INFOTECH journal
ISSN : 2460-1861 (Print), 2615-4250 (Online)
Vol. 12 No. 2 2026

PERBANDINGAN PERFORMA FRAMEWORK VUE.JS, REACT.JS, DAN SVELTE.JS TERHADAP APLIKASI WEB BERBASIS JAVASCRIPT

I Made Putra Sedana [1], Eddy Muntina Dharma [2], Jauzaa Maylia Suhendro [3]
[1,2,3]*Universitas Primakara, Kota Denpasar, Bali, Indonesia*
Penulis Korespondensi: putrasedana03@gmail.com

ABSTRAK

Pengembangan aplikasi web modern saat ini sangat bergantung pada penggunaan *front-end framework* JavaScript. Meskipun *framework* berbasis *runtime virtual DOM* seperti React.js dan Vue.js sangat populer, kehadirannya berpotensi menimbulkan *overhead rendering* pada aplikasi skala besar. Di sisi lain, *framework* berbasis *compile-time* seperti Svelte.js menawarkan alternatif tanpa *virtual DOM*. Kurangnya studi yang membandingkan ketiga *framework* ini secara bersamaan dengan metrik seragam memicu kesulitan bagi pengembang dalam memilih teknologi yang optimal. Penelitian ini bertujuan mengukur dan membandingkan performa React.js, Vue.js, dan Svelte.js dari aspek kecepatan pemuatan, interaktivitas, dan stabilitas tata letak. Metode eksperimen kuantitatif diterapkan dengan membangun tiga aplikasi *Single Page Application* (SPA) berstruktur identik berbasis konsep TodoMVC dengan data awal 100 item. Pengujian dilakukan berulang sebanyak 15 kali memanfaatkan platform GTmetrix. Hasil penelitian menunjukkan Svelte.js memberikan performa paling optimal dengan nilai *Total Blocking Time* (TBT) terendah sebesar 3,47 ms, jauh mengungguli Vue.js (33,87 ms) dan React.js (37,67 ms). Vue.js dan Svelte.js meraih skor performa sempurna 18/18, sedangkan React.js meraih skor 17/18 karena rata-rata metrik Largest Contentful Paint (LCP) mencapai 1,29 s.

***Kata Kunci**: Performa Website, Vue.js, React.js, Svelte.js, GTmetrix*

Riwayat Artikel :

Tanggal diterima : 22-07-2026 Tanggal terbit : 13-09-2026

Kutipan :

Sedana, I. M. P., Dharma, E. M., & Suhendro, J. M. (2026). PERBANDINGAN PERFORMA FRAMEWORK VUE.JS, REACT.JS, DAN SVELTE.JS TERHADAP APLIKASI WEB BERBASIS JAVASCRIPT. INFOTECH Journal, 12(2), 240–244. https://doi.org/10.31949/infotech.v12i2.19403

1. PENDAHULUAN

Perkembangan teknologi web modern mendorong kebutuhan akan aplikasi yang tidak hanya fungsional, tetapi juga memiliki performa tinggi dan responsivitas yang optimal (Arya et al., 2024; Iswari & Dharma, 2025; Krisna et al., 2024). *Website* saat ini telah bertransformasi menjadi platform utama layanan digital yang secara langsung memengaruhi pengalaman pengguna serta daya saing suatu organisasi (Martha et al., 2025; Ngurah et al., 2022). Dalam konteks ini, kualitas performa menjadi faktor krusial; *website* yang responsif dan stabil akan meningkatkan kenyamanan pengguna, sementara performa yang buruk dapat menurunkan efektivitas sistem secara keseluruhan (Octaviani & Andraini, 2022; Selsa et al., 2024). Oleh karena itu, pemilihan teknologi pengembangan, khususnya pada sisi *front-end*, menjadi faktor penentu utama dalam menjamin kualitas dan efisiensi aplikasi yang dihasilkan (Sofi'ie & Qoiriah, 2023).

