

Perbandingan Mekanisme Rendering Untuk Optimasi Website Dengan Studi Kasus Website Penitipan Hewan

Moh Alvian Noer¹, I Made Suartana²

^{1,2} Program Studi S1 Teknik Informatika, Universitas Negeri Surabaya

¹moh.20037@mhs.unesa.ac.id

²madesuartana@unesa.ac.id

Abstrak— Dalam pengembangan sebuah website, pemilihan metode *rendering* sangat berpengaruh untuk memastikan performa, aksesibilitas, dan pengalaman pengguna yang optimal. Penelitian ini berfokus pada perbandingan metode rendering *Client Side Rendering (CSR)*, *Server Side Rendering (SSR)*, *Progressive Render (LazyLoad)* dengan studi kasus website penitipan hewan. Website penitipan hewan dibangun dengan menggunakan framework *NextJS* dari *Javascript* karena dirasa paling tepat untuk menerapkan ketiga metode rendering yang akan diujikan. Perbandingan metode rendering dalam sebuah website akan memberikan dampak positif dalam proses pengembangannya. Dengan melakukan pengujian menggunakan *GTMetriks*, penelitian ini akan mengevaluasi mengenai keefektifan dari masing-masing metode rendering dalam meningkatkan kecepatan dan responsifitas sebuah halaman web. Secara keseluruhan, dengan melakukan perbandingan ini dapat tergambarkan dengan jelas mengenai perbedaan dalam segi performa dari masing-masing metode rendering untuk pengembangan sebuah website terutama pada pemilihan metode rendering yang tepat pada situs *website* penitipan hewan. Berdasarkan hasil perbandingan dari ke 3 metode rendering yang telah diujikan, *Server Side Rendering (SSR)* merupakan metode rendering yang paling optimal untuk pengembangan sebuah *website*. Dengan melakukan penerapan pada metode *Server Side Rendering (SSR)* akan memberikan waktu muat yang lebih cepat, meningkatkan pengalaman dalam menampilkan konten yang lengkap, serta mengurangi beban pemrosesan pada sisi *client* karena semuanya akan dirender melalui sisi *server*.

Kata Kunci— *CSR*, *SSR*, *Progressive*, *Lazyload*, *NextJS*, *Javascript*, *GTMetriks*, *Website*.

I. PENDAHULUAN

Teknologi informasi menjadi bagian yang sangat penting dan tidak dapat dipisahkan dari kehidupan manusia. Di Indonesia sekarang, perkembangan teknologi informasi menjadi sangat pesat [1]. Dengan berkembang dan majunya teknologi informasi, hal ini semakin memudahkan manusia untuk mengakses dan mencari berbagai macam informasi di internet [2]. Pada saat ini, teknologi informasi sudah menjadi peran penting dalam kehidupan manusia. Website merupakan teknologi perantara internet yang saat ini sedang ramai digunakan. Dengan adanya website, segala informasi dapat diakses secara mudah menggunakan jaringan internet [3]. Menurut Global Digital Overview, jumlah dari pengguna internet di dunia telah mencapai 4,66 miliar dari total populasi penduduk sebanyak 7,83 miliar, dari data tersebut dapat dikatakan bahwa hampir 59.5% penduduk di dunia sudah

terhubung dengan internet. Dataindonesia.id juga melaporkan bahwa pengguna internet di Indonesia telah mencapai 205 juta jiwa, yang meningkat sebanyak lima kali lipat dibandingkan pengguna internet pada tahun 2017 yang sebanyak 136 juta jiwa [4]. Menurut laporan diatas dapat disimpulkan bahwa perkembangan teknologi informasi tidak pernah menurun dan akan terus berkembang secara pesat. Tetapi seiring berkembangnya sebuah teknologi dan semakin banyaknya jumlah dari pengguna internet dapat memberikan dampak buruk untuk performa sebuah website.

Semakin meningkatnya permintaan sesuai dengan pengalaman pengguna dalam menggunakan website yang responsif dan interaktif, penting bagi sebuah website dalam memperhatikan proses rendering yang cepat dan efisien [5]. Oleh karena itu pemilihan metode atau teknologi rendering sangat penting dan berpengaruh untuk mengoptimasi performa dari sebuah website. Pemilihan metode rendering sangat penting untuk memastikan pengalaman pengguna website menjadi lebih optimal [6]. Pemegang tanggung jawab didalam suatu proses rendering berada di sisi *FrontendRendering* website sendiri merupakan sebuah proses yang menghasilkan tampilan visual dari sebuah halaman website pada sebuah browser maupun perangkat lainnya, baik dari segi konten maupun data statis atau dinamis [7]. Ketika user menjalankan sebuah website, maka peramban dari website tersebut harus mengunduh beberapa konten dari *HTML*, *CSS*, *Javascript*, dan lainnya untuk digabungkan menjadi satu dan ditampilkan menjadi sebuah tampilan yang dapat dilihat dan diinteraksikan oleh user. Berbicara tentang pembuatan website yang baik, kualitas render dari sebuah website menjadi bagian utama yang harus diperhatikan [8].

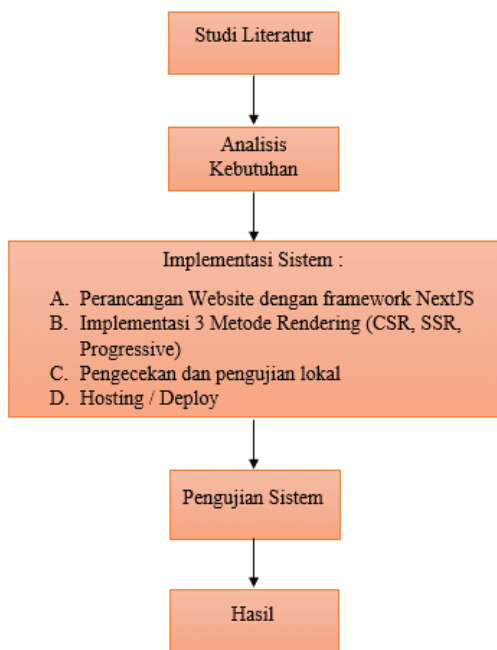
Terdapat beberapa mekanisme dari metode rendering yang tersedia, misalnya *Server Side Rendering (SSR)*, *Client Side Rendering (CSR)*, dan *progressive Rendering (Lazy Load)*. *Server Side Rendering* atau yang sering disebut dengan *SSR* merupakan sebuah metode rendering dimana server akan turun tangan secara langsung dalam menghasilkan dan mengirimkan halaman *HTML* yang akan ditampilkan kepada *client* setiap terdapat suatu permintaan. *Client Side Rendering* yang biasa disebut dengan *CSR* juga memiliki sebuah metode yang berbeda, yaitu dengan mengalihkan sebuah proses rendering ke sisi klien, dimana sebuah browser akan melibatkan *javascript* untuk merender dan menghasilkan tampilan konten secara dinamis setelah menerima sebuah data dari server. Sedangkan *Progressive Rendering* merupakan sebuah strategi render yang menekankan pengalaman

pengguna dalam memuat konten secara bertahap saat tersedia, tanpa harus menunggu halaman tersebut selesai dimuat secara keseluruhan. Dengan adanya penerapan Lazy Load dalam Progressive Rendering, user dapat melihat dan berinteraksi dengan bagian awal dari sebuah halaman dengan kecepatan yang lebih baik meskipun konten lainnya masih dalam proses pemuatan[9]. Penerapan dari ke tiga metode rendering diatas dapat diterapkan di dalam framework NextJS yang merupakan framework dari library ReactJS [10].

