



ISSN : 2339 - 1871

JURNAL ILMIAH BETRIK

Besemah Teknologi Informasi dan Komputer

Editor Office : LPPM Sekolah Tinggi Teknologi Pagar Alam, Jln. Masik Siagim No. 75
Simpang Mbacang, Pagar Alam, SUM-SEL, Indonesia
Phone : +62 852-7901-1390.
Email : betrik@sttpagaralam.ac.id | admin.jurnal@sttpagaralam.ac.id
Website : <https://ejournal.sttpagaralam.ac.id/index.php/betrik/index>

ANALISIS RENDERING PERFORMA ANTARA SERVER SIDE DAN CLIENT SIDE PADA WEB APPLICATION

Herman¹, Alvi Geovanny²

Fakultas Ilmu Komputer Universitas Internasional Batam¹²

Baloi-Sei Ladi, Jl. Gajah Mada, Tiban Indah, Kec. Sekupang, Kota Batam

Sur-el : herman@uib.ac.id¹, 1931143.alvi@uib.edu²

Abstrak: Salah satu cara mengakses informasi ialah menggunakan aplikasi website. Aplikasi web merupakan suatu aplikasi yang dapat diakses dari browser web melalui Internet atau jaringan intranet. Perkembangan ini mengakibatkan banyak *traffic user* yang memengaruhi performa website, sehingga munculnya istilah rendering sisi klien walaupun saat ini menurut beberapa penelitian, masing-masing ada yang lebih cepat dibanding lainnya. Oleh karena itu, tujuan dari penelitian ini adalah merancang dua buah aplikasi website yang kedua frameworknya sama-sama menggunakan bahasa pemrograman Javascript yaitu ReactJS untuk sisi kliennya dan NextJS untuk sisi servernya. Hasil akhir dari penelitian ini adalah dua buah aplikasi website, untuk keseluruhan performa ReactJS mendapatkan skor 68 sedangkan NextJS memperoleh angka 35. Pengujian dilakukan agar dapat memberikan gambaran untuk pemilihan *framework* kedepannya bagi pengembang.

Kunci Utama: sisi server, sisi klien, reactjs, nextjs, aplikasi website

Abstract: *One way to access information is to use a website. A web application is an application that can be accessed from a web browser via the Internet or an intranet network. This development resulted in a lot of user traffic affecting website performance, so that the term client-side rendering emerged, although currently server-side rendering is still proven to be faster, but the popularity of frameworks that use client-side actually ranks first. Therefore, the purpose of this research is to design two websites where both frameworks use the Javascript programming language, named ReactJS for the client-side and NextJS for the server-side.*

Keywords : *server-side, client-side, reactjs, nextjs, website application*

1. PENDAHULUAN

Pada abad ke-21 ini, teknologi informasi sudah menjadi salah satu bagian yang sudah tidak bisa dihapuskan dalam kehidupan manusia. Perkembangan teknologi informasi pada era saat ini sudah semakin pesat, terutama di negara Indonesia. Dengan berkembangnya teknologi informasi, telah memudahkan manusia untuk

melakukan pencarian informasi dimana aja dan kapan saja [1]. Teknologi informasi ini sudah menjadi peranan penting dalam kehidupan manusia, teknologi perantara internet yang sering dipakai saat ini salah satunya adalah website [2].

Dari laporan Digital 2021 : GLOBAL DIGITAL OVERVIEW mengungkapkan bahwa jumlah pengguna media internet didunia sudah mencapai 4,66 miliar pada

tahun 2021 dari total populasi didunia sebanyak 7,83 miliar, ini menunjukkan bahwa hampir 59.5% populasi dari dunia sudah terhubung dengan internet [3]. Laporan dari dataindonesia.id juga menyatakan bahwa pengguna internet di Indonesia mencapai 205 juta pada Januari 2021, ini meningkat sebanyak lima kali lipat jika dibandingkan dengan jumlah pengguna pada tahun 2017 yang jumlahnya sebanyak 136 juta [4]. Dari hal tersebut membuktikan bahwa perkembangan teknologi tidak pernah menurun dan akan terus meningkat. Namun seiring meningkatnya jumlah pengguna internet menyebabkan dampak performa pada website [5].