Dalam lanskap pengembangan *front-end* saat ini, *framework* JavaScript seperti React.js dan Vue.js menjadi teknologi yang paling dominan digunakan (Wijaya & Farisi, 2025). Keduanya mengadopsi konsep *Virtual DOM* untuk meningkatkan efisiensi *rendering* antarmuka pengguna, namun mekanisme ini tetap memerlukan proses komputasi tambahan berupa *diffing* antara *Virtual DOM* dan *Real DOM* yang berpotensi memengaruhi performa pada aplikasi dengan kompleksitas tinggi. Sebagai alternatif, Svelte.js hadir sebagai *compiler framework* yang melakukan kompilasi kode pada tahap *build-time* (Zulhijaya et al., 2025), sehingga mampu menghasilkan aplikasi yang lebih ringan dan lebih cepat dieksekusi oleh browser (Dubaj & Pańczyk, 2022).

Sejumlah penelitian terdahulu telah membahas perbandingan performa antar-*framework* tersebut, namun dengan cakupan yang terbatas. Studi oleh (Sofi'ie & Qoiriah, 2023) menunjukkan bahwa Vue.js memiliki performa lebih baik dibandingkan React.js, sementara penelitian oleh (Zulhijaya et al., 2025) menemukan bahwa Svelte.js unggul dalam kecepatan pemuatan. Meskipun demikian, sebagian besar literatur yang ada cenderung membandingkan hanya dua *framework* secara berpasangan (Sofi'ie & Qoiriah, 2023; Wijaya & Farisi, 2025; Zulhijaya et al., 2025). Kondisi ini menunjukkan bahwa masih belum banyak penelitian yang melakukan evaluasi secara simultan terhadap tiga *framework* seperti React.js, Vue.js, dan Svelte.js menggunakan metrik performa yang seragam, sehingga belum tersedia gambaran komprehensif mengenai *framework* mana yang paling optimal dalam standar aplikasi web modern saat ini.

Berdasarkan kesenjangan tersebut, penelitian ini bertujuan melakukan analisis komparatif performa React.js, Vue.js, dan Svelte.js melalui metode eksperimen kuantitatif, dengan pengujian yang difokuskan pada performa *loading* halaman web dari sisi *front-end* di *browser* pengguna. Eksperimen dilakukan pada aplikasi *Single Page Application* (SPA) dengan struktur fungsionalitas identik mengacu pada standar TodoMVC, menggunakan *framework* sebagai variabel independen dan performa aplikasi sebagai variabel dependen yang diukur secara objektif menggunakan *tool* GTmetrix (Marsa et al., 2025), mencakup indikator *First Contentful Paint* (FCP), *Largest Contentful Paint* (LCP), *Speed Index* (SI), *Time to Interactive* (TTI), *Total Blocking Time* (TBT), dan *Cumulative Layout Shift* (CLS) (Pandya et al., 2024; Widi et al., 2024). Hasil analisis ini diharapkan dapat menjadi referensi objektif bagi pengembang dalam menentukan *framework* yang paling sesuai dengan karakteristik dan kebutuhan proyek.

2. METODE

2.1. Desain Eksperimen Perbandingan Framework

Penelitian ini menggunakan pendekatan eksperimen kuantitatif untuk membandingkan performa tiga *framework* JavaScript *front-end*, yaitu React.js, Vue.js, dan Svelte.js. Metode eksperimen dipilih karena memungkinkan peneliti mengamati pengaruh variabel independen terhadap variabel dependen dalam kondisi lingkungan yang terkontrol (Iriyadi, 2024). Ketiga aplikasi uji dibangun dengan struktur, tampilan, dan fungsionalitas yang identik mengacu pada konsep TodoMVC, sehingga perbedaan hasil pengujian hanya dipengaruhi oleh *framework* yang digunakan, bukan faktor lain di luar variabel penelitian. Setiap aplikasi diimplementasikan sebagai *Single Page Application* (SPA) dengan fitur *Create*, *Read*, *Update*, dan *Delete* (CRUD) terhadap data tugas. Data dikelola melalui *state* internal aplikasi dan disimpan secara lokal menggunakan *localStorage* tanpa komunikasi jaringan eksternal, sehingga hasil pengujian tidak dipengaruhi oleh variabel latensi jaringan. Ketiga aplikasi dikompilasi dalam mode *production* dan di-*deploy* pada satu *Virtual Private Server* (VPS) yang sama menggunakan *web server* Nginx untuk menjamin konsistensi lingkungan pengujian.