Dengan memahami perbandingan dari penerapan metode rendering secara menyeluruh, pengembang akan dapat membuat keputusan yang lebih tepat dalam memilih pendekatan yang sesuai dengan kebutuhan dan tujuan dari proyek yang mereka kembangkan terutama pada sebuah website. Hal tersebut meliputi tentang loading tampilan awal dari sebuah website, tingkat interaktivitas yang tinggi, dan yang terakhir adalah persepsi kecepatan akses yang maksimal [11]. Oleh karena itu, didalam penelitian ini peneliti tertarik untuk melakukan perbandingan mengenai mekanisme rendering yang terdiri dari Client Side Rendering (CSR), Server Side Rendering (SSR), dan Progressive Rendering menggunakan Lazy load untuk mengoptimasi sebuah tampilan website dengan studi kasus website penitipan hewan dengan harapan menemukan hasil yang optimal mengenai proses rendering dalam pengembangan suatu website menggunakan framework NextJS [10].

II. METODE PENELITIAN

Penelitian ini dilakukan melalui beberapa tahapan terstruktur pada diagram alur penelitian pada Gbr. 1



Gbr. 1 Diagram Alur Penelitian

Pada Gbr.1 menjelaskan bahwa metode penelitian yang dilakukan memiliki beberapa alur tahapan. Berikut merupakan penjelasan dari setiap alur tahapan pada metode penelitian yang digunakan :

A. Studi Literatur

Dalam tahapan studi literatur ini, penelitian ini akan berfokus terhadap eksplorasi mengenai konsep kunci dalam pengembangan website penitipan hewan dengan menggunakan 3 metode rendering yang berbeda : Client Side Rendering (CSR), Server Side Rendering (SSR), dan progressive rendering dengan menggunakan Lazy Load. Studi Literatur ini akan mencakup mengenai analisis tentang prinsip dasar dari penggunaan masing-masing metode rendering, proses dari CSR dalam melakukan render pada sisi klien, proses dari SSR dalam melakukan render dari sisi server, dan yang terakhir adalah Progressive rendering yang melakukan proses render dengan teknik Lazy load yang akan memuat konten secara bertahap. Selain itu, penelitian ini juga akan berfokus terhadap penggunaan framework NextJS yang digunakan untuk pengembangan antarmuka dalam konteks CSR, SSR, dan penerapan teknik Lazy Load dengan menggunakan bahasa pemrograman JavaScript untuk mempercepat proses pemuatan sebuah halaman dalam menggunakan Progressive Rendering. Dengan adanya Studi Literatur ini, pemahaman yang mendalam mengenai kekurangan, kelebihan, dan prinsip dasar mengenai masing-masing metode rendering telah diperoleh. Hal tersebut akan membantu serta memandu penelitian lebih lanjut dalam tahap perancangan dan pengujian website penitipan hewan.

B. Analisis Kebutuhan

Analisis Kebutuhan yang digunakan untuk mendukung penelitian ini, antara lain :

Kebutuhan Fungsional dan Non fungsional

a. Kebutuhan Fungsional

- Sistem dapat menerapkan metode rendering CSR
- Sistem dapat menerapkan metode rendering SSR
- Sistem dapat menerapkan metode Progressive rendering dengan teknik Lazy Load

b. Kebutuhan Non Fungsional

- Sistem dapat menampilkan UI tampilan dengan baik
- Tidak adanya bug system

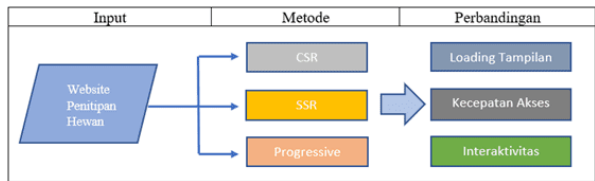
C. Implementasi Sistem

1. Perancangan Website Dengan Framework NextJS

Perancangan website dengan menggunakan framework NextJS sendiri merupakan proses pembangunan website dengan teknik modern yang memudahkan proses penggunaan. NextJS sendiri merupakan sebuah kerangka dari Javascript yang memiliki berbagai macam kemudahan dalam mengembangkan website yang cepat, responsif dan mudah dimaintenance. Pemilihan NextJS sebagai framework dalam pengembangan website ini merupakan langkah yang penting, dikarenakan dalam proses pengembangan suatu website sangat tergantung dengan kebutuhan dan preferensi dari developer. Dengan menggunakan NextJS dapat memudahkan dalam pembuatan struktur folder seperti konten, template, dan aset. Selain itu Framework NextJS juga memiliki kelebihan dalam proses perancangannya karena dirasa lebih efisien,

fleksibel dan dapat menghasilkan website dengan tampilan yang dinamis dan responsif.

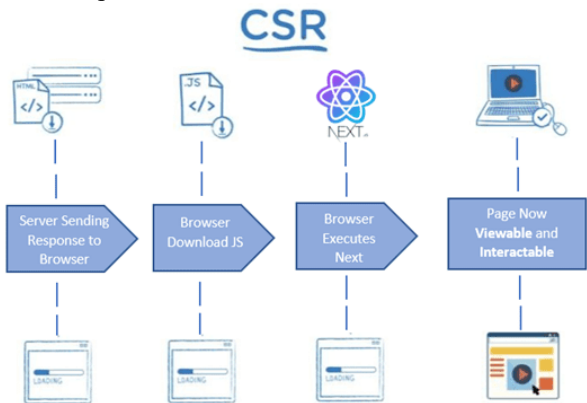
2. Implementasi 3 Metode Rendering



Gbr. 2 Diagram Implementasi 3 Metode Rendering

Pada Gbr. 2 dapat diambil kesimpulan bahwa dengan melakukan Implementasi 3 metode rendering memiliki tujuan untuk membandingkan beberapa metode rendering dengan fokus pada loading tampilan awal, kecepatan akses yang optimal, dan tingkat interaktivitas. Berikut merupakan penjelasan serta perbedaan dari 3 metode rendering yang akan digunakan :

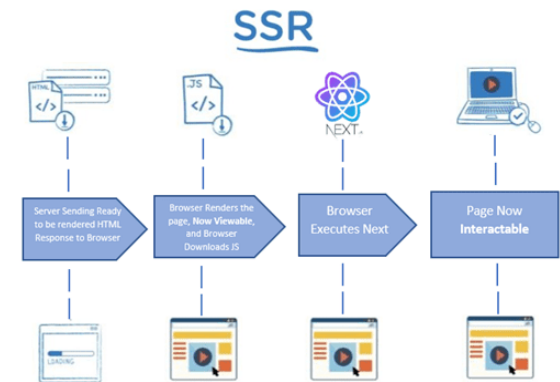
a. Implementasi Metode CSR



Gbr. 3 Diagram Implementasi Metode CSR [10].