Untuk mengoptimasi performa web, munculah teknologi *Client-side rendering* dan *Server-side rendering*. *Client-side rendering (CSR)* adalah metode dimana situs web dirender bukan di server melainkan di render browser menggunakan *javascript*, jadi browser hanya mendapatkan semua konten dari dokumen HTML dan file JS yang akan dirender [6]. Sedangkan *Server-side rendering (SSR)* memperlakukan data dan front-end keduanya secara terpisah, bagian belakang dikembangkan dalam satu bahasa pemrograman, dan memaparkan datanya melalui API untuk diambil dan direnderkan oleh klien [7]. Salah satu contoh framework yang menerapkan *client-side rendering* dan *server-side rendering* adalah NextJS, yang merupakan framework dari library ReactJS [8].

Melalui framework ReactJS terbukti yang digunakan oleh salah satu platform terbesar yaitu Facebook, dapat dengan efisien mengelola dan meningkatkan performa websitenya. Faktanya, Facebook dulunya menggunakan bahasa pemrograman PHP sebelum bermigrasi ke

ReactJS. Namun, banyaknya iklan yang mengakibatkan platform sangat lambat dan sulit untuk melakukan *maintenance* dengan harus melibatkan banyak orang. [9] [8].

Menurut penelitian dari [10], penelitian tersebut membangun dua website yang satunya menggunakan *HTML, CSS, dan PHP*, dan lainnya dengan *HTML, CSS, dan Javascript* dengan *Google Audit* sebagai tempat dasar uji. Hasil yang diperoleh menunjukkan bahwa secara keseluruhan performa sisi server terbukti lebih cepat dibanding sisi klien. Beberapa metriknya seperti *Performance, Accessibility*, serta *SEO*. Namun untuk *Best Practices* keduanya sama-sama mendapatkan skor 93.

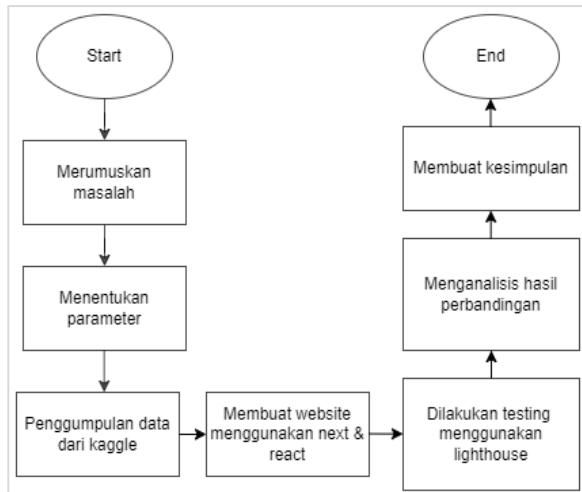
Penelitian selanjutnya diambil dari [11]. Pada jurnal ini, penelitian tersebut mengambil dua buah website *ecommerce* yang menggunakan framework ReactJS untuk sisi klien dengan NextJS untuk sisi servernya. Untuk jumlah baris kode pada ReactJS sebanyak 36626, sedangkan NextJS berjumlah hanya 10983 baris. Masing-masing *framework* diuji sebanyak 5 kali dengan *Google Lighthouse*. Namun hasil yang didapatkan justru diunggulkan oleh ReactJS.

Berdasarkan uraian tersebut, maka tujuan penelitian ini adalah untuk membandingkan empat parameter dari rendering sisi klien dengan sisi server rendering yaitu *performance, accessibility, best practices*, dan *SEO* dengan membangun dua buah website untuk diuji coba pada *Lighthouse Chrome DevTools*. Dalam merancang website, metode *Scrum Methodology* juga akan digunakan dengan tujuan bekerja lebih terkordinasi dan organisir dengan membagi tugas menjadi bagian-bagian kecil yang terurut berdasarkan prioritas [12].

2. METODE PENELITIAN

2.1 Alur Penelitian

Alur penelitian merupakan rangkaian proses penulis untuk melakukan penelitian yang ditunjukkan pada gambar 3.1.



