

Implementasi Server Side Rendering Pada Sistem Travel Berbasis Website

Handie Pramana Putra¹, Anggraini Puspita Sari^{2*}

^{1,2} Magister Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur

123066020008@student.upnjatim.ac.id

2anggraini.puspita.if@upnjatim.ac.id

*Corresponding author email: anggraini.puspita.if@upnjatim.ac.id

Abstrak— Perkembangan teknologi informasi telah mengubah pengembangan website dari statis menjadi lebih kompleks dan interaktif. Server-side Rendering (SSR) adalah teknologi yang merender konten di sisi server sebelum mengirimkannya ke klien, meningkatkan kecepatan muat halaman dan performa website. Penelitian ini mengimplementasikan SSR pada sistem travel berbasis website menggunakan Next.js untuk meningkatkan performa dan waktu respons. Metode penelitian yang digunakan adalah pendekatan kuantitatif, dengan mengumpulkan data dari jurnal dan buku melalui Google Scholar, Springer, dan situs jurnal lainnya. Implementasi SSR melibatkan konfigurasi awal, penggunaan `getServerSideProps`, pengoptimalan kinerja, dan penerapan SEO. Pengujian performa dilakukan menggunakan Lighthouse, dan browser. Hasil penelitian menunjukkan bahwa SSR meningkatkan kecepatan muat halaman pertama dan optimasi mesin pencari (SEO). Teknik caching di sisi server juga mengurangi beban server dan meningkatkan efisiensi. Kesimpulannya, SSR tidak hanya meningkatkan performa dan waktu respons, tetapi juga kualitas pengalaman pengguna. Penelitian ini memberikan praktik praktis bagi pengembang web dalam mengadopsi SSR untuk meningkatkan performa dan kualitas layanan website. Saran untuk pengembangan lebih lanjut mencakup strategi caching yang lebih baik, monitoring performa, peningkatan keamanan, pengembangan fitur tambahan, dan pembaruan berkala.

Kata Kunci— Server side rendering, Cloud computing, caching, website, travel

I. PENDAHULUAN

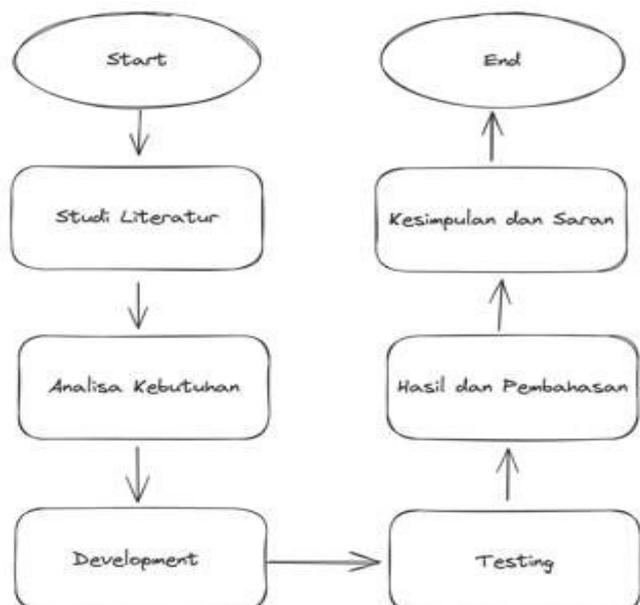
Perkembangan teknologi informasi dalam pengembangan website telah mengalami kemajuan signifikan, dari web statis yang sederhana hingga website yang lebih kompleks dan interaktif. Dengan semakin tingginya ekspektasi pengguna terhadap kecepatan dan performa website, teknologi rendering menjadi salah satu aspek yang sangat penting dalam pengembangan website modern. Salah satu teknologi yang telah terbukti mampu meningkatkan performa dan kecepatan muat halaman website adalah Server-side Rendering (SSR). Server-side rendering (SSR) merupakan teknologi rendering yang memungkinkan konten halaman website dirender di sisi server sebelum dikirimkan ke klien. Dengan SSR, halaman website dapat ditampilkan lebih cepat kepada pengguna karena proses rendering telah dilakukan sebelumnya di server. Ini berbeda dengan Client-side Rendering (CSR) yang memerlukan browser untuk merender konten setelah halaman dimuat, yang dapat menyebabkan waktu muat yang lebih lambat dan pengalaman pengguna yang kurang optimal. Implementasi SSR pada sistem travel berbasis website bertujuan untuk meningkatkan performa dan response time saat merender halaman website. Dengan SSR, pengguna dapat