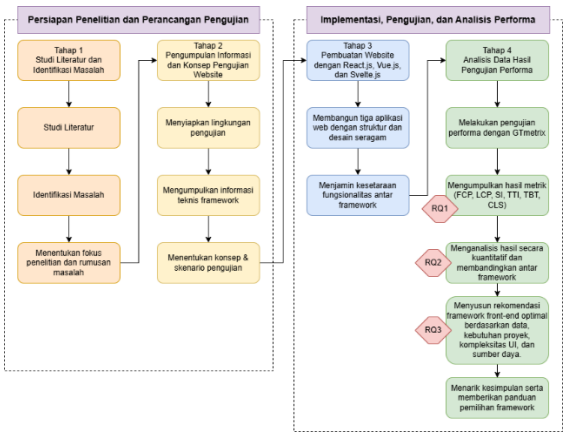
2.2. Pengujian Performa Menggunakan GTmetrix

GTmetrix merupakan *tool* analisis performa *website* yang digunakan untuk mengukur kinerja teknis suatu aplikasi web secara terukur, mencakup indikator seperti *performance grade*, *page load time*, *total page size*, jumlah permintaan (*total requests*), serta *optimization score* yang mencerminkan efisiensi pengelolaan kode dan aset web (Marsa et al., 2025). Data performa dikumpulkan melalui pengujian langsung terhadap ketiga aplikasi menggunakan *tool* ini. Setiap halaman aplikasi diuji sebanyak 15 kali pengulangan untuk memperoleh nilai rata-rata tiap metrik yang representatif dan mengurangi bias akibat variasi kondisi jaringan atau server pengujian. Skenario pengujian mencakup pemuatan awal halaman melalui akses URL langsung serta proses *rendering* data dinamis pada halaman daftar tugas berisi 100 *item*. Variabel kontrol dijaga

konsistensinya, meliputi jenis *browser* (Google Chrome), spesifikasi perangkat keras (laptop Asus Vivobook, prosesor Intel Core i3, RAM 8 GB), konfigurasi jaringan pengujian melalui platform GTmetrix, struktur dan fitur aplikasi, serta lingkungan *deployment* (VPS dan *web server* Nginx yang sama).

2.3. Alur Penelitian dan Variabel Uji

Penelitian dilaksanakan melalui tahapan sistematis sebagaimana ditunjukkan pada Gambar 1.



Gambar 1. Alur Penelitian

Tahapan diawali dari studi literatur untuk membangun pemahaman mengenai performa *framework front-end* dan memetakan penelitian terdahulu yang relevan, dilanjutkan dengan identifikasi masalah berdasarkan kesenjangan yang ditemukan pada studi literatur. Tahap berikutnya adalah perancangan skenario pengujian, yaitu penyusunan kerangka aplikasi, penentuan fitur yang diimplementasikan, serta penetapan metrik performa evaluasi. Selanjutnya, tiga aplikasi web dibangun menggunakan React.js, Vue.js, dan Svelte.js secara terpisah dengan desain, struktur halaman, dan data uji yang identik, kemudian di-*build* dalam mode *production* dan diunggah sebagai berkas statis ke *hosting* yang sama. Tahap pengujian performa dilakukan menggunakan GTmetrix, diikuti analisis data secara kuantitatif melalui perhitungan nilai rata-rata, standar deviasi, dan pembobotan skor untuk menentukan *framework* dengan kinerja paling optimal, dan diakhiri dengan penarikan kesimpulan sebagai dasar rekomendasi pemilihan *framework front-end* bagi pengembang. Variabel independen dalam penelitian ini adalah *framework* JavaScript yang diuji (React.js, Vue.js, Svelte.js), sedangkan variabel dependen adalah performa aplikasi web yang diukur melalui enam metrik dari hasil pengujian GTmetrix.