Gambar 1. Flowchart

Berikut mengenai penjelasan alur diatas:

1. Pertama merumuskan masalah berdasarkan identifikasi masalah yang tertulis pada latar belakang.
2. Setelah itu menentukan parameter yang dibutuhkan untuk membangun website data rendering dan melakukan testing.
3. Ketiga mengumpulkan data review dari kaggle mengenai review customer.
4. Membuat website rendering data dengan metode iterative incremental development menggunakan framework Next JS dan React JS.
5. Kemudian dilakukan testing menggunakan lighthouse sebagai objek penelitian dan hasil yang diperoleh diperbandingkan dengan variabelnya adalah performance, accessibility, best practices, dan SEO.

6. Terakhir setelah mendapatkan semua hasil, dilakukan rangkuman ke dalam kesimpulan sebagai kajian akhir oleh penulis.

2.2 Parameter yang Diukur

Parameter yang dijadikan sebagai acuan validasi untuk mengukur kecepatan rendering antara rendering sisi klien dengan sisi server rendering diambil berdasarkan referensi dari jurnal utama, yaitu *performance*, *accessibility*, *best practices*, dan *search engine optimization (SEO)*. Untuk mengetahui seberapa cepat performa dari kedua website, akan digunakan *Lighthouse* untuk uji coba.

Performance adalah kecepatan munculnya tampilan website untuk digunakan oleh user. Semakin pendek durasi *loading*, semakin besar peluang pula muncul di hasil pencarian yang teratas. Untuk parameter *performance* terdapat 6 metrik, yakni:

1. *First Contentful Paint*, merupakan waktu yang dibutuhkan website saat gambar dan teks pertama dirender [13].
2. *Time to Interactive*, waktu yang dibutuhkan *user* menunggu untuk bisa berinteraksi dengan suatu halaman.
3. *Total Blocking Time*, adalah waktu antara *First Contentful Paint* dan *Time to Interactive*. Pada momen ini user bisa melihat halaman, tapi belum bisa berinteraksi.
4. *Speed Index*, waktu yang dibutuhkan untuk elemen visual suatu halaman di-load sepenuhnya.
5. *Largest Contentful Paint*, adalah waktu yang dibutuhkan untuk memunculkan teks atau gambar terbesar [14].

6. *Cumulative Layout Shift*, merupakan pergerakan tata letak keseluruhan yang terjadi secara tidak terduga pada halaman situs [15]. Jika elemen yang ada tidak stabil dan bergerak, akan berdampak negatif pada *lighthouse*.

Accessibility bertujuan membantu memastikan situs web dapat digunakan oleh semua orang dan tidak melanggar hukum yang ada. Untuk penilaian aksesibilitas ini, *lighthouse* akan mengecek elemen seperti HTML, tags, ALT text, ARIA landmark, dan sebagainya.

Best practices adalah peraturan yang harus dipenuhi oleh suatu website agar bisa mendapatkan 100 persen di *lighthouse*. Beberapa berikut yakni hindari pengalihan halaman arahan, kecepatan halaman saat pertama kali buka adalah tiga detik, dan ukuran atau berat halaman di bawah 500 KB. Keamanan juga wajib diperhatikan seperti penggunaan HTTPS, outbound link yang aman, dan Javascript yang terbaru.

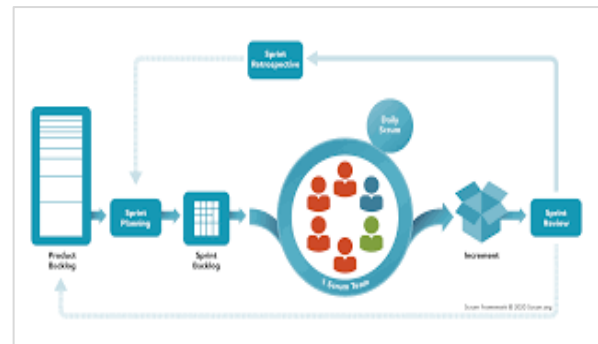
Search Engine Optimization (SEO) adalah teknik yang digunakan untuk memaksimalkan sebuah website agar lebih populer atau terbaca di search engine guna meningkatkan peringkat mesin pencari untuk menarik lebih banyak pengunjung dengan memanfaatkan algoritma dan mesin pencari [16].