Pada Gbr.3 dapat diambil kesimpulan bahwa Implementasi metode CSR merupakan proses penerapan sebuah metode dalam pengembangan website dimana halaman website akan ditampilkan dan dirender pada sisi klien atau lebih tepatnya dibrowser pengguna. Dalam proses implementasinya, pengguna akan mengunjungi suatu halaman website dan server hanya mengirim file html dan javascript kosong yang kemudian akan diunduh dan dieksekusi oleh browser pengguna.

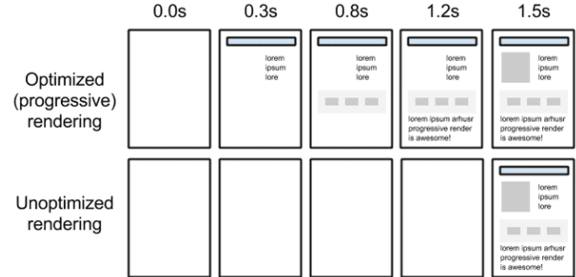
b. Implementasi Metode SSR



Gbr. 4 Diagram Implementasi Metode SSR [10].

Pada Gbr. 4 dapat diambil kesimpulan bahwa Implementasi metode SSR merupakan proses penerapan sebuah metode dimana proses halaman web akan dihasilkan oleh server secara dinamis sesuai dengan permintaan dari penggunaanya. Berbeda dengan metode CSR yang akan dirender seluruhnya pada sisi browser dari penggunaanya, SSR sendiri akan melibatkan pembuatan halaman secara menyeluruh pada sisi server sebelum ditampilkan didalam website penggunaanya.

c. Implementasi Progressive Rendering (Lazyload)



Gbr. 5 Diagram Implementasi Progressive Rendering [15].

Pada Gbr. 5 dapat diambil kesimpulan bahwa Implementasi Progressive Rendering dengan menggunakan teknologi lazy loading merupakan sebuah proses dimana website akan menampilkan suatu tampilan secara bertahap saat halaman berhasil dimuat. Hal tersebut akan mempersingkat proses pemuatan halaman dan memberikan kesan yang lebih cepat. Didalam penerapan progressive rendering, lazy load diterapkan untuk memuat konten secara bertahap hingga pengguna membutuhkannya atau pada saat pengguna berinteraksi dengan elemen tertentu.

3. Pengecekan dan Pengujian Lokal

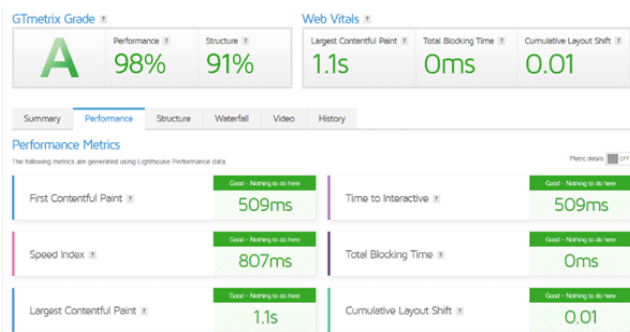
Pengecekan dan pengujian secara local yang merupakan sebuah proses penting dalam pengembangan sebuah website. Setelah berhasil melakukan perancangan dan penerapan sebuah metode

didalam suatu code, pada tahap ini akan dilakukan sebuah pengecekan dan pengujian untuk mengantisipasi kemungkinan terjadinya sebuah bug. Dengan adanya pengecekan dan pengujian secara local, developer akan secara mudah mengidentifikasi dan memperbaiki sebuah masalah sebelum project tersebut masuk kedalam proses pengunggahan atau biasa disebut dengan produksi. Proses ini bertujuan untuk memastikan agar situs web dapat berfungsi dengan baik tanpa adanya kendala sedikitpun saat dijalankan.

4. Hosting / Deploy

Hosting/deploy yang merupakan sebuah tahapan untuk menghadirkan sebuah situs website ke dunia online dengan tujuan agar dapat diakses oleh siapa saja. Dengan menggunakan platform hosting yang tepat serta mengikuti proses deployment yang baik, maka website dapat diakses oleh pengguna secara global. Tujuan dari proses hosting pada tahap ini adalah agar memudahkan website yang telah dikembangkan ke dalam 3 metode rendering tadi dapat diuji menggunakan Web Core Vital menggunakan GT Metriks untuk mengetahui kelebihan dan kekurangan dari setiap metode yang diterapkan.

D. Pengujian Sistem

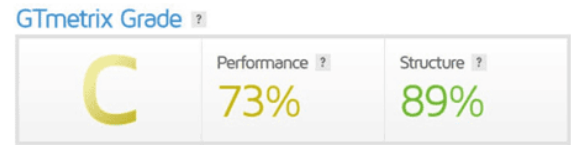


Gbr. 6 Pengujian Sistem Menggunakan GTMetrix

Pada Gbr. 6 dapat diambil kesimpulan bahwa pengujian sistem menggunakan Core Web Vitals dengan GT Metrik, berfokus pada evaluasi menyeluruh terhadap kinerja dan pengalaman pengguna dalam menggunakan website penitipan hewan dengan menerapkan tiga metode rendering yang berbeda yaitu : Client Side Rendering (CSR), Server Side Rendering (SSR), dan Progressive Rendering menggunakan Teknologi Lazy Load. Dengan metode rendering CSR, pengujian dilakukan untuk memastikan bahwa halaman website dapat ditampilkan secara cepat dan responsif di sisi klien sesuai dengan Standar dari Core Web Vitals untuk kecepatan dan Interaktivitas. Pada metode rendering SSR, pengujian dilakukan untuk memantau waktu respon server dan kecepatan rendering pada suatu halaman website dengan fokus pada kinerja saat mengakses konten dan perpindahan halaman secara cepat. Sedangkan pada Progressive Rendering dengan menggunakan teknologi Lazy Load akan berfokus pada kecepatan pemuatan konten tambahan saat user melakukan sebuah interaksi, serta kemampuan sebuah

halaman website saat mempertahankan kualitas interaktivitas saat berjalannya proses rendering. Terdapat beberapa hasil analisis yang dihasilkan dalam proses pengujian menggunakan GTMetriks :

1. GTMetrix Grade

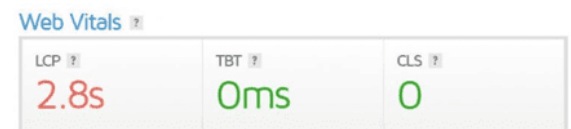


Gbr. 7 GTMetrix Grade

Pada Gbr. 7 dapat diambil kesimpulan bahwa GTMetrix Grade merupakan hasil keseluruhan dari performa website yang diujikan. Pada bagian ini, terdapat tiga metriks yang diperhatikan, yaitu :

- Overall Score : Skor keseluruhan dari website dalam bentuk grade (A, B, C).
- Performance : Skor performa dari kecepatan akses sebuah website dalam bentuk persentase.
- Structure : Skor struktur yang menilai seberapa baik website tersebut dibangun dalam bentuk persentase.