2.4. Metrik dan Sistem Penilaian Performa Web

Evaluasi performa menggunakan enam metrik utama yang secara umum digunakan untuk mengukur kecepatan pemuatan dan interaktivitas halaman web. *First Contentful Paint* (FCP) mengukur waktu yang dibutuhkan *browser* untuk mulai menampilkan konten pertama pada halaman, seperti teks atau gambar, setelah proses pemuatan dimulai (Widi et al., 2024). *Largest Contentful Paint* (LCP) mengukur waktu hingga elemen konten terbesar yang terlihat di *viewport* seperti gambar utama atau blok teks besar selesai dirender, mencerminkan kecepatan pemuatan konten inti dari sudut pandang pengguna (Pandya et al., 2024). *Speed Index* (SI) menunjukkan seberapa cepat tampilan visual halaman terbentuk secara keseluruhan selama proses pemuatan berlangsung, dengan nilai yang semakin kecil menandakan halaman terasa lebih cepat termuat (Tengriano et al., 2022).

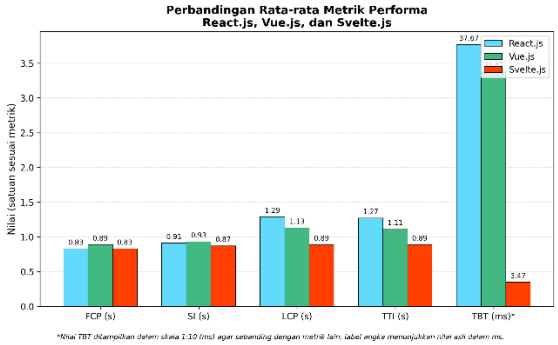
Time to Interactive (TTI) menunjukkan durasi hingga halaman sepenuhnya siap digunakan, yaitu ketika seluruh konten telah tampil dan pengguna dapat berinteraksi tanpa mengalami jeda (Pandya et al., 2024). *Total Blocking Time* (TBT) mengukur akumulasi durasi ketika *thread* utama *browser* terhambat oleh eksekusi JavaScript yang memakan waktu lama, sehingga menghalangi respons terhadap *input* pengguna, semakin besar nilai TBT, semakin buruk kualitas interaksi yang dirasakan (Hidayati, 2022). Terakhir, *Cumulative Layout Shift* (CLS) mengukur sejauh mana elemen-elemen halaman mengalami pergeseran posisi secara tak terduga selama pemuatan, di mana nilai yang tinggi mengindikasikan pengalaman pengguna yang kurang baik akibat tampilan yang tidak stabil (Widi et al., 2024).

Setiap metrik diberi skor pada rentang 0–3 berdasarkan *threshold* GTmetrix: skor 3 untuk kategori "*Good*", skor 2 untuk "*OK, but consider improvement*", skor 1 untuk "*Longer than recommended*", dan skor 0 untuk "*Much longer than recommended*". Skor maksimal yang dapat diperoleh satu halaman aplikasi adalah 18 poin, yaitu akumulasi skor tertinggi dari keenam metrik yang diuji. Sistem penilaian ini diterapkan agar hasil evaluasi performa antar-*framework* dapat dibandingkan secara kuantitatif dan objektif.