2.3 Teknik Pengumpulan Data

Teknik pengumpulan data yang dilakukan pada penelitian ini adalah dengan mengambil salah satu dataset dari *kaggle* yang isinya mengenai review/kepuasan konsumen terhadap suatu produk dan dimasukkan ke dalam database MySQL dan di hosting ke *google cloud*.

2.4 Metode Pengembangan Website

Berdasarkan alur penelitian yang ada, penulis menetapkan metode *agile software development methods*, dengan menggunakan model Scrum. Scrum adalah metodologi pengembangan aplikasi yang membantu individu, tim, atau organisasi mencapai tujuan yang ingin dicapai. Metodologi ini berfokus pada transparansi, tinjauan, dan penyesuaian proses pengembangan sistem [17]. Dalam metode ini, terdapat empat tahap yang perlu dilakukan yaitu:



Gambar 2. Scrum Metodologi

1. Product Backlog

Menentukan keseluruhan fitur yang harus dilaksanakan selama proyek berlangsung. Elemen dari setiap fiturnya disusun secara hierarkis dan setiap tugas memiliki prioritas, deskripsi, tingkat kesusahan, dan lainnya. Isi dalam fitur tersebut dapat ditambahkan setiap saat [18].

2. Sprint Planning

Melakukan penjadwalan perencanaan kegiatan yang akan dilaksanakan untuk memenuhi kebutuhan yang ditetapkan dalam *backlog*.

3. Daily Scrum

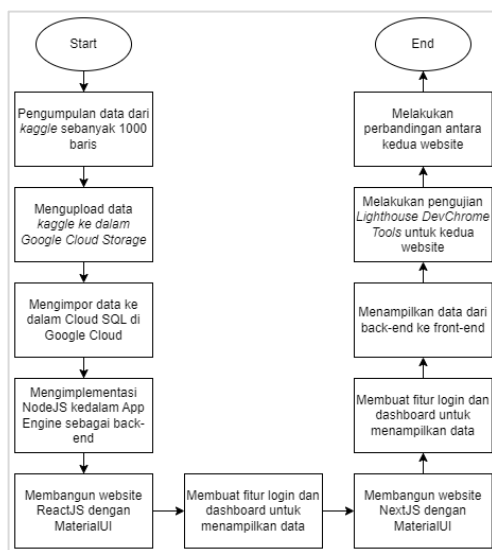
Membuat laporan dimana isinya mengandung tiga hal utama yaitu progress apa yang sudah dilakukan, apa yang akan dilakukan pada hari ini, dan hambatan yang ditemui dan

estimasi durasi biasanya sekitar 15 menit [19].

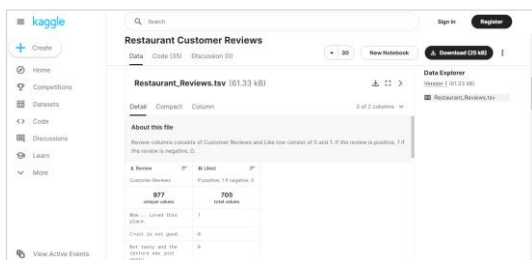
4. Sprint Review

Melakukan uji coba dari pengembangan fitur yang sudah jadi untuk memastikan apakah website tersebut sudah siap untuk di *hosting* atau tidak.

2.5 Rancangan Eksperimen



Gambar 3. Flowchart Implementasi Rancangan



Gambar 4. Data Ulasan Pelanggan

Setelah mendapatkan dataset dari *kaggle* dengan jumlah total 1000 data mengenai *customer review*, maka selanjutnya penulis akan menggunakan dua metode yaitu penerapan dan pengujian. Untuk metode penerapan, pertama-tama menginisialisasi kedua proyek sisi klien dan server yang masing-masing menggunakan ReactJS dan NextJS dengan me-render

semua data yang sudah tersedia di Google Cloud ke dalam homepage yang dilakukan dengan *scrum*.

