untuk keputusan arsitektur yang tepat [9].

Penelitian ini mengevaluasi kinerja Livewire pada sistem presensi dosen melalui studi komparatif dengan pendekatan laravel menggunakan K6 dan interaksi browser menggunakan Chrome Developer Tools. Tujuan penelitian: mengukur metrik kinerja kunci, mengidentifikasi penggunaan bandwidth, dan menyusun rekomendasi praktis pemilihan pendekatan. Kontribusi penelitian mencakup data kinerja empiris, analisis mekanisme snapshot dan penggunaan.

2. KAJIAN TEORITIS

2.1 Laravel dan Livewire

Laravel mengadopsi arsitektur MVC dengan penekanan pada produktivitas pengembang [10]. Pendekatan laravel menggunakan full page reload untuk rendering HTML dengan siklus permintaan-respons stateless. Laravel Livewire memungkinkan pengembang membangun antarmuka dinamis tanpa JavaScript ekstensif [2]. Livewire bekerja dengan mekanisme snapshot yang menserialisasi status komponen dan mengirimkannya bersama setiap permintaan AJAX [3]. Livewire memungkinkan pengembang untuk membuat aplikasi Single Page Application (SPA) dengan pendekatan incremental load, di mana setelah pemuatan halaman awal (initial load), interaksi pengguna selanjutnya hanya mengirimkan data perubahan state dan menerima pembaruan DOM secara parsial tanpa harus mengunduh ulang seluruh aset statis seperti CSS dan JavaScript. Mekanisme ini meminimalkan beban jaringan melalui transmisi data yang hanya berfokus pada komponen yang berubah, sehingga memberikan pengalaman pengguna yang lebih mulus dan responsif dibandingkan dengan siklus permintaan-respons tradisional yang memerlukan pemuatan ulang halaman secara penuh (full page reload).

2.2 Metrik Evaluasi Kinerja

Waktu respons merupakan indikator utama kinerja yang dipersepsikan pengguna [11]. Standar Nielsen Norman Group (2020) menyatakan <100ms dipersepsikan instan, 100-300ms cepat, >1000ms lambat [12]. Throughput (req/s) mengindikasikan kapasitas sistem dalam menangani permintaan konkuren [13]. Bandwidth mencakup data terkirim dan diterima, berdampak pada biaya infrastruktur dan pengalaman pengguna mobile. Tingkat kesalahan mencerminkan keandalan sistem under beban [8].

2.3 Tools Pengujian

K6 merupakan alat pengujian beban modern dengan scripting JavaScript dan eksekusi Go untuk kinerja optimal [14]. Virtual Users (VUs) mengeksekusi skenario secara simultan dengan pola ramp-up untuk mensimulasikan lalu lintas realistis [15]. K6 menonjol sebagai alat **pengujian beban modern karena** pendekatan scripting berbasis JavaScript yang sederhana, ekspresif, dan mudah diintegrasikan dengan alur pengembangan perangkat lunak. Selain itu, implementasi K6 menggunakan bahasa Go memberikan **efisiensi eksekusi dan stabilitas tinggi**, sehingga mampu menangani simulasi beban secara andal pada skenario lalu lintas yang kompleks dan berskala besar [16].

Selain pengujian beban otomatis, penelitian ini menggunakan Google Chrome Developer Tools sebagai instrumen pengujian manual melalui fitur Network Tab [17]. Alat ini digunakan untuk mengamati pola komunikasi aktual pada sisi klien, mengukur jumlah permintaan HTTP, serta menganalisis konsumsi bandwidth secara incremental. Melalui Chrome DevTools, perilaku Single Page Application (SPA) pada Livewire dapat divalidasi dengan memantau proses pembaruan DOM secara parsial dan memastikan tidak terjadinya pemuatan ulang halaman secara penuh (full page reload) selama interaksi pengguna berlangsung.

2.4 Penelitian Terkait

Penelitian [4] membandingkan Laravel, Django, dan Node.js dalam REST API, menemukan Laravel memiliki pemrosesan lebih tinggi namun ekosistem matang. [6] menganalisis Laravel vs Express.js menggunakan JMeter, menunjukkan Express.js memiliki throughput lebih tinggi. [7] menemukan CodeIgniter lebih cepat (1369 ms vs 1984 ms) namun Laravel unggul dalam maintainability. [8] melakukan pengujian API dengan load testing, mengidentifikasi batasan kapasitas. Meskipun penelitian ada memberikan wawasan, terdapat kesenjangan dalam evaluasi empiris Livewire menggunakan pengujian beban sistematis terhadap overhead snapshot.