merasakan peningkatan kecepatan dalam mengakses halaman-halaman website, yang sangat penting dalam industri travel yang memerlukan interaksi cepat dan responsif. Kecepatan muat halaman yang tinggi dapat meningkatkan kenyamanan pengguna, yang pada gilirannya dapat meningkatkan kepuasan dan loyalitas pelanggan. Tujuan penelitian ini tidak hanya berkontribusi pada perkembangan teknologi informasi, tetapi juga pada peningkatan kualitas layanan dalam industri travel. Dengan penerapan SSR, diharapkan sistem travel berbasis website dapat memberikan pengalaman pengguna yang lebih baik dan efisien, yang pada akhirnya dapat mendukung pertumbuhan dan perkembangan industri travel secara keseluruhan. Selain itu, penelitian ini juga bertujuan untuk memberikan panduan praktis bagi pengembang website dalam menerapkan SSR, sehingga dapat meningkatkan performa dan kualitas website mereka.

Penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam bidang teknologi informasi dan industri travel, serta menjadi referensi bagi penelitian-penelitian selanjutnya dalam implementasi teknologi rendering pada berbagai jenis website.

II. METODOLOGI

Jenis metode penelitian yang diadopsi adalah pendekatan kuantitatif untuk mengimplementasikan Server-side Rendering pada sistem travel. Tujuan dari pendekatan ini adalah untuk meningkatkan performa dan waktu respons saat merender halaman website.



Gambar. 1 Metodologi Penelitian

A. Studi Literatur

Pada tahap ini, informasi yang diperlukan akan dikumpulkan dari jurnal dan buku melalui Google Scholar, Springer, dan situs jurnal lainnya. Fokus utama adalah teknologi Server-side Rendering pada framework Next.js.

B. Analisa Kebutuhan

Pada tahap ini, data dan informasi akan dikumpulkan dari berbagai jurnal dan buku untuk memungkinkan peneliti melakukan analisis kebutuhan, baik fungsional maupun non-fungsional. Kebutuhan fungsional untuk sistem absensi berbasis website adalah sebagai berikut:

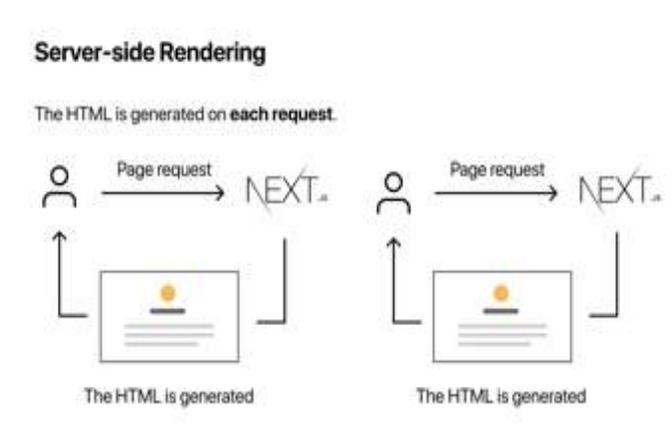
- Sistem dapat digunakan untuk melihat informasi tour
- Sistem dapat menampilkan informasi data detail tour
- Sistem dapat digunakan untuk melihat artikel mengenai tour

Sedangkan untuk kebutuhan non-fungsional sebagai berikut:

- Sistem mudah digunakan dengan antarmuka navigasi yang baik dan responsive
- Sistem mudah diakses dengan berbagai tipe jenis peramban dan perangkat yang berbeda dan telah terhubung ke internet.

C. Development

Pada tahap ini, Sistem Travel akan menggunakan metode server side rendering untuk melihat dan mengukur kecepatan load time respon saat merender halaman dengan data yang dikirimkan oleh server.