Setelah selesai, dilanjutkan metode eksperimen yang diambil dari hasil penerapan yang sudah jadi sebelumnya. Pada tahap ini pengukuran kecepatan performa rendering akan dilakukan dengan *lighthouse*. Hasil utama yang disajikan oleh *lighthouse* adalah *performances*, *accessibility*, *best practices*, dan *SEO*. Untuk metrik performance, Metrik yang digunakan pada pengukuran kecepatan respon adalah detik. Indikasi baik ditandai oleh lingkaran hijau, indikasi perlu peningkatan ditandai dengan kotak informasi berwarna kuning, dan buruk ditandai dengan segitiga peringatan merah.

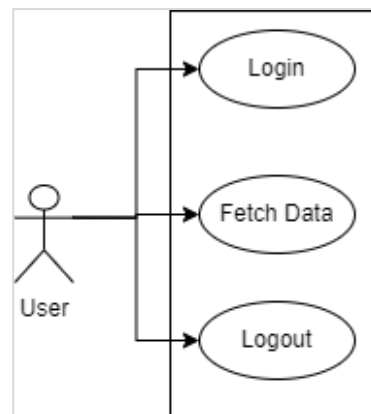
2.6 Menganalisis Hasil Perbandingan

Setelah semua data sudah diperoleh dari hasil kedua metode tersebut, penulis dapat menarik kesimpulan dari hasil analisa berdasarkan pengukuran menggunakan parameter yang ditetapkan sebagai nilai pada masing-masing sisi rendering.

3. HASIL DAN PEMBAHASAN

3.1 Sprint Pertama

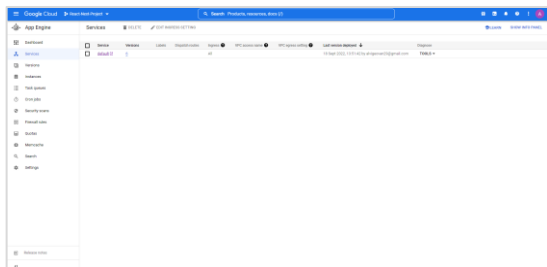
Pada sprint pertama, penulis merancang diagram *use case* agar mendapatkan gambaran sistem yang akan dibangun.



Gambar 5. Use Case Diagram

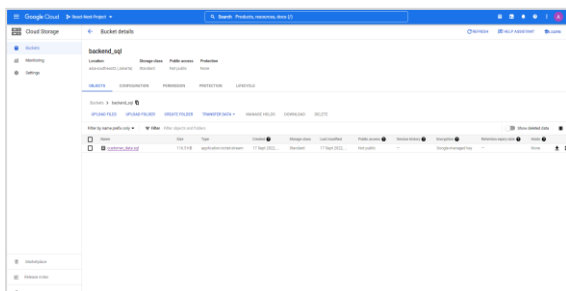
User dapat melakukan 3 aksi yaitu *Login*, *Fetch Data*, dan *Logout* untuk kedua *website*. Jumlah data yang diambil dari *database kaggle* sebanyak 1000 buah. *Use Case Diagram* adalah penggambaran grafis yang terdiri dari actor, fitur, dan hubungannya yang menghadirkan kerangka kerja [20].

3.2 Sprint Kedua



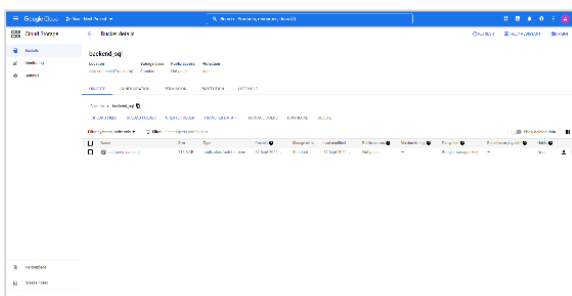
Gambar 6. Halaman App Engine

Berikut ini adalah pengimplementasi *back-end* pada *App Engine*.