2. Web Vitals

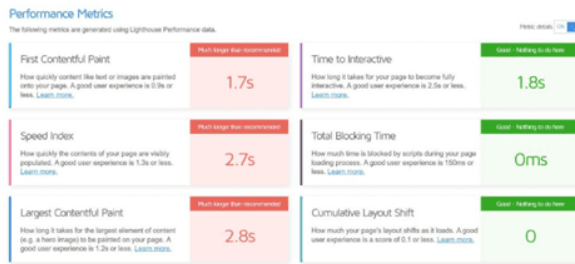


Gbr. 8 Web Vitals

Pada Gbr. 8 dapat diambil kesimpulan bahwa Web Vitals merupakan sebuah hasil dari pengukuran beberapa elemen inti dari website yang diuji. Web vitals berguna untuk menghitung waktu loading dari website yang merupakan faktor penting dalam mempengaruhi ranking website dalam pencarian google. Terdapat tiga metriks yang diperhatikan dalam Web Vitals, yaitu :

- Largest Contentful Paint (LCP) : Mengukur waktu loading pemuatan elemen konten yang terbesar.
- Total Blocking Time (TBT) : Mengukur waktu loading halaman web sampai semua fungsi dapat diakses keseluruhan oleh pengguna.
- Content Layout Shift (CLS) : Mengukur perubahan layout halaman saat proses loading.

3. Perfomance Metrics



Gbr. 9 Performance Metrics

Pada Gbr. 9 dapat diambil kesimpulan bahwa Performance Metrics merupakan pengukuran sebuah website menggunakan tools Lighthouse. Terdapat beberapa metrik yang diperhatikan, antara lain :

- First Contentful Paint : Mengukur waktu yang dibutuhkan saat konten pertama ditampilkan.
- Speed Index : Mengukur kecepatan yang dibutuhkan agar website dapat terlihat secara penuh.
- Time to Interactive : Mengukur waktu yang dibutuhkan agar website dapat digunakan sepenuhnya.
- Largest Contentful Paint : Mengukur waktu loading pemuatan elemen konten yang terbesar.
- Total Blocking Time : Mengukur waktu loading halaman web sampai semua fungsi dapat diakses keseluruhan oleh pengguna.
- Cumulative Layout Shift : Mengukur perubahan layout halaman saat proses loading.

Pengujian perbandingan 3 metode rendering pada website ini akan berfokus pada tampilan halaman login, register, landing page, reservation, dan dashboard pada website penitipan hewan. Dengan melakukan pengujian Core Web Vitals menggunakan GT Metrik, diharapkan pengujian sistem dapat memastikan bahwa website penitipan hewan dapat memberikan pengalaman pengguna yang lebih baik dan optimal dari segi kecepatan loading dalam memuat halaman, kecepatan akses, dan interaktivitas sesuai dengan tujuan dari pengembangan website tersebut.

E. Hasil

Dari hasil pengujian halaman login, register, landing page, reservation, dan dashboard yang menerapkan 3 metode rendering berbeda pada halaman website penitipan hewan menggunakan tools GTMetrik diharapkan dapat memberikan pemahaman yang mendalam mengenai kinerja dan pengalaman pengguna dalam menggunakan website tersebut, misalnya :

1. Pengujian halaman login dan register akan memperlihatkan kecepatan loading suatu halaman dan response agar pengguna dapat mengakses halaman tersebut dengan cepat agar tidak menghambat proses login maupun pendaftaran.
2. Pengujian halaman landing page yang bertujuan untuk mengidentifikasi potensi penyempitan titik pemuatan suatu halaman yang akan memperlambat proses konversi

dan akan berdampak buruk pada tampilan awal sebuah website.

3. Pengujian halaman reservation yang bertujuan untuk memastikan bahwa proses pemesanan dapat berjalan dengan lancar dan memastikan pengguna terhindar dari adanya gangguan dalam mengakses halaman transaksi.
4. Pengujian dashboard yang bertujuan untuk mengidentifikasi dimana dashboard mengalami penundaan dan ketidaksempurnaan dalam memberikan data yang akurat serta waktu respon yang cepat kepada pengguna.

Dengan menggunakan GTMetrik dalam menguji halaman-halaman yang menerapkan 3 metode rendering yang berbeda ini diharapkan dapat mengoptimalkan kinerja dan meningkatkan pengalaman pengguna untuk menambah nilai positif dalam menggunakan dan mengakses website penitipan hewan ini.

III. HASIL DAN PEMBAHASAN

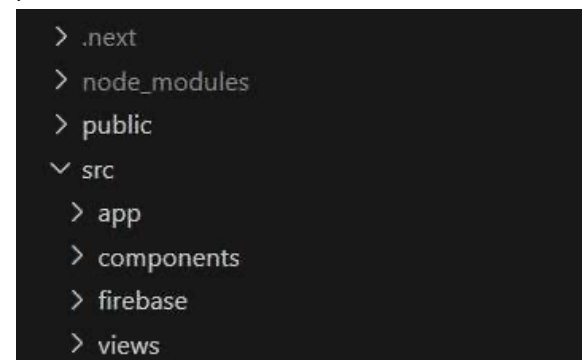
Pada bab ini, menjelaskan hasil dari implementasi perancangan pengujian yang telah disusun sebelumnya. Pembahasan mencakup tentang implementasi rancangan penelitian yang dimulai dari perancangan website dan penerapan metode render sampai dengan pengujian dan perbandingan terhadap sebuah website penitipan hewan.

A. Implementasi Rancangan Penelitian

Pada bagian ini, menjelaskan mengenai tahapan-tahapan yang dilakukan dalam mengimplementasikan penerapan 3 metode rendering yang telah dijelaskan pada bab sebelumnya.

1. Perancangan Website penitipan hewan dengan Framework NextJS

Dalam pembangunan website penitipan hewan ini menggunakan framework NextJS yang merupakan framework dari javascript yang bersifat open-source dan berbasis React. Didalam penerapannya, framework NextJS memiliki struktur sebagai berikut :



Gbr. 10 Struktur File NextJS

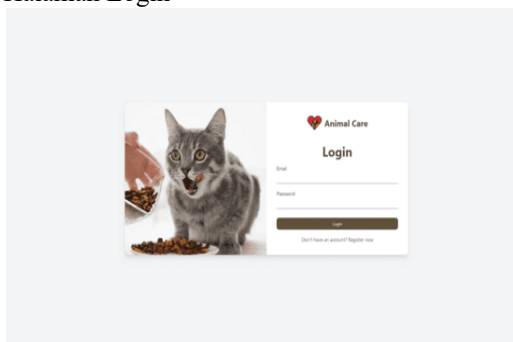
Pada Gbr. 10 dapat diambil kesimpulan bahwa NextJS akan menyimpan code yang telah kita buat menjadi lebih terstruktur dengan baik. Hal ini akan mendukung proses pengembangan website menjadi lebih baik juga untuk kedepannya. Terdapat penerapan

struktur didalam pengembangan website penitipan hewan ini, diantaranya :

- .next merupakan sebuah folder direktori build yang dihasilkan oleh NextJS setiap kali menjalankan sebuah perintah build.
- node_modules merupakan sebuah tempat penyimpanan depedensi dan modul yang telah diinstall melalui npm.
- public merupakan tempat penyimpanan file statis yang disediakan secara langsung tanpa perlu proses build. Biasanya file public berisikan image.
- src merupakan sebuah folder yang digunakan untuk mengorganisir code didalam proyek yang sedang dikembangkan. Di dalam folder src berisikan beberapa folder berupa app, components, firebase, dan views.
- app merupakan sebuah folder yang digunakan untuk mengorganisir code yang berisikan page atau halaman yang digunakan didalam proyek yang sedang dikembangkan.
- Components merupakan sebuah folder yang digunakan untuk mengorganisir code yang berisikan komponen-komponen yang terpakai didalam file page.
- Firebase merupakan sebuah folder yang berisikan firebase config, yang berguna untuk menghubungkan antara aplikasi dengan layanan firebase yang digunakan.
- Views merupakan sebuah folder yang berisikan tampilan gabungan dari component yang telah dibuat.