Gambar. 2 Server Side Rendering

Server-side Rendering (SSR) pada Next.js adalah teknik di mana halaman web dirender di server sebelum dikirim ke klien. Dalam konteks Next.js, ini berarti bahwa saat permintaan dibuat ke server, server menjalankan kode JavaScript untuk menghasilkan HTML yang sudah lengkap dengan data yang diperlukan, lalu mengirimkan HTML tersebut ke browser. Hal ini berbeda dengan Client-side Rendering (CSR), di mana HTML dasar dikirim ke browser dan JavaScript di browser yang bertanggung jawab untuk merender halaman.

D. Testing

Pada tahap ini aplikasi akan diuji dengan mengecek hasil performance, dan memory application pada browser, dan pada server saat eksekusi dilakukan. Selain itu juga menggunakan aplikasi Google Lighthouse untuk mendapatkan hasil yang detail lainnya.

III. HASIL DAN PEMBAHASAN

Berikut hasil dari penelitian atau hasil dari yang diamati atau ditemukan ketika melakukan penelitian.

A. Implementasi SSR

Proses implementasi SSR pada system travel berbasis website dilakukan menggunakan framework Next.js. Karena framework tersebut memiliki *lifecycle* yang baik untuk proses pengembangan aplikasi. Terdapat beberapa tahapan sebagai berikut:

```

***
import type { InferGetServerSidePropsType, GetServerSideProps } from 'next'

type Repo = {
  name: string
  stargazers_count: number
}

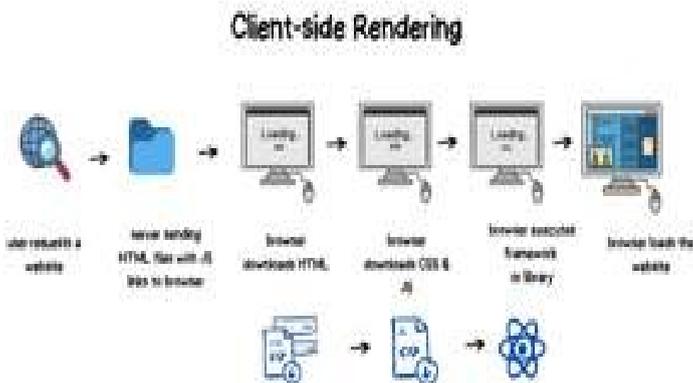
export const getServerSideProps = async () => {
  // Fetch data from external API
  const res: Response = await fetch('https://api.github.com/repos/nextjs/next.js')
  const repo: Repo = await res.json()
  // Pass data to the page via props
  return { props: { repo } }
} satisfies GetServerSideProps<{ repo: Repo }>

export default function Page({ no usage
  repo,
}: InferGetServerSidePropsType<typeof getServerSideProps>) {
  return (
    <main>
      <p>{repo.stargazers_count}</p>
    </main>
  )
}

```

Gambar. 4 Implementasi Server Side Rendering

- Konfigurasi project dan dependensi yang diperlukan.
- Menggunakan fungsi 'getServerSideProps' untuk mengambil data dari server sebelum halaman di render
- Integrasi API untuk mendapatkan data yang diperlukan
- Implementasi caching di sisi server untuk meningkatkan kecepatan response.
- Penggunaan teknik lazy loading untuk memuat gambar dan konten lainnya secara dinamis.
- Konfigurasi meta tags dan Open Graph tags untuk meningkatkan SEO.



Gambar. 3 Client Side Rendering

B. Pengujian performa

Untuk mengukur performa dan response time dari system travel berbasis website dengan SSR, di uji menggunakan Google Lighthouse dengan hasil:

- Pengujian dilakukan untuk mengukur performa, aksesibilitas, dan SEO dari halaman yang dirender menggunakan SSR.
- Hasil menunjukkan peningkatan signifikan pada waktu muat halaman pertama dibandingkan dengan pendekatan Client-side Rendering (CSR).

C. Analisa Hasil

Berdasarkan hasil pengujian performa yang dilakukan, beberapa temuan penting dapat diidentifikasi sebagai berikut:

1) Kecepatan halaman muat



Gambar. 5 Hasil pengujian sebelum menggunakan SSR

Hasil dengan menggunakan CSR dengan performa 77 menggunakan Lighthouse.