3. PEMBAHASAN

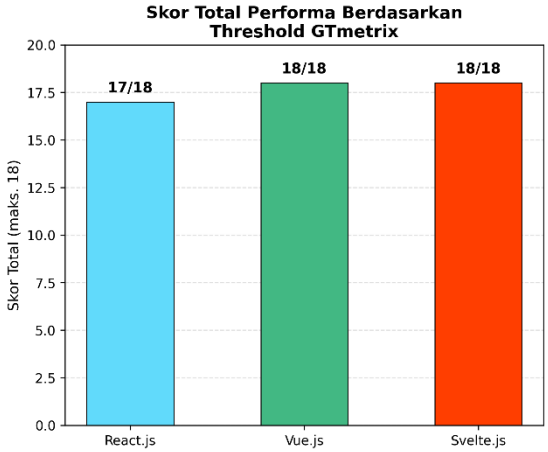
3.1. Perbandingan Metrik Performa React.js, Vue.js, dan Svelte.js

Penelitian ini berfokus pada perbandingan performa tiga *framework front-end* JavaScript yaitu React.js, Vue.js, dan Svelte.js dalam aspek kecepatan pemuatan halaman, interaktivitas, dan stabilitas tata letak. Pengujian dilakukan menggunakan GTmetrix dengan enam metrik (FCP, LCP, SI, TTI, TBT, dan CLS), masing-masing yaitu sebanyak 15 kali pengulangan untuk memperoleh nilai rata-rata yang representatif.



Gambar 2. Perbandingan rata-rata metrik performa React.js, Vue.js, dan Svelte.js

Berdasarkan Gambar 2, Svelte.js secara konsisten mencatat nilai terendah pada hampir seluruh metrik, terutama TBT yang hanya sebesar 3,47 ms, jauh di bawah Vue.js (33,87 ms) dan React.js (37,67 ms). React.js menunjukkan nilai tertinggi pada metrik LCP dan TTI, mengindikasikan waktu pemuatan konten dan interaktivitas yang relatif lebih lambat.



Gambar 3. Skor total performa React.js, Vue.js, dan Svelte.js berdasarkan threshold GTmetrix

Gambar 3 menunjukkan Vue.js dan Svelte.js memperoleh skor sempurna (18/18), sedangkan React.js memperoleh skor 17/18 akibat nilai LCP yang berada pada kategori "*OK, but consider improvement*". Rincian nilai tiap metrik ditampilkan pada Tabel 1.

Tabel 1. Perbandingan Performa React.js, Vue.js, dan Svelte.js

No.	Metrik	React	Vue	Svelte
1	FCP (s)	0.83	0.89	0.83
2	SI (s)	0.91	0.93	0.87
3	LCP (s)	1.29	1.13	0.89
4	TTI (s)	1.27	1.11	0.89
5	TBT (ms)	37.67	33.87	3.47
6	CLS	0.03	0.03	0.03
7	Skor Total	17	18	18

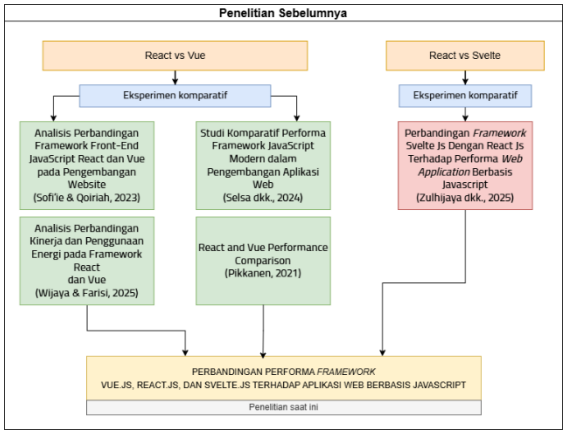
Pada metrik CLS, ketiga *framework* menunjukkan nilai identik (0,03) dan seluruhnya berada pada kategori "*Good*", mengindikasikan stabilitas tata letak halaman yang konsisten pada seluruh aplikasi uji.