Gambar 7. Halaman Cloud Storage

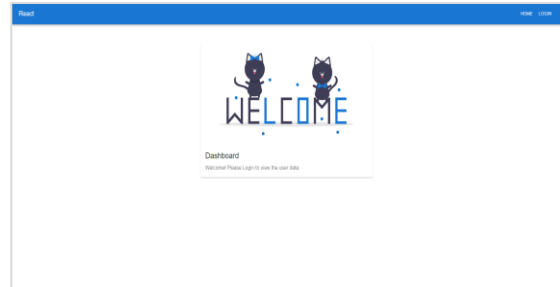
Selanjutnya data disimpan dalam *Cloud Storage* dalam bentuk file *SQL*.



Gambar 8. Halaman Cloud SQL

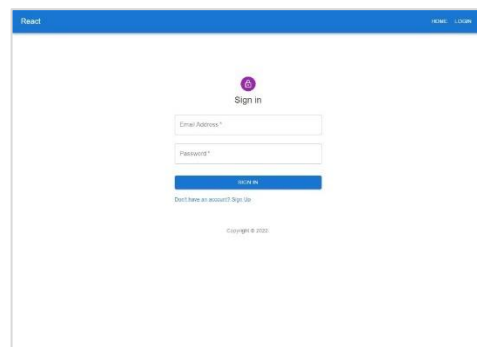
Terakhir file *SQL* diimport ke *Cloud SQL* sebagai *database MySQL*.

3.3 Sprint Ketiga



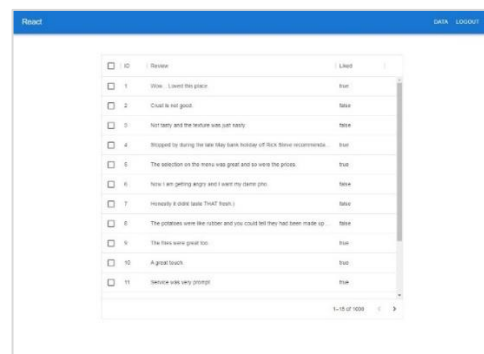
Gambar 9. Halaman Dashboard

Pada tahap ketiga, mulai mengembangkan halaman *dashboard* sebagai halaman utama sebelum *login*.



Gambar 10. Halaman Login

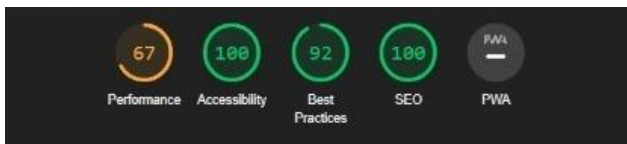
Halaman *login* juga dibuat seperti gambar diatas.



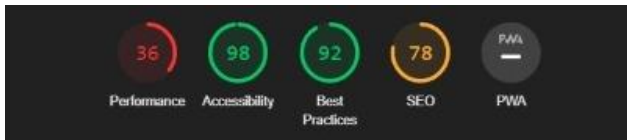
Gambar 11. Halaman Data

Setelah *login*, *user* dapat melihat halaman data seperti diatas ini.

3.4 Sprint Keempat



Gambar 12. Hasil ReactJS



Gambar 13. Hasil NextJS

Kedua gambar diatas ini merupakan hasil dari pengujian *lighthouse* melalui *chrome dev tools* setelah website sudah dibangun.

Pembahasan

Implementasi *website* tersebut menggunakan ReactJS untuk rendering sisi *klien*, dan NextJS untuk rendering sisi *server*. Kedua *website* dibuat sama persis baik fiturnya maupun UI/UX nya menggunakan MaterialUI sebagai framework CSS tujuannya hasil yang didapatkan dapat lebih akurat. Untuk data, diambil berdasarkan rekap yang sudah jadi dari *kaggle* mengenai *customer review* kemudian di *convert* supaya dapat diimport ke dalam database dan mudah dibaca oleh *user* melalui API. Tahap selanjutnya *back-end* dibangun menggunakan NodeJS dan dihosting ke *google cloud* agar dapat dimanfaatkan secara global APInya oleh ReactJS dan NextJS sekaligus. Website yang sudah selesai diuji cobakan pada *lighthouse*. Parameter yang diambil pada pengujian ini ada empat, yaitu *performance*, *accessibility*, *best practices*, dan *seo*. Analisis ini bertujuan agar pengembang mendapatkan gambaran untuk dipertimbangkan atau meningkatkan

performa dan popularitas website yang sedang dikembangkan.