Jurnal Manajemen Teknologi dan Sistem Informasi (JMS)
Volume 6, Nomor 1, Maret 2026
ISSN 2808-5450 (media cetak), ISSN 2808-5019 (media online)
Available Online at https://ejournal.unama.ac.id/index.php/jms
DOI https://doi.org/ 10.33998/jms.v6i1

3. METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Penelitian menggunakan pendekatan kuantitatif dengan desain studi komparatif, dilakukan melalui tahapan:

Gambar 1 Kerangka Kerja Penelitian

Gambar 2 merupakan kerangka kerja penelitian ini, berikut adalah penjelasannya:

1. Studi Literatur
   Kajian terhadap literatur terkait performa kerangka kerja PHP, Laravel Livewire, metrik evaluasi kinerja aplikasi web, dan metodologi pengujian beban menggunakan K6 dan pengujian menggunakan Google Chrome Developer Tools.
2. Persiapan Objek Penelitian
   Penyiapan sistem presensi dosen dalam dua versi: versi laravel menggunakan Laravel 6 dengan template Blade, dan versi Livewire menggunakan Laravel 11 dengan Livewire 3. Kedua versi mengimplementasikan logika bisnis identik dengan database yang sama.

Tabel 1 Perbandingan Implementasi Teknis

Halaman/Fitur	Laravel Blade (Tradisional)	Laravel Livewire (Server-Driven)
Pola Navigasi	*Full Page Reload* (Siklus permintaan-respons penuh)	*Partial Reload* (SPA-like melalui AJAX)
Dashboard Presensi	Merender seluruh struktur HTML di server (*Full SSR*)	Memuat struktur awal + sinkronisasi *state* komponen (*Snapshot*)
Update Status	Memerlukan penulisan JavaScript manual (AJAX/Fetch)	Berbasis *Event-driven* tanpa menulis JS manual (*Declarative*)
Payload Response	Dokumen HTML lengkap atau data JSON murni	Objek JSON berisi *State Snapshot*, *Checksum*, dan *HTML Diff*
Mekanisme Update DOM	Manipulasi DOM manual atau penggantian seluruh halaman	*Automated patching* menggunakan algoritma Morphdom

3. Pengembangan Skenario Pengujian
   Pembuatan skrip pengujian K6 yang mensimulasikan perjalanan pengguna: autentikasi, navigasi ke halaman presensi, dan pembaruan 6 status kehadiran mahasiswa dari total 36 mahasiswa. Konfigurasi menggunakan 5 Virtual Users konkuren dengan durasi pengujian 5 menit per eksekusi.

Tabel 2 Skenario Pengujian

Item	Detail
K6 Version	v0.48.0
Vus	5 concurrent
Duration	5 menit
Flow	Login → Load (36 data) → Update 6x → Repeat
Metrics	Response time, Throughput, Bandwidth, Error rate
Runs	3× per implementasi

Jurnal Manajemen Teknologi dan Sistem Informasi (JMS)
Volume 6, Nomor 1, Maret 2026
ISSN 2808-5450 (media cetak), ISSN 2808-5019 (media online)
Available Online at https://ejournal.unama.ac.id/index.php/jms
DOI https://doi.org/ 10.33998/jms.v6i1

Selain pengujian beban menggunakan k6, dilakukan pengujian interaksi manual menggunakan Chrome DevTools Network. Tujuannya adalah untuk mengukur efisiensi sistem dalam kondisi penggunaan nyata (real-world usage) dengan memantau transmisi data pada setiap tahapan interaksi pengguna secara sekuensial (akses beranda, navigasi, dan pembaruan data).