3.2. Pengaruh Arsitektur Rendering terhadap Performa Framework

Perbedaan performa antara React.js, Vue.js, dan Svelte.js dipengaruhi oleh pendekatan arsitektur masing-masing *framework*. React.js dan Vue.js menggunakan *virtual DOM* yang memerlukan proses *diffing* sebelum perubahan ditampilkan ke halaman, sehingga menambah *overhead* pemrosesan saat aplikasi dijalankan. Sebaliknya, Svelte.js menerapkan pendekatan *compile-time* tanpa *virtual DOM*, di mana sebagian besar proses optimasi dilakukan pada tahap *build* aplikasi. Pendekatan ini menghasilkan ukuran *bundle* dan proses *rendering* yang lebih ringan saat aplikasi dijalankan di *browser*, sehingga menjelaskan keunggulan Svelte.js pada metrik SI, LCP, TTI, dan TBT. Pada sisi lain, React.js memiliki waktu eksekusi yang relatif lebih tinggi pada beberapa metrik karena ukuran *library* yang lebih besar, ketergantungan pada *virtual DOM*, serta kebutuhan *library* tambahan untuk mendukung berbagai fitur aplikasi, meskipun tetap berada dalam kategori performa yang baik berdasarkan *threshold* GTmetrix.

3.3. Posisi Temuan terhadap Penelitian Terdahulu

Posisi penelitian ini dalam mengisi kesenjangan (*research gap*) pada literatur dapat dikelompokkan menjadi dua jalur penelitian sebelumnya perbandingan React.js vs Vue.js dan React.js vs Svelte.js sebagaimana terlihat pada Gambar 4.



Gambar 4. Posisi Penelitian

Temuan Svelte.js pada penelitian ini menunjukkan keunggulan performa yang konsisten dengan temuan (Zulhijaya et al., 2025), yang membandingkan Svelte.js dengan React.js menggunakan GTmetrix dan menyimpulkan Svelte.js unggul pada sebagian besar parameter yang diuji. Namun, penelitian tersebut hanya membandingkan dua *framework* dan terbatas pada tiga halaman sederhana, sehingga belum menggambarkan posisi Svelte.js relatif terhadap Vue.js. Penelitian ini melengkapi keterbatasan tersebut dengan menyertakan Vue.js sebagai pembanding ketiga secara simultan, menggunakan metrik dan alat ukur yang sama untuk seluruh *framework*.

Pada sisi Vue.js, hasil skor sempurna (18/18) yang diperoleh dalam penelitian ini sejalan dengan temuan (Sofi'ie & Qoiriah, 2023) dan (Wijaya & Farisi, 2025), yang masing-masing menyatakan Vue.js unggul dalam performa dan kecepatan pemuatan halaman dibandingkan React.js. Kedua penelitian tersebut menggunakan kombinasi alat ukur berbeda *js-framework-benchmark*, Google Lighthouse, dan IBM SPSS 27 pada studi pertama, serta Lighthouse, Google DevTools, dan Globalmallow pada studi kedua namun keduanya juga tidak menyertakan Svelte.js sebagai pembanding, sehingga belum dapat memastikan apakah keunggulan Vue.js tetap konsisten ketika dihadapkan dengan *framework compile-time* seperti Svelte.js. Temuan penelitian ini menunjukkan bahwa meskipun Vue.js unggul dibandingkan React.js, posisinya tetap berada di bawah Svelte.js pada mayoritas metrik, khususnya TBT.