Dari perancangan Website penitipan hewan dengan framework Next JS menghasilkan beberapa tampilan halaman sebagai berikut :

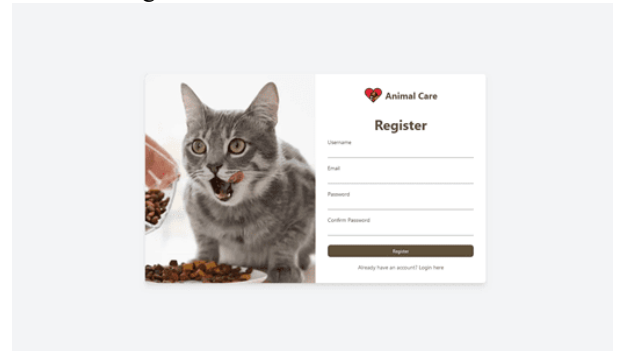
Halaman Login



Gbr. 11 Halaman Login

Pada Gbr. 11 diatas merupakan tampilan dari halaman login yang merupakan sebuah halaman yang diakses oleh pengguna setelah berhasil melakukan register. Didalam page ini pengguna diharuskan masuk ke dalam akun mereka masing-masing dengan mengisi alamat email dan password yang telah didaftarkan sebelumnya.

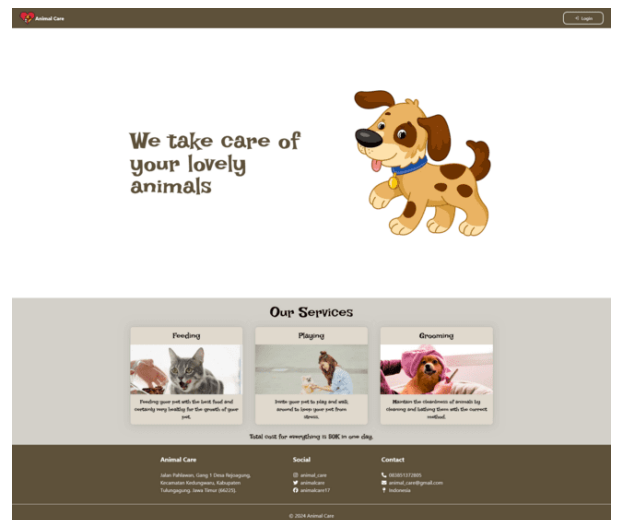
Halaman Register



Gbr. 12 Halaman Register

Pada Gbr. 12 diatas merupakan tampilan dari halaman Register yang merupakan sebuah halaman yang memungkinkan pengguna untuk membuat akun baru yang berisikan formulir dengan informasi pribadi seperti username, email, password, dan confirm password.

Halaman Utama



Gbr. 13 Halaman Utama (Landing Page)

Pada Gbr. 13 diatas merupakan tampilan dari halaman Landing Page yang merupakan sebuah halaman utama dari sebuah website yang berisikan sebuah informasi dan gambaran mengenai layanan dan fasilitas yang disediakan didalam website penitipan hewan.

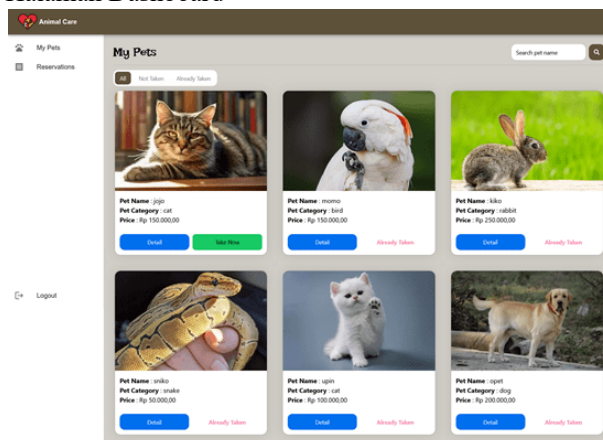
Halaman Reservasi

No	Name	Price	Pet Name	Pet Category	Day
1	Sal	87500000	Sal	Sal	1
2	Wah	11000000	Wah	Wah	2
3	Wah	11000000	Wah	Wah	3
4	Sal	87500000	Sal	Sal	4
5	Wah	11000000	Wah	Wah	5

Gbr. 14 Halaman Reservasi

Pada Gbr. 14 diatas merupakan tampilan dari halaman Reservasi yang merupakan halaman dimana pengguna diharuskan mengisi form reservasi yang berisikan nama, nomor telepon, alamat email, nama hewan yang dititipkan, jenis hewan yang dititipkan, serta berapa lama hewan tersebut akan dititipkan (dalam hitungan hari). Pengguna juga langsung dapat melihat, mengedit, dan menghapus data reservasi secara langsung didalam page reservation ini jika terdapat sebuah kesalahan dan keraguan dalam proses reservation.

Halaman Dashboard



Gbr. 15 Halaman Dashboard

Pada Gbr. 15 diatas merupakan tampilan dari halaman Dashboard yang merupakan sebuah halaman yang menampilkan tentang data keseluruhan hewan yang dititipkan. Didalam halaman dashboard terdapat beberapa fitur yang disediakan seperti fitur search, fitur filtering, fitur details dan fitur takenow.

2. Implementasi 3 Metode Rendering (CSR, SSR, Progressive)

a. Implementasi Metode Rendering CSR

Implementasi metode rendering CSR (Client Side Rendering) merupakan tahapan mengimplementasikan sebuah code yang menangani sebuah rendering

komponen dan halaman pada sisi client. Pada penerapan next JS menggunakan App Router agar semua halaman dan komponen di render pada sisi klien (CSR) hanya cukup menambahkan 'use client' pada bagian atas dari code yang dikembangkan. Hal tersebut akan memberi tahu NextJS bahwa komponen tersebut harus dirender pada sisi klien.

```

"use client";
import Image from "next/image";
import { Card, CardHeader, CardBody, CardFooter } from "@nextui-org/card";
import Navbar from "@components/Navbar/Navbar";
import Footer from "@components/Footer/Footer";
import { irish } from "../fonts";

export default function Home() {
  return (
    <div>
      <Navbar />
      <main>
        <section className="h-[calc(100vh-64px)] grid grid-cols-1 md:grid-cols-2">
          <div className="text-center">
            <h1 className={irish.className} font-bold text-7xl text-primary>
              We take care of your lovely animals
            </h1>
            <div className="order-first md:order-none md:justify-self-end">
              <Image
                src="/img/Gambar 1.png"
                alt="gambar anjing"
                width={448}
                height={477}
              />
            </div>
          </div>
          <div id="service" className="bg-[rgb(94,81,58,0.27)]">
            <div className="p-3 text-center">
              <h2 className={irish.className} text-[45px]>Our Services</h2>
            </div>
          </div>
        </section>
      </main>
      <Footer />
    </div>
  );
}