4. Eksekusi Pengujian Beban
   Pelaksanaan load testing menggunakan K6 versi 0.48.0 pada lingkungan lokal (Intel i7-10700K, 32GB RAM, PHP 7.4/8.2, MySQL 8.0, Apache 2.4). Setiap implementasi diuji 3 kali untuk memastikan konsistensi hasil. Pengujian interaksi browser dilakukan menggunakan Google Chrome dengan mode incognito untuk memastikan request dilakukan tanpa adanya *cache*.
5. Pengumpulan Data
   Pengumpulan metrik kinerja meliputi: waktu respons (rata-rata, median, p95, maksimum), throughput (req/s), bandwidth (data terkirim/diterima), tingkat kesalahan, dan tingkat penyelesaian iterasi. Data di-ekspor dalam format JSON dari K6 dan menggunakan nilai request, transferred, resources, DOMContentLoaded yang ada di Developer Tools.
6. Analisis Data
   Analisis menggunakan statistik deskriptif untuk perbandingan kinerja antar implementasi. Root cause analysis dilakukan melalui inspeksi lalu lintas jaringan menggunakan Developer Tools dan profiling ukuran snapshot komponen Livewire.
7. Penarikan Kesimpulan
   Penyusunan kesimpulan berdasarkan hasil analisis data, identifikasi trade-off antara kedua pendekatan, dan perumusan rekomendasi praktis untuk pemilihan pendekatan yang sesuai dengan karakteristik aplikasi.

Keterbatasan penelitian meliputi perbedaan versi Laravel (v6 vs v11), lingkungan lokal yang tidak sepenuhnya representatif terhadap produksi, dan tingkat konkurensi terbatas (5 VUs)

4. HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian Beban

Tabel 3. Perbandingan Hasil Pengujian Beban

Metrik	Laravel Blade	Livewire	Rasio
Total Permintaan	866	981	**1,13×**
Throughput (req/s)	4,07	4,61	**1,13×**
Tingkat Kesalahan	6 (0,69%)	0 (0%)	0×
Waktu Respons Rata-rata	104,00 ms	280,84 ms	**2,70×**
Waktu Respons Median	98,50 ms	280,00 ms	**2,84×**
Waktu Respons P95	155,70 ms	417,39 ms	**2,68×**
Waktu Respons Maksimum	4096,81 ms	1415,11 ms	**0,35×**
Data Diterima	43,45 MB	258,43 MB	**5,95×**
Data Dikirim	0,64 MB	2,30 MB	**3,59×**

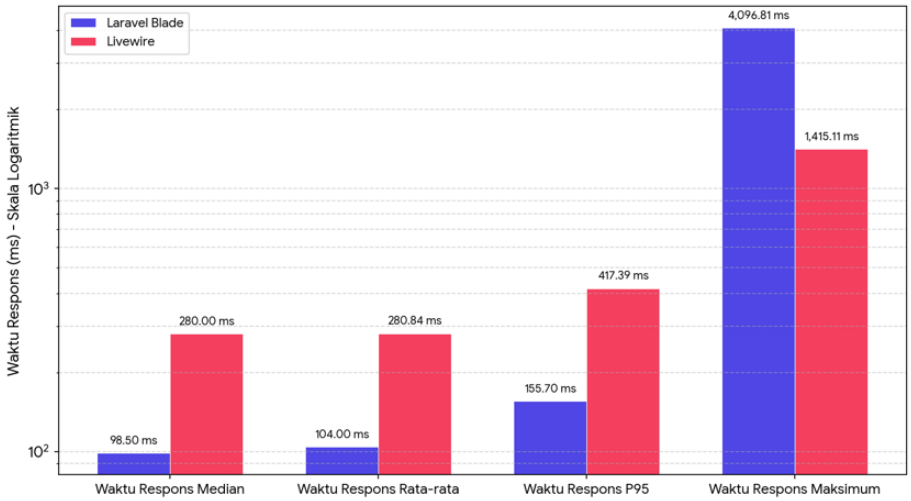
Sumber: Hasil Pengujian K6, Januari 2026

Pendekatan Laravel Blade mencatatkan throughput 4,07 req/s dengan respons rata-rata 104,00 ms, menunjukkan efisiensi tinggi pada server-side rendering. Namun, adanya nilai maksimum hingga 4.096,81 ms dan tingkat kesalahan 0,69% mengindikasikan kerentanan terhadap kegagalan sporadis di bawah beban konkuren. Sebaliknya, Livewire menunjukkan keandalan sempurna (0% error) dengan peningkatan throughput sebesar 13% (4,61 req/s), meski dengan rata-rata respons 280,84 ms atau 2,7 kali lebih tinggi dibandingkan Blade. Dari sisi jaringan, Livewire mengonsumsi bandwidth yang jauh lebih besar, menerima 258,43 MB (sekitar 6x lipat) dan mengirim 2,30 MB data. Tingginya konsumsi ini merupakan konsekuensi dari mekanisme snapshot dan sinkronisasi state pada pengujian beban repetitif k6, yang cenderung mengabaikan efisiensi incremental load pada skenario penggunaan nyata.