Sementara itu, performa React.js yang relatif lebih rendah pada penelitian ini turut didukung oleh temuan (Selsa et al., 2024), yang menyatakan Vue.js lebih cepat dan ringan dibandingkan React.js, meskipun React.js dinilai lebih unggul dalam pembaruan data dinamis dan manajemen *state*. Studi tersebut turut mencatat keterbatasan berupa belum disertakannya Svelte.js sebagai *framework* pembanding gap yang sama, yang coba dijawab oleh penelitian ini. Berbeda dengan hasil tersebut, penelitian (Pikkanen, 2021) justru menemukan bahwa React sedikit lebih cepat dibandingkan Vue pada *ranking test*, meski perbedaannya dinilai kecil dan tidak signifikan bagi pengguna. Perbedaan arah temuan ini kemungkinan dipengaruhi oleh perbedaan jenis aplikasi uji dan alat ukur yang digunakan (Lighthouse, Performance API, dan Puppeteer), yang menegaskan pentingnya standar pengukuran yang seragam seperti diterapkan pada penelitian ini agar hasil antar-*framework* dapat dibandingkan secara adil.

Secara umum, kelima penelitian terdahulu di atas memperlihatkan pola yang sama: perbandingan dilakukan secara berpasangan (dua *framework*) dan menggunakan alat ukur yang bervariasi antar studi, sehingga sulit menyimpulkan posisi ketiga *framework* secara bersamaan dalam satu standar pengukuran. Penelitian ini menjawab kesenjangan tersebut dengan mengevaluasi React.js, Vue.js, dan Svelte.js secara simultan menggunakan GTmetrix sebagai satu-satunya alat ukur, sehingga hasil perbandingan antar ketiganya lebih dapat dipertanggungjawabkan secara metodologis.

4. KESIMPULAN

Penelitian ini berhasil menjawab tujuan yang ditetapkan, yaitu melakukan analisis komparatif performa React.js, Vue.js, dan Svelte.js secara simultan menggunakan metrik yang seragam (FCP, LCP, SI, TTI, TBT, dan CLS) melalui pengujian GTmetrix. Hasil menunjukkan Svelte.js memiliki performa paling optimal, ditandai dengan nilai TBT yang jauh lebih rendah (3,47 ms) dibandingkan Vue.js (33,87 ms) dan React.js (37,67 ms), sejalan dengan pendekatan *compile-time* yang meniadakan kebutuhan *virtual DOM* saat *runtime*. Vue.js memperoleh skor sempurna (18/18) dengan performa yang stabil dan seimbang antara kecepatan dan kemudahan pengembangan, sedangkan React.js memperoleh skor sedikit lebih rendah (17/18) akibat kompleksitas *rendering*, meskipun tetap didukung ekosistem yang luas dan fleksibilitas tinggi. Dengan demikian, Svelte.js direkomendasikan untuk proyek yang mengutamakan performa, Vue.js untuk keseimbangan performa dan kemudahan pengembangan, sedangkan React.js untuk proyek berskala besar yang membutuhkan dukungan ekosistem luas. Penelitian selanjutnya disarankan menguji ketiga *framework* pada aplikasi berskala besar dengan manajemen *state* yang lebih kompleks, serta menambahkan aspek pengukuran lain seperti konsumsi memori dan ukuran *bundle* untuk memperoleh gambaran yang lebih komprehensif.

PUSTAKA

Arya, K., Wirya, B., Yudi, I. N., Wijaya, A., Juliana, I. G., & Putra, E. (2024). *Implementasi Next . Js , Typescript , Dan Tailwind Css Untuk Pengembangan Aplikasi Frontend Sistem Inventory Perusahaan Apar (Studi Kasus : CV Indoka Surya Jaya) Implementation Next . Js , Typescript , And Tailwind Css For The Development Of Apar Company Inventory System Frontend Application (Case Study : Cv Indoka Surya Jaya)*. *14*(2), 95–108.

Dubaj, S., & Pańczyk, B. (2022). Comparative of React and Svelte programming frameworks for creating SPA web applications. *Journal of Computer Sciences Institute*, *25*(June), 345–349. https://doi.org/10.35784/jcsi.3020

Hidayati, N. H. (2022). *ANALISIS PERFORMA WEBSITE KANTOR PENCARIAN DAN PERTOLONGAN PALEMBANG MENGGUNAKAN GTMETRIX*.