Gbr. 6 Hasil pengujian sesudah menggunakan SSR

Hasil dengan SSR mendapatkan nilai 97 untuk perbaikan performa menggunakan Lighthouse.

- Secara performa mengalami kenaikan signifikan yang di awal angka 77 menjadi 97 saat menggunakan SSR.
- Implementasi SSR menghasilkan waktu muat halaman pertama yang lebih cepat, memberikan pengalaman pengguna yang lebih baik.
- Pengguna dapat melihat konten lebih cepat, yang dapat meningkatkan tingkat retensi pengguna dan mengurangi bounce rate.

2) SEO yang lebih baik

- Halaman yang dirender dengan SSR lebih mudah diindeks oleh mesin pencari, meningkatkan visibilitas website pada hasil pencarian.
- Peningkatan SEO ini dapat berdampak positif pada jumlah kunjungan organik ke website.

3) Efisiensi server

- Penggunaan caching di sisi server membantu mengurangi beban pada server, meningkatkan efisiensi dan skalabilitas sistem.
- Kombinasi SSR dan teknik caching memastikan bahwa halaman tetap responsif meskipun ada lonjakan trafik.

D. Pembahasan

Implementasi Server-side Rendering pada sistem travel berbasis website telah menunjukkan beberapa manfaat

signifikan. Kecepatan muat halaman yang lebih cepat dan peningkatan SEO adalah dua keuntungan utama yang dapat meningkatkan pengalaman pengguna dan visibilitas website.

Namun, ada beberapa tantangan yang perlu diperhatikan, seperti peningkatan kompleksitas dalam pengembangan dan kebutuhan untuk pengelolaan server yang lebih baik.

Penggunaan SSR juga memerlukan pertimbangan terhadap caching strategi untuk memastikan performa yang konsisten.

Secara keseluruhan, implementasi SSR pada sistem travel berbasis website tidak hanya memberikan peningkatan pada performa dan response time tetapi juga meningkatkan kualitas layanan yang diberikan kepada pengguna. Penelitian ini memberikan wawasan berharga bagi pengembang web dalam mengadopsi teknologi SSR untuk aplikasi mereka, terutama dalam industri yang sangat bergantung pada kecepatan dan performa seperti industri travel.

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Dari hasil penelitian dan implementasi Server-side Rendering (SSR) pada sistem travel berbasis website, terdapat beberapa kesimpulan yang dapat diambil yaitu. Implementasi SSR terbukti meningkatkan performa halaman website, dengan waktu muat halaman pertama yang lebih cepat dibandingkan dengan pendekatan Client-side Rendering (CSR). Pengguna dapat mengakses konten dengan lebih cepat, yang meningkatkan pengalaman pengguna secara keseluruhan. SSR membantu meningkatkan optimasi mesin pencari (SEO) dengan menyediakan HTML yang sudah dirender lengkap dengan konten yang dapat diindeks oleh mesin pencari. Ini meningkatkan visibilitas website pada hasil pencarian dan berpotensi meningkatkan jumlah kunjungan. Penggunaan teknik caching di sisi server bersama dengan SSR mengurangi beban server dan meningkatkan efisiensi sistem. Ini memastikan bahwa halaman tetap responsif bahkan selama lonjakan trafik. Dengan SSR, pengguna merasakan pengalaman yang lebih cepat dan responsif, yang dapat meningkatkan kepuasan dan loyalitas pengguna terhadap layanan travel yang ditawarkan.

B. Saran

Berdasarkan hasil penelitian, berikut beberapa saran yang dapat dipertimbangkan untuk pengembangan lebih lanjut:

- 1) Penerapan caching yang lebih baik: Implementasikan strategi caching yang lebih canggih untuk meningkatkan efisiensi dan performa.
- 2) Monitoring dan analisis performa: Terus melakukan monitoring dan analisis performa, karena seiring berjalannya waktu dan jumlah data yang semakin berkembang akan berpengaruh pada cache.
- 3) Peningkatan keamanan: Melakukan implementasi fitur keamanan yang lebih banyak untuk menjamin data dan pengalaman pengguna.
- 4) Evaluasi dan Pembaruan berkala: Melakukan evaluasi dan pembaruan berkala terhadap teknologi dan metode yang digunakan untuk memastikan bahwa sistem tetap up-to-date dengan perkembangan terbaru dalam teknologi web dan kebutuhan pengguna.

