

Judul: Analisis Performa *Loading Time* dan SEO Website Menggunakan *Server-Side Rendering (SSR)* Next.js Dibandingkan *Client-Side Rendering (CSR)*

BAB I: PENDAHULUAN

1.1 Latar Belakang Masalah

Studi industri secara konsisten menunjukkan bahwa keterlambatan sepersekian detik saja dapat menyebabkan pengguna meninggalkan situs (Gallino et al., 2023). Metrik yang diperkenalkan Google, seperti *Core Web Vitals (CWV)*, kini menjadi standar industri untuk mengukur kesehatan dan performa website, yang secara langsung memengaruhi peringkat di mesin pencari (Król & Sroka, 2023). Website yang lambat tidak hanya memberikan pengalaman buruk bagi pengguna, tetapi juga akan mendapatkan penalti peringkat dari Google.

Perkembangan *framework* JavaScript modern seperti React dan Vue telah membawa revolusi dalam interaktivitas web. *Framework* ini mempopulerkan arsitektur *Client-Side Rendering (CSR)*, di mana server hanya mengirimkan satu file HTML minimalis beserta file JavaScript. Seluruh proses *rendering*, pengambilan data, dan perakitan halaman kemudian dieksekusi oleh *browser* pengguna (*client*) (Ardiyanto & Ardianto, 2024). Pendekatan ini berhasil menciptakan nuansa aplikasi yang sangat responsif setelah halaman berhasil dimuat.

Meskipun unggul dalam interaktivitas *post-load*, model CSR memiliki dua kelemahan fundamental. Pertama, performa *loading* awal (*initial load*) yang cenderung lambat. Pengguna seringkali dihadapkan pada layar putih (*blank screen*) selagi menunggu *browser* mengunduh dan mengeksekusi paket JavaScript yang berukuran besar, yang berdampak buruk pada metrik *First Contentful Paint (FCP)* (Han et al., 2014). Kedua, model CSR memiliki tantangan signifikan dalam hal SEO. *Crawler* mesin pencari yang pada dasarnya membaca HTML, akan kesulitan mengindeks konten yang belum di-*render* oleh JavaScript. Walaupun *crawler* modern mulai bisa mengeksekusi JavaScript, prosesnya tidak efisien dan sering gagal pada aplikasi yang kompleks (Kowalczyk & Szandala, 2024).

Untuk mengatasi kelemahan tersebut, teknik *Server-Side Rendering (SSR)* kembali populer. Pada model SSR, server mengambil data dan merakit halaman HTML secara penuh sebelum mengirimkannya ke *browser*. Hal ini memungkinkan *browser* untuk segera menampilkan konten, sehingga secara drastis memperbaiki FCP dan menyediakan konten HTML yang lengkap untuk diindeks oleh *bot* mesin pencari (Han et al., 2014), ("Analyzing the Impact of Next.JS on Site Performance and SEO," 2023). Next.js, sebuah *framework* React, hadir sebagai salah satu solusi terdepan yang mengimplementasikan SSR (serta *Static Site Generation/SSG*) secara efisien untuk mengatasi masalah performa dan SEO yang dihadapi oleh aplikasi React standar (Thakkar, 2020).

Meskipun SSR secara teoretis menawarkan keunggulan dalam *loading time* awal dan SEO, perlu dilakukan analisis kuantitatif untuk membuktikan dampaknya secara nyata. Terdapat *trade-off* seperti potensi peningkatan beban server pada SSR yang perlu dipertimbangkan (Ardiyanto & Ardhiyanto, 2024). Oleh karena itu, penelitian ini bertujuan untuk melakukan analisis perbandingan performa *loading time* (diukur dengan *Core Web Vitals*) dan efektivitas SEO antara aplikasi web yang dibangun menggunakan SSR (dengan Next.js) dan aplikasi web yang menggunakan CSR murni (React standar).

BAB X: PENUTUP

1.1 Daftar Pustaka

Analyzing the Impact of Next.JS on Site Performance and SEO. (2023). *International Journal of Computer Applications Technology and Research*.

<https://doi.org/10.7753/IJCATR1210.1004>

Ardiyanto, R., & Ardhiyanto, E. (2024). Analisa Performasi Metode Client Side Rendering, Server Side Rendering, dan Incremental Static Regeneration dalam Proses Website Rendering. *Computer Science (CO-SCIENCE)*, 4(1), 19–27.

<https://doi.org/10.31294/coscience.v4i1.2427>

Gallino, S., Karacaoglu, N., & Moreno, A. (2023). Need for Speed: The Impact of In-Process Delays on Customer Behavior in Online Retail. *Operations Research*, 71(3), 876–894.

<https://doi.org/10.1287/opre.2022.2262>

Han, H., Xue, Y., Oyama, K., & Liu, Y. (2014). Practice and Evaluation of Pagelet-Based Client-Side Rendering Mechanism. *IEICE Transactions on Information and Systems*, E97.D(8), 2067–2083. <https://doi.org/10.1587/transinf.E97.D.2067>

Kowalczyk, K., & Szandala, T. (2024). Enhancing SEO in Single-Page Web Applications in Contrast With Multi-Page Applications. *IEEE Access*, 12, 11597–11614.

<https://doi.org/10.1109/ACCESS.2024.3355740>

Król, K., & Sroka, W. (2023). Internet in the Middle of Nowhere: Performance of Geoportals in Rural Areas According to Core Web Vitals. *ISPRS International Journal of Geo-Information*, 12(12), 484. <https://doi.org/10.3390/ijgi12120484>

Thakkar, M. (2020). *Building React Apps with Server-Side Rendering: Use React, Redux, and Next to Build Full Server-Side Rendering Applications*. Apress.
<https://doi.org/10.1007/978-1-4842-5869-9>