Untuk mengetahui bagaimana masing-masing skor metrik dihitung cepat, sedang, atau lambat, website *Google Lighthouse* menyediakan tabel seperti berikut ini.

Tabel 1. Pembagian jangkauan skor metrik

Kecepatan	Skor Metrik
Lambat (Merah)	0 – 49
Sedang (Orange)	50 – 89
Cepat (Hijau)	90 - 100

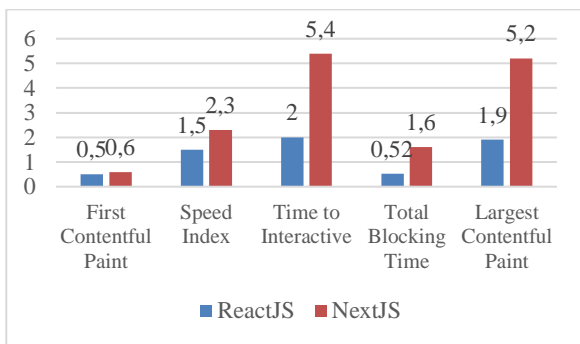
Pengujian test kecepatan pemuatan ini dilakukan pada mesin Windows 10 dan peramban Google Chrome. Aplikasi diuji dengan localhost dan *loading speed test* dilakukan menggunakan *lighthouse* yang merupakan sebuah tools yang tersedia di Chrome browser DevTools. Masing-masing website dilakukan uji testing sebanyak 5 kali untuk mendapatkan hasil yang lebih maksimal. Untuk lebih jelasnya akan ditampilkan dibawah ini.

Tabel 2. Hasil Pengujian ReactJS

ReactJS	Test 1	Test 2	Test 3	Test 4	Test 5	Rata-rata
Performance	70	83	82	70	69	74,8
Accessibility	100	100	100	100	100	100
Best Practices	92	92	92	92	92	92
SEO	100	100	100	100	100	100
Total	420	290	290	440	470	382
Blocking Time	ms	ms	ms	ms	ms	ms

Tabel 3. Hasil Pengujian NextJS

NextJS	Test 1	Test 2	Test 3	Test 4	Test 5	Rata-rata
Performance	34	36	36	47	48	40,2
Accessibility	98	86	98	86	86	90,8
Best Practices	83	92	83	92	92	88,4
SEO	78	89	78	89	89	84,6
Total	1.870	1.080	1.590	930	920	1.278
Blocking Time	ms	ms	ms	ms	ms	ms



Gambar 14. Grafik Perbandingan Kedua Metrik

4. SIMPULAN

Penelitian ini dilakukan dengan tujuan agar dapat mengetahui perbandingan antara *client-side rendering* dengan *server-side rendering*. Parameter yang dihitung mencakup *performance*, *accessibility*, *best practices*, dan *SEO*. Hasil eksperimen ini membuktikan bahwa kecepatan *client-side rendering* lebih cepat sehingga ReactJS menjadi framework yang paling disukai dan dicari oleh pengembang di peringkat pertama pada website user survei stackoverflow Tahun 2022. Namun walaupun ReactJS sangat populer, grafik pengguna NextJS juga terus meningkat drastis setiap Tahun. Perbedaan popularitas ReactJS dan NextJS tidak dapat dibandingkan juga karena umur pengeluaran yang jauh. Jika ingin mendapatkan hasil SEO yang lebih baik,

maka gunakan SSR, namun jika ingin membuat aplikasi website yang lebih responsif dan refresh halaman saat navigasi maka gunakan CSR.