4.2 Analisis Komparatif

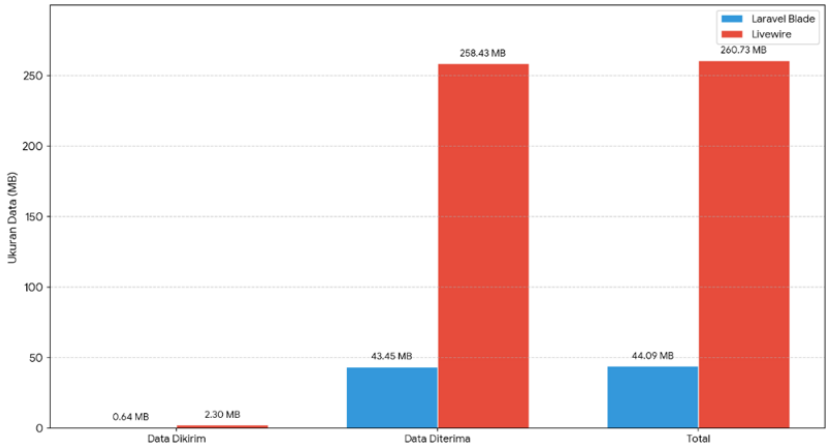
Berdasarkan Tabel 3, Laravel Blade mencatatkan *throughput* 4,07 req/s dengan respons rata-rata 104,00 ms, yang menunjukkan efisiensi tinggi pada pemrosesan *server-side rendering*. Namun, adanya nilai maksimum hingga 4.096,81 ms dan tingkat kesalahan 0,69% mengindikasikan kerentanan terhadap kegagalan sporadis di bawah beban tinggi.

Jurnal Manajemen Teknologi dan Sistem Informasi (JMS)
Volume 6, Nomor 1, Maret 2026
ISSN 2808-5450 (media cetak), ISSN 2808-5019 (media online)
Available Online at https://ejournal.unama.ac.id/index.php/jms
DOI https://doi.org/ 10.33998/jms.v6i1



Gambar 2 Perbandingan Waktu Respon Aplikasi.

Grafik di gambar 2 mempertegas perbedaan karakteristik arsitektural kedua pendekatan. Meskipun Livewire memiliki *baseline* latensi yang lebih tinggi pada kondisi normal (rata-rata dan P95), ia menawarkan perilaku eksekusi yang lebih deterministik dan stabil pada kondisi terburuk, yang ditunjukkan oleh nilai respons maksimum yang jauh lebih rendah dibandingkan Laravel Blade. Berikut perbandingan penggunaan bandwidth selama durasi pengujian:



Gambar 3: Perbandingan Penggunaan Bandwidth

Berdasarkan Gambar 3, Livewire mencatatkan konsumsi data yang jauh lebih besar, menerima 258,43 MB data (sekitar 6x lipat) dibandingkan Laravel Blade. Namun, angka ini merupakan hasil dari mekanisme pengujian repetitif pada k6 yang memicu sinkronisasi *state* penuh secara terus-menerus. Untuk melihat performa efisiensi Livewire pada skenario penggunaan nyata (*Single Page Application*), dilakukan pengujian interaksi browser secara manual.

Namun, konsumsi bandwidth yang tinggi pada pengujian beban otomatis (k6) perlu ditinjau lebih dalam melalui pengujian interaksi manual guna melihat perilaku sistem pada skenario penggunaan nyata (*real-world usage*). Pengujian ini penting karena pengujian beban cenderung mengulang seluruh proses permintaan, sedangkan interaksi browser sebenarnya menunjukkan efisiensi beban incremental (tambahan).