```

Gbr. 16 Contoh Implementasi Metode Rendering CSR

Pada Gbr. 16 diatas merupakan penerapan 'use client' pada halaman landing page, yang menandakan bahwa NextJS harus merender halaman landing page pada sisi client server (CSR).

b. Implementasi Metode Rendering SSR

Implementasi metode rendering SSR (Server Side Rendering) merupakan tahapan proses mengimplementasikan sebuah code yang menangani sebuah rendering komponen dan halaman yang akan diunduh dan dirender seluruhnya pada sisi server. Pada penerapan NextJS menggunakan App Router, default dari semua halaman dan komponen di dalam direktori 'app' semuanya akan dirender menggunakan Server Side Rendering (SSR). Jadi user tidak perlu menambahkan apapun jika ingin semua halaman dan komponen dirender pada sisi server.

```

import Image from "next/image";
import { Card, CardHeader, CardBody, CardFooter } from "@nextui-org/card";
import Navbar from "@components/Navbar/Navbar";
import Footer from "@components/Footer/Footer";
import { irish } from "../fonts";

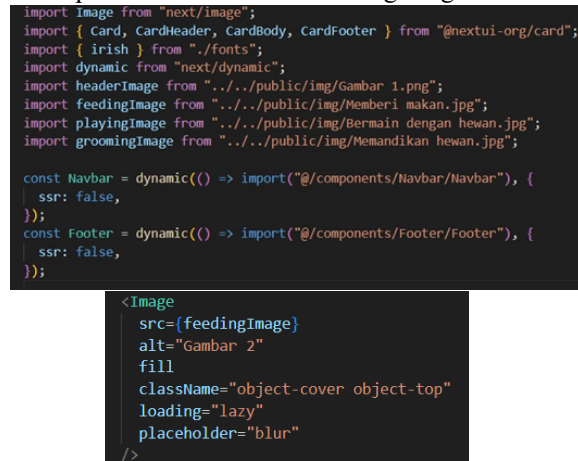
export default function Home() {
  return (
    <div>
      <Navbar />
      <main>
        <section className="h-[calc(100vh-64px)] grid grid-cols-1 md:grid-cols-2">
          <div className="text-center">
            <h1 className={irish.className} font-bold text-7xl text-primary>
              We take care of your lovely animals
            </h1>
            <div className="order-first md:order-none md:justify-self-end">
              <Image
                src="/img/Gambar 1.png"
                alt="gambar anjing"
                width={448}
                height={477}
              />
            </div>
          </div>
          <div id="service" className="bg-[rgb(94,81,58,0.27)]">
            <div className="p-3 text-center">
              <h2 className={irish.className} text-[45px]>Our Services</h2>
            </div>
          </div>
        </section>
      </main>
      <Footer />
    </div>
  );
}

```

Gbr. 17 Contoh Implementasi Metode Rendering SSR

Pada Gbr. 17 diatas menjelaskan bahwa tidak adanya tambahan apapun didalam halaman landing page pada Implementasi SSR. Hal tersebut menandakan bahwa halaman dan komponen pada halaman landing page akan dirender secara server side rendering (SSR) karena default dari directori 'App' merupakan perenderan pada sisi server.

c. Implementasi Metode Rendering Progressive



Gbr. 18 Contoh Implementasi Metode Progressive

Pada Gbr. 18 diatas terdapat penerapan dynamic routing pada komponen navbar dan footer. Berbeda dengan image, pada component image tersebut menerapkan Image loading = lazy dan placeholder = blur. Hal tersebut akan membuat komponen navbar, footer, dan image dirender secara dinamis atau bertahap. Dengan kesimpulan bahwa halaman landingpage akan dirender secara progressive dengan komponen navbar, footer, dan image yang akan dirender secara bertahap dengan menggunakan lazyload.

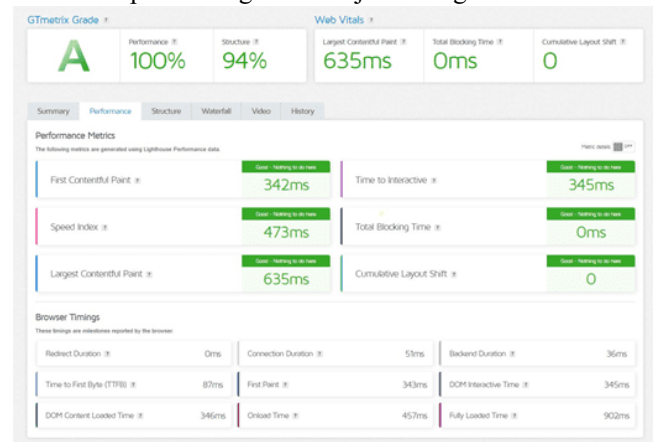
B. Pengujian dan Hasil Perbandingan

Pada bagian ini, menjelaskan mengenai tahapan-tahapan yang dilakukan dalam pengujian dan hasil perbandingan dari penerapan 3 metode rendering yang telah dijelaskan pada bab sebelumnya. Didalam pengujian metode rendering ini menggunakan tools GTMetrik yang merupakan sebuah langkah terpenting dalam memastikan sebuah website yang telah dibangun memiliki kinerja yang baik dan optimal. GTMetrik digunakan untuk menganalisa kinerja atau performa dari sebuah website untuk mengukur apakah website tersebut tergolong baik atau malah sebaliknya. GTmetrik memiliki ketentuan seperti GTmetrik grade yang berisikan skor penilaian dari performa website yang sedang dianalisa. GTMetrik dapat memberikan hasil analisis mengenai kecepatan dan efisiensi rendering situs website yang dibangun, baik itu dengan menggunakan metode rendering CSR (Client Side Rendering), SSR (Server Side Rendering, Progressive Rendering (LazyLoad).

1. Pengujian

a. Pengujian Metode Rendering CSR

Pengujian ini befokus untuk memastikan bahwa proses dari tampilan website yang dirender melalui sisi client dapat berfungsi dan berjalan dengan baik.



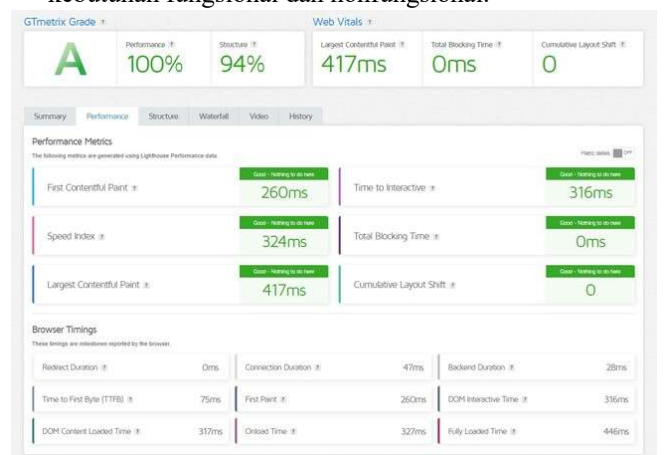
Gbr. 19 Contoh Pengujian Metode Rendering CSR

Pada Gbr. 19 diatas bahwa pengujian halaman landing page dengan menggunakan metode rendering CSR mendapatkan skor dari beberapa matrix penilaian yang berbeda. Pada pengujian menggunakan GTMetrik mendapatkan hasil sebagai berikut :

- FCP (First Contentfull Paint) = 342ms
- SI (Speed Index) = 473ms
- LCP (Largest Contentfull Paint) = 635ms
- TTI (Time To Interactive) = 345ms
- TBT (Total Blocking Time) = 0ms
- CLS (Cumulative Layout Shift) = 0

b. Pengujian Metode Rendering SSR

Pengujian ini berfokus untuk memastikan bahwa proses dari tampilan website yang dirender melalui sisi server dapat berjalan dengan baik dan dapat memenuhi kebutuhan fungsional dan nonfungsional.