Iriyadi, D. (2024). Telaah Kritis Metode-Metode Dalam Penelitian Ilmiah. *Qouba Jurnal Pendidikan*, *1*, 22–28.

Iswari, N. M. S., & Dharma, E. M. (2025). Pelatihan Pengembangan Aplikasi Web Menggunakan Laravel Untuk. *Jurnal Pengabdian Masyarakat*, *5*(2), 719–730.

Krisna, B., Saifudin, I., & Muharom, L. A. (2024). *Analisis Performa Aplikasi Portal Portofolio Virtual Reality Berbasis Website Dengan Apache Benchmark Di PT Telekomunikasi Indonesia*. 34–45.

Marsa, A. R., Salam, R. I., & Manda, P. (2025). *Evaluasi Kinerja Aplikasi E-Alumni Berbasis Web menggunakan Metode PIECES dan GTmetrix*. *5*(2), 359–364.

Martha, G., Wijaya, I. N. Y. A., & Utami, N. W. (2025). *Rancang Bangun Company Profile Berbasis Content Management System*

Menggunakan Framework Next . Js Pada Satelit NET Komputer Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer). *24*, 152–161.

Ngurah, I. G., Prawirayuda, A., Estiyanti, N. M., & Dharma, E. M. (2022). Model Aplikasi Sistem Simpan Pinjam Berbasis Mobile Web Pada Usaha Koperasi. *Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi*, *11*(1), 153–164.

Octaviani, A., & Andraini, R. (2022). Analisis Website Fakultas Teknik Universitas Negeri Jakarta Menggunakan Pingdom Tools dan GTmetrix. *SINTESIA: Jurnal Sistem Dan Teknologi Informasi Indonesia*, *1*(2), 76–80. https://tools.pingdom.com/

Pandya, D. N., Suryadharma, D., Wulandhari, L. A., & Alam, I. N. (2024). *Assessing University Website Performance : A Comparative Analysis Using GTmetrix*. *1*(1), 33–38. https://doi.org/10.21512/ijcshaijournal.v1i1.12152

Pikkanen, M. (2021). React and Vue performance comparison. *Metropolia University of Applied Sciences*, *June*, 31. https://www.theseus.fi/bitstream/handle/10024/501354/Pikkanen_Markus.pdf

Selsa, O., Anggraeni, I., & Informatika, P. T. (2024). Studi Komparatif Performa Framework Javascript Modern dalam Pengembangan Aplikasi Web. *Jurnal Informatika Dan Sains Teknologi*, *2*(4), 162–177.

Sofi'ie, F. A. F., & Qoiriah, A. (2023). Analisis Perbandingan Framework Front-End Javascript React dan Vue Pada Pengembangan Website. *Journal of Informatics and Computer Science (JINACS)*, *5*(02), 157–164. https://doi.org/10.26740/jinacs.v5n02.p157-164

Tengriano, H. A., Yunus, A., & Sudirman. (2022). ANALISIS PERFORMA WEBSITE AYOMULAI MENGGUNAKAN GTMETRIX DAN PAGESPEED INSIGHT. *Jurnal KHARISMA Tech*, *17*(02), 199–213.

Widi, A., Sediyono, E., & Hendry. (2024). *Analisa Performa Website Organisasi Akuatik Menggunakan Automated Software Testing GTmetrix*. *5*(September), 25–33. https://doi.org/10.35957/jtsi.v5i2.5925

Wijaya, I., & Farisi, A. (2025). Analisis Perbandingan Kinerja dan Penggunaan Energi pada Framework React dan Vue. *Techno.Com*, *24*(1), 104–116. https://doi.org/10.62411/tc.v24i1.12092

Zulhijaya, Mustari S Lamada, & Abdul Wahid. (2025). Perbandingan Framework Svelte Js Dengan React Js Terhadap Performa Web Application Berbasis Javascript. *Jurnal Riset Teknik Komputer*, *2*(2), 66–81. https://doi.org/10.69714/f2h1kn88