DAFTAR RUJUKAN

- [1] F. Husnah, "Analisis Penggunaan Internet Dalam Pembelajaran Sains Dikelas Xii Sma Negeri 1 Tualang Riau," Vol. 4, Pp. 349–356, 2022.
- [2] T. Nasution, W. Susanti, Y. Armi, And R. Rahmadian, "Edumatic : Jurnal Pendidikan Informatika Aplikasi Panic Button Untuk Keamanan Warga Berbasis Android," Vol. 6, No. 1, Pp. 39–48, 2022, Doi: 10.29408/Edumatic.V6i1.5127.
- [3] Simon Kemp, "Digital 2020: Global Digital Overview," 2020. <https://Datareportal.Com/Reports/Digital-2020-Global-Digital-Overview> (Accessed Oct. 26, 2022).
- [4] Alif Karnadi, "Pengguna Internet Di Indonesia Capai 205 Juta Pada 2022," 2022. <https://Dataindonesia.Id/Digital/Detail/Pengguna-Internet-Di-Indonesia-Capai-205-Juta-Pada-2022> (Accessed Oct. 26, 2022).
- [5] D. A. Derm And D. W. Awan, Chamdan Mashuri, Ginanjar Setyo Permadi, Duta Alif Gunawan, "Membuat Game Berbasis Website Menggunakan Bahasa Javascript Dan Php," 2022.
- [6] T. Khanum, "Server-Side Rendering (Ssr) Vs Client-Side Rendering (Csr)," 2021. <https://Dev.To/Codewithtee/Server-Side-Rendering-Ssr-Vs-Client-Side-Rendering-Csr-3m24>.
- [7] M. Beke, "On The Comparison Of Software Quality Attributes For Client-Side And Server-Side Rendering," No. June, Pp. 14–15, 2018.
- [8] M. Fariz, S. Lazuardy, And D. Anggraini, "Modern Front End Web Architectures With React . Js And Next . Js," Vol. 7, No. 1, Pp. 132–141, 2022.

- [9] Ferenc Hámori, "The History Of React.Js On A Timeline," 2022. <https://Blog.Risingstack.Com/The-History-Of-React-Js-On-A-Timeline/> (Accessed Oct. 27, 2022).
- [10] T. Fadhilah Iskandar, M. Lubis, T. Fabrianti Kusumasari, And A. Ridho Lubis, "Comparison Between Client-Side And Server-Side Rendering In The Web Development," *Iop Conf. Ser. Mater. Sci. Eng.*, Vol. 801, No. 1, 2020, Doi: 10.1088/1757-899x/801/1/012136.
- [11] Z. Dinku, "React.Js Vs. Next.Js," No. May, 2022.
- [12] S. A. Aklani *Et Al.*, "Web-Based Operation Dashboard," Vol. 6, No. 1, Pp. 65–74, 2022.
- [13] B. Wisnuadhi *Et Al.*, "Studi Komparatif Fungsionalitas, Performance Dan User Experience Pada Website Geopark," *J. Inf. Syst. Res.*, Vol. 3, No. 4, Pp. 579–589, 2022, Doi: 10.47065/Josh.V3i4.1879.
- [14] E. A. Phie, S. Rahman, And P. S. Informatika, "Implementasi Teknik Search Engine Optimization Pada," No. 02, Pp. 35–49, 2021.
- [15] I. Kainu, "Iida Kainu Optimization In React . Js Methods , Tools , And Techniques To Improve Perfor- Mance Of Modern Web Applications," No. May, 2022.
- [16] P. Calvin Kosno, A. Muawwal, A. Yunus, S. Informasi, And S. Kharisma Makassar, "Implementasi Search Engine Optimization Pada Website Lelangyuk Untuk Meningkatkan Ranking Search Engine Result Page (Serp)," *Jtriste*, Vol. 9, No. 1, Pp. 56–70, 2022, [Online]. Available: www.Superbengkel.Co.Id.
- [17] S. Sentosa *Et Al.*, "Online Paxjoo Dengan Menerapkan," Vol. 4, No. 2.
- [18] A. Wahyudi, I. Riadi, And A. Dahlan Jl Ahmad Yani Tamanan, "Peran Strategis Scrum Master Pada Pengembangan Perangkat Lunak Perpustakaan Sekolah Berbasis Android."
- [19] G. D. Mulyadi, "Perancangan Sistem Informasi Manajemen Pt. Xyz Sepatu Pernikahan Online Menggunakan Scrum."
- [20] F. S. Am, A. Harris, And A. Nugroho, "Perancangan Sistem Informasi Akademik Berbasis Web Pada Sman 15 Muaro Jambi," Pp. 285–294, 2022.