Tabel 4. Hasil Pengujian Interaksi Browser (Server-Driven UI)

Halaman / Aksi	Metrik	Laravel Blade (Tradisional)	Laravel Livewire (SPA)	Efisiensi / Selisih
/beranda	HTTP Requests	59	17	**-71,1%**
(Initial Load)	Transfer Data	3,4 MB	2,4 MB	**-29,4%**
/presensi	HTTP Requests	59	7	**-88,1%**
(Navigasi)	Transfer Data	3,6 MB	766 kB (0,76 MB)	**-78,7%**
/update	HTTP Requests	2	6	+200%*

Volume 6, Nomor 1, Maret 2026
ISSN 2808-5450 (media cetak), ISSN 2808-5019 (media online)
Available Online at https://ejournal.unama.ac.id/index.php/jms
DOI https://doi.org/ 10.33998/jms.v6i1

(Aksi Data)	Transfer Data	1,4 kB	464 kB (0,46 MB)	+Overhead Snapshot

Hasil pengujian interaksi browser mengonfirmasi efisiensi arsitektur Single Page Application (SPA) pada Livewire dalam mereduksi beban transmisi data. Pada tahap navigasi dari /beranda ke /presensi, Laravel Blade melakukan pemuatan ulang aset secara penuh (full page reload) sebesar 3,6 MB melalui 59 permintaan HTTP. Sebaliknya, Livewire hanya memerlukan beban incremental sebesar 0,76 MB melalui 7 permintaan HTTP, yang merepresentasikan efisiensi bandwidth sebesar 78,7%. Meskipun terdapat overhead snapshot pada aksi /update, secara akumulatif Livewire jauh lebih efisien dalam penggunaan berkelanjutan karena berhasil mengeliminasi pengunduhan ulang aset statis secara redundan melalui pembaruan DOM parsial.

5. DISKUSI DAN KESIMPULAN

5.1 Diskusi

Livewire menunjukkan waktu respons rata-rata 2,7× lebih tinggi dibandingkan Laravel Blade (median 2,8× dan P95 2,7×). Namun, Livewire menawarkan stabilitas yang lebih unggul dengan waktu respons maksimum 0,35× lebih rendah. Fenomena ini mencerminkan karakteristik arsitektural di mana jalur pemrosesan Blade lebih efisien pada kondisi normal tetapi rentan terhadap outlier ekstrem, sedangkan Livewire menghasilkan perilaku eksekusi yang lebih deterministik melalui mekanisme sinkronisasi state.

Dari aspek konsumsi jaringan, pengujian beban otomatis (k6) mencatat transfer data Livewire 5,9× lebih besar dibandingkan Blade. Namun, hasil tersebut tidak merepresentasikan efisiensi Livewire dalam skenario dunia nyata karena simulasi k6 melakukan pemuatan ulang halaman secara penuh (full page reload) pada setiap iterasi, yang secara teknis meniadakan keunggulan arsitektur Single Page Application (SPA). Sebaliknya, pengujian interaksi browser memberikan validasi empiris atas efisiensi Livewire, dengan reduksi jumlah permintaan HTTP hingga 88,1% saat navigasi (dari 59 menjadi 7 permintaan).

Data pengujian menunjukkan bahwa setelah pemuatan awal (initial load), navigasi ke halaman presensi pada Livewire hanya memerlukan beban incremental sebesar 0,76 MB, jauh di bawah Laravel Blade yang mencapai 3,6 MB akibat pengunduhan ulang aset statis secara redundan. Meskipun terdapat overhead sebesar 464 kB pada aksi pembaruan data akibat pengiriman snapshot, Livewire secara akumulatif jauh lebih efisien dalam sesi penggunaan berkelanjutan. Dengan tingkat kesalahan 0%, Livewire menjadi solusi arsitektural yang lebih unggul untuk sistem presensi akademik yang memprioritaskan stabilitas eksekusi dan efisiensi sumber daya jaringan jangka panjang

5.2 Kesimpulan

Pengujian yang dilakukan menggunakan k6 tidak merepresentasikan interaksi yang terjadi di browser, terutama pada arsitektur SPA. Dari hasil eksperimen ini, penggunaan Livewire dapat meningkatkan performa aplikasi lebih cepat dengan menerapkan SPA yang mengeliminasi siklus full page reload di setiap navigasi. Peningkatan performa ini dicapai melalui optimasi transmisi data, di mana browser hanya meminta dan menerima fragmen HTML atau state komponen yang berubah secara dinamis melalui pembaruan DOM parsial. Akibatnya, beban kerja jaringan berkurang drastis setelah pemuatan awal (initial load), yang tidak hanya mempercepat waktu respons antarmuka tetapi juga menjamin stabilitas eksekusi dengan tingkat kesalahan 0%. Dengan demikian, Livewire terbukti lebih efisien dalam penggunaan riil berkelanjutan karena mampu mempertahankan responsivitas aplikasi tanpa harus mengunduh ulang aset statis secara redundan