REFERENSI

- [1] Sampath, S, & Borkar, V 2020, 'Optimizing web applications using a rendering engine', US Patent 10,567,537, Google Patents.
- [2] Posin, D 2021, 'Browser-based cloud gaming', US Patent 11,130,054, Google Patents.
- [3] Greenebaum, KI, Guo, H, Pan, H, Cote, G, & ... 2020, 'Server-side adaptive video processing', US Patent 10,880,549, Google Patents.
- [4] Neela, S.A. 2021, 'Cloud Computing Based Learning Web Application through Amazon Web Services', 2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021, pp. 472-475.
- [5] Nakamizo, Y. 2022, 'A SERVER-SIDE RENDERING METHOD FOR HANDLING LARGE-VOLUME 3D MODELS OF ENVIRONMENTAL DESIGN IN WEB-BASED AUGMENTED REALITY', *Journal of Environmental Engineering (Japan)*, vol. 87, no. 792, pp. 157-168, ISSN 1348-0685.
- [6] Keshari, P. 2023, 'Web Development Using ReactJS', *Proceedings - IEEE 2023 5th International Conference on Advances in Computing, Communication Control and Networking, ICAC3N 2023*, pp. 1571-1575.
- [7] Carvalho, F.M. 2023, 'Enhancing SSR in Low-Thread Web Servers: A Comprehensive Approach for Progressive Server-Side Rendering with any Asynchronous API and Multiple Data Models', *International Conference on Web Information Systems and Technologies, WEBIST - Proceedings*, pp. 37-48, ISSN 2184-3252.
- [8] Alshafaey, M.S. 2021, 'A new cloud-based classification methodology (CBCM) for efficient semantic web service discovery', *Cluster Computing*, vol. 24, no. 3, pp. 2269-2292, ISSN 1386-7857.
- [9] Sahria, Y 2020, 'Implementasi Teknik Web Scraping pada jurnal SINTA untuk analisis topik penelitian kesehatan Indonesia', *Prosiding University Research Colloquium*, repository.urecol.org.
- [10] Dwiyatno, S, Rachmat, E, Sari, AP, & ... 2020, 'Implementasi virtualisasi server berbasis docker container', ... *Riset Dan Observasi ...*, e-jurnal.lppmunsera.org.
- [11] Wardhana, WG, Arwani, I, & Rahayudi, B 2020, 'Implementasi Teknologi Restful Web Service Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis Website (Studi Kasus: Fakultas ...)', ... *Teknologi Informasi dan Ilmu ...*, j-ptiik.ub.ac.id.
- [12] Purnamasari, SD 2022, 'Implementasi Usability Testing dalam Evaluasi Website Sekolah', *Implementasi Usability Testing dalam ...*, eprints.binadarma.ac.id.
- [13] Kurniawan, AA 2020, 'Analisis Performa Progressive Web Application (PWA) Pada Perangkat Mobile', *Jurnal Ilmiah Informatika Komputer*, ejournal.gunadarma.ac.id.
- [14] Agustian, A, Andriyani, I, Khoerunisa, S, & ... 2020, 'Implementasi Teknik Equivalence Partitioning pada Pengujian Aplikasi E-learning Berbasis Web', ... *Sistem Informasi dan ...*, core.ac.uk.
- [15] Berners-Lee, T, Hendler, J, & Lassila, O 2023, 'The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities', ... *Invention of the World Wide Web*, dl.acm.org.
- [16] Vaio, A Di, & Varriale, L 2020, 'Blockchain technology in supply chain management for sustainable performance: Evidence from the airport industry', *International Journal of Information Management*, Elsevier.
- [17] Zheng, Y, Liu, Y, Xie, X, Liu, Y, Ma, L, & ... 2021, 'Automatic web testing using curiosity-driven reinforcement learning', 2021 *IEEE/ACM 43rd ...*, ieeexplore.ieee.org