Gbr. 20 Contoh Pengujian Metode Rendering SSR

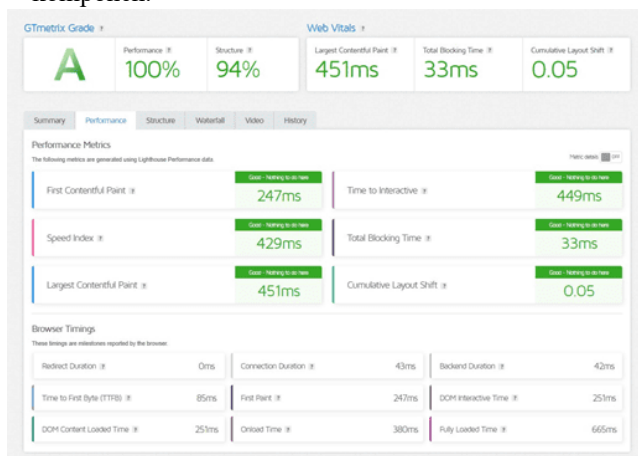
Pada Gbr. 20 diatas bahwa pengujian halaman landing page dengan menggunakan metode rendering SSR mendapatkan skor dari beberapa matrix penilaian yang

berbeda. Pada pengujian menggunakan GTMetrik mendapatkan hasil sebagai berikut :

- FCP (First Contentfull Paint) = 260ms
- SI (Speed Index) = 324ms
- LCP (Largest Contentfull Paint) = 417ms
- TTI (Time To Interactive) = 316ms
- TBT (Total Blocking Time) = 0ms
- CLS (Cumulative Layout Shift) = 0

c. Pengujian Metode Rendering Progressive

Pengujian ini berfokus untuk memastikan bahwa proses dari tampilan website yang dirender menggunakan Progressive render dengan teknologi LazyLoad dapat berjalan dengan baik dan dapat mengoptimalkan dalam pemuatan halaman dan komponen.



Gbr. 21 Contoh Pengujian Metode Rendering Progressive

Pada Gbr. 21 diatas bahwa pengujian halaman landing page dengan menggunakan metode rendering Progressive dengan teknologi Lazyload mendapatkan skor dari beberapa matrix penilaian yang berbeda. Pada pengujian menggunakan GTMetrik mendapatkan hasil sebagai berikut :

- FCP (First Contentfull Paint) = 247ms
- SI (Speed Index) = 429ms
- LCP (Largest Contentfull Paint) = 451ms
- TTI (Time To Interactive) = 449ms
- TBT (Total Blocking Time) = 33ms
- CLS (Cumulative Layout Shift) = 0

2. Hasil Perbandingan

Berdasarkan hasil perbandingan tampilan website yang diuji dengan menggunakan 3 metode rendering yang berbeda, menunjukkan hasil dan perbedaan yang sangat signifikan dalam kinerja dan performa sebuah website. Hasil tersebut dirangkum didalam sebuah tabel berikut :

TABEL 1
TABEL PENGUJIAN METODE RENDERING CSR

No	Halaman Yang Diuji	Matrixs Pengujian					
		FCP	SI	LCP	TTI	TBT	CLS
1	Login	446ms	443ms	446ms	446ms	0ms	0
2	Register	251ms	243ms	251ms	251ms	0ms	0
3	Landing Page	342ms	473ms	635ms	345ms	0ms	0
4	Reservasi	321ms	304ms	321ms	321ms	0ms	0.01
5	Dashboard	253ms	875ms	1.2s	1.3s	31ms	0.05
Jumlah		1613ms	2338ms	2853ms	2663ms	31ms	0.06
Rata - Rata		322,6ms	467,6ms	570,6ms	532,6ms	6,2ms	0.012

TABEL 2
TABEL PENGUJIAN METODE RENDERING SSR

No	Halaman Yang Diuji	Matrixs Pengujian					
		FCP	SI	LCP	TTI	TBT	CLS
1	Login	234ms	226ms	234ms	234ms	0ms	0
2	Register	211ms	197ms	211ms	235ms	0ms	0
3	Landing Page	260ms	324ms	417ms	316ms	0ms	0
4	Reservasi	307ms	281ms	307ms	552ms	62ms	0.01
5	Dashboard	229ms	705ms	901ms	1s	21ms	0.05
Jumlah		1241ms	1733ms	2070ms	2337ms	83ms	0.06
Rata - Rata		248,2ms	346,6ms	414ms	467,4ms	27,6ms	0.012

TABEL 3
TABEL PENGUJIAN METODE RENDERING PROGRESSIVE

No	Halaman Yang Diuji	Matrixs Pengujian					
		FCP	SI	LCP	TTI	TBT	CLS
1	Login	232ms	270ms	622ms	605ms	33ms	0.05
2	Register	251ms	303ms	348ms	446ms	15ms	0.12
3	Landing Page	247ms	429ms	451ms	449ms	33ms	0.05
4	Reservasi	252ms	418ms	252ms	481ms	50ms	0.01
5	Dashboard	316ms	916ms	1.3s	558ms	41ms	0.05
Jumlah		1298ms	2336ms	2973ms	2539ms	172ms	0.28
Rata - Rata		259,6ms	467,2ms	594,6ms	507,8ms	34,4ms	0.056

Dari hasil perbandingan yang didapatkan melalui perhitungan rata-rata dari setiap metode rendering, menunjukkan bahwa :

- Metode rendering Server Side Rendering (SSR) menjadi yang paling optimal dalam segala aspek yang diperhitungkan yaitu dengan rata-rata skor FCP=248,2ms, SI=346,6ms, LCP=414ms, TTI=467,4ms, TBT=27,6ms, dan CLS=0.012.
- Selanjutnya disusul dengan metode rendering Progressive (Lazyload) dengan rata-rata skor FCP=259,6ms, SI=467,2ms, LCP=594,6ms, TTI = 507,8ms, TBT=34,4ms, dan CLS=0.056.
- Dan yang terakhir yaitu metode rendering Client Side Rendering (CSR) dengan rata-rata skor FCP=322,6ms, SI=467,6ms, LCP=570,6ms, TTI= 532,6ms, TBT=6,2ms, dan CLS=0.012.