REFERENCES

[1] JetBrains, “Developer Ecosystem Survey 2023.” 2023. [Online]. Available: https://www.jetbrains.com/lp/devecosystem-2023/

[2] C. Porzio, “Laravel Livewire Documentation.” 2024. [Online]. Available: https://livewire.laravel.com/docs/

[3] M. Stauffer, “Understanding Livewire: Server-Side Framework for Laravel.” 2023. [Online]. Available: https://mattstauffer.com/blog/

[4] A. Amarulloh, K. Kurniasih, and M. Muchlis, “Analisis Perbandingan Performa Web Service REST Menggunakan Framework Laravel, Django, dan Node JS pada Aplikasi Berbasis Website,” *J. Tek. Inform.*, vol. 9, no. 1, pp. 12–17, 2023.

[5] B. Purwanto, H. Nugroho, and A. Wijayanto, “Pengembangan Dashboard Akademik Berbasis Data Warehouse di Perguruan Tinggi,” *J. Teknol. Inf. Dan Komput.*, vol. 4, no. 1, pp. 12–20, 2018.

[6] M. S. Hadinata and R. W. Stianingsih, “Analisis Perbandingan Performa RESTful API antara Express.js dengan Laravel Framework,” *J. Inform. Dan Tek. Elektro Terap. JITET*, vol. 7, no. 2, pp. 545–554, 2024, doi: 10.35313/jitet.v7i2.3845.

Volume 6, Nomor 1, Maret 2026
ISSN 2808-5450 (media cetak), ISSN 2808-5019 (media online)
Available Online at https://ejournal.unama.ac.id/index.php/jms
DOI https://doi.org/ 10.33998/jms.v6i1

[7] W. M. Kansha, "Analisis Perbandingan Struktur dan Performa Framework CodeIgniter dan Laravel dalam Pengembangan Web Application," *J. Tek. Inform. Stmik Antar Bangsa*, vol. 9, no. 1, pp. 25–30, 2023.

[8] 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, 2023, doi: 10.38101/sisfotek.v13i1.2656.

[9] F. A. Pradana, "Perancangan Sistem Presensi Deteksi Wajah Berbasis Website (Studi Kasus Laboratorium Sistem Manufaktur Terintegrasi UII)," PhD Thesis, Universitas Islam Indonesia, 2024.

[10] T. Otwell, "Laravel Documentation." 2024. [Online]. Available: https://laravel.com/docs/

[11] W. Liu, L. Guo, Y. W. Heng, C. Li, A. E. Hassan, and others, "Screencast-Based Analysis of User-Perceived GUI Responsiveness," *ArXiv Prepr. ArXiv250801337*, 2025.

[12] A. N. Hikmat, "Evaluasi Desain Interface Pada Aplikasi Ipusnas Berdasarkan Teori Evaluasi Heuristik Nielsen," *LIBRIA*, vol. 16, no. 1, pp. 16–36, 2024.

[13] M. P. Ambara, M. Sudiarta, S. M. Suryaniadi, and I. G. T. S. Dharma, "Implementation of the CodeIgniter Framework in the Development of a Cloud-Based Competency Assessment System," in *International Conference on Sustainable Green Tourism Applied Science-Engineering Applied Science 2025 (ICOSTAS-EAS 2025)*, Atlantis Press, 2025, pp. 421–430.

[14] Grafana Labs, "K6 Load Testing Documentation." 2024. [Online]. Available: https://k6.io/docs/

[15] Y. Teslenko, "PERFORMANCE PERCENTILE ANALYSIS FOR API-BASED TESTING," *Авторський Колектив*, p. 235, 2025.

[16] F. Pratama and A. Farisi, "Analisis Perbandingan Kinerja Backend API Menggunakan PHP, Golang, dan JavaScript.," *Techno Com*, vol. 24, no. 1, 2025.

[17] A. Setiawan and Y. Yamasari, "Evaluasi Performa Website Sistem Informasi Kesejahteraan Sosial (SIKS-NG) Di Kelurahan Lembeyan Kulon Kabupaten Magetan Dengan Google Light House," *J. Inform. Comput. Sci. JINACS*, vol. 7, no. 01, pp. 218–227, 2025.