Dapat disimpulkan bahwa Server Side Rendering (SSR) merupakan metode rendering yang paling cepat diantara metode rendering yang lain karena SSR merupakan sebuah metode yang cepat dalam pemuatan file seperti HTML karena metode tersebut melakukan rendering dan chacing pada sisi server, sehingga akan mengurangi beban pada klien dan meningkatkan performa dari sebuah website yang dijalankan. Selanjutnya Progressive rendering dengan teknologi lazyload juga bisa menjadi opsi ke 2 setelah SSR. Karena penggunaan Progresive rendering dengan teknologi lazy load akan membantu pengguna dalam meningkatkan performa dengan memuat konten secara bertahap dan hanya ketika dibutuhkan saja. Dengan adanya lazy load, bagian bagian yang penting atau yang diinginkan akan dimuat terlebih dahulu, sementara

konten yang lainnya akan dimuat pada latar belakang. Hal tersebut akan membantu mengurangi waktu muat awal halaman dan penggunaan bandwidth yang berlebih. Sementara dengan Client Side Rendering (CSR) menjadi metode rendering yang paling lambat dikarenakan metode CSR selalu melakukan proses pengunduhan, parsing, dan eksekusi javascript melalui browser Client. Hal tersebut menyebabkan waktu muat awal yang lebih lama. Selain itu CSR juga memiliki interaksi awal yang lambat dan FCP yang lambat dikarenakan semua konten serta proses rendering yang terjadi akan dijalankan dan diunduh pada sisi client.

IV. KESIMPULAN

Hasil pengujian ini menunjukkan bahwa pemilihan metode rendering sangat berpengaruh terhadap proses optimasi dan mengatasi performa yang buruk dalam pengembangan sebuah website. Berdasarkan hasil perbandingan dari ke 3 metode rendering yang telah diujikan, Server Side Rendering (SSR) merupakan metode rendering yang paling optimal untuk pengembangan sebuah website. Dengan melakukan penerapan pada metode Server Side Rendering (SSR) akan memberikan waktu muat yang lebih cepat, meningkatkan pengalaman dalam menampilkan konten yang lengkap, serta mengurangi beban pemrosesan pada sisi client karena semuanya akan dirender melalui sisi server.

Hal tersebut merupakan langkah yang penting dalam memastikan aksesibilitas dan performa yang baik dalam sebuah website, terutama untuk user yang menggunakan perangkat dengan spesifikasi yang rendah atau dengan minimnya koneksi internet. Sehingga dapat disimpulkan bahwa Implementasi metode rendering Server Side Rendering (SSR) dapat membantu dalam mengatasi performa yang buruk dan mengoptimalkan kinerja website secara keseluruhan.

REFERENSI

- [1] Andriansyah, Doni. (2019). Performance Dan Stress Testing Dalam Mengoptimasi Website. Jakarta, Indonesia: Cbis Journal - Vol. 07 No. 01 (2019) : Maret.
- [2] Baehaqi, Ahmad, Muhamad Subhi Bashit, Richardus Eko Indrajit, dan Rido Dwi Kurniawan. (2023). Front End Learning Management System Development Using The Nextjs Framework. Tangerang, Banten: Jurnal Teknik Informatika (JUTIF) DOI: <https://doi.org/10.52436/1.jutif.2023.4.4.1273> Vol. 4, No. 4, August 2023, pp. 899-911.
- [3] Iskandar, Taufan Fadhilah, Muharman Lubis, Tien Fabrianti Kusumasari, dan Arif Ridho Lubis. (2020). Comparison between client-side and server-side rendering in the web development. Jalan Telekomunikasi No. 1, Bandung, 40257: IOP Conf. Series: Materials Science and Engineering 801 (2020) 012136 doi:10.1088/1757-899X/801/1/012136.
- [4] Herman, dan Alvi Geovanny. (2022). Analisis Rendering Performa Antara Server Side Dan Client Side Pada Web Application. Tiban Indah, Kec. Sekupang, Kota Batam: Jurnal Ilmiah Betrik, Volume.13, No.03.
- [5] Akbarrizky, Aldy, Afdal Ramdan Daman Huri, Bambang Wisnuadhi, dan Lukmannul Hakim Firdaus. (2023). Optimasi micro frontend website untuk meningkatkan load times: teknik, tantangan, dan best practice. Riau, Indonesia: Jurnal Computer Science and Information Technology (CoSciTech) Vol. 4, No. 2, Agustus 2023, hal. 366-375.
- [6] Koivukoski, Jesse. (2021). Reducing the loading time of a single-page web application. Espoo, Finland: Aalto University School of Science.
- [7] Ardiyanto, Roni dan Eka Ardianto. (2024). Analisa Peformasi Metode Rendering Website: Client Side, Server Side, dan Incremental Static Regeneration. Jl. Trilomba Juang No 1, Semarang, Jawa Tengah 5024, Indonesia: Computer Science (CO-SCIENCE) Vol. 4 No. 1 Januari 2024.
- [8] Widyani, Ni Putu Kerti, A.A KOMPIANG Oka Sudana, dan I Nyoman Piarsa. (2021). Pengujian Performa Sistem Informasi Perpustakaan Online pada Universitas Hindu Indonesia (Astakali UNHI) Menggunakan Tools GTmetrix. Bali: JITTER- Jurnal Ilmiah Teknologi dan Komputer Vol. 2, No. 3 Desember 2021.
- [9] Virtanen, Aki. (2023). The Integration of Native Mobile App Features into a Progressive Web App. Lahti, Finland: LAB University of Applied Sciences Bachelor of Engineering, Information Technology.
- [10] A Jartarghar, H. ., Rao Salanke, G. ., A.R, A. K. ., G.S, S. ., & Dalali, S. . (2022). React Apps with Server-Side Rendering: Next.js. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 14(4), 25–29.
- [11] Mongkau,, Desmoon Christopher, Arya Berelaku, dan Sitti Arni. (2023). Analisis Performa Website Menggunakan GTMetrix. Makassar: Jurnal Minfo Polgan Volume 12, Nomor 2, Juni 2023 DOI : <https://doi.org/10.33395/jmp.v12i2.12518>.
- [12] Andini, Asri Putri Dwi Gita, Dian Wahyuningsih, dan Mahmud Yunus. (2022). Analisis Dan Peningkatan Performa Aplikasi Berbasis Website Menggunakan Stress Tools Gtmetrix. Malang, Jawa Timur: Jurnal Teknologi Informasi Komunikasi (e-Journal) Vol. 9 No. 2 (2022) 191 - 201.
- [13] Dewi, Ida Ayu Pradita, A.A. KOMPIANG Oka Sudana, dan I Made Suwija Putra. (2021). Pengujian performa website Sistem Manajemen Registrasi Terintegrasi (SMRTI) pada Universitas Hindu Indonesia menggunakan tools GTmetrix. Bali: JITTER- Jurnal Ilmiah Teknologi dan Komputer Vol. 2, No. 3 Desember 2021.
- [14] Kertamukti, Rama, Mochamad Sodik, dan B.J. Sujibto. (2021). Website Work Quality Assessment to Improve Webometrics Rank on the Website of <https://uin-suka.ac.id/>. Yogyakarta, Indonesia: Jurnal Warta Indonesia Volume 3 No 2 – 2021, page 105-123.
- [15] Faradilla A. (2020). Cara SEO Wordpress untuk menaikkan peringkat website. Diakses pada 19 Juni 2024 dari <https://www.hostinger.co.id/tutorial/cara-seo-wordpress>