

Pengembangan *Frontend* Sistem Informasi Layanan Tekos Berbasis Web Menggunakan *Framework* NextJS

1st Maulana Akbar Ramadhan

Fakultas Informatika,
Universitas Telkom

Bandung, Indonesia

maulanaakbarramadhan@students.telkomuniversity.ac.id

2nd Mira Kania Sabariah

Fakultas Informatika
Universitas Telkom

Bandung, Indonesia

mirakania@telkomuniversity.ac.id

3rd Shinta Yulia Puspitasari

Fakultas Informatika,
Universitas Telkom

Bandung, Indonesia

³shintayulia@telkomuniversity.ac.id

Abstrak - Penelitian ini bertujuan untuk mengembangkan aplikasi Sistem Informasi Tekos menggunakan framework NextJS dengan menerapkan Single-Page Application (SPA) dan Client-Side Rendering (CSR). Metode pengujian yang digunakan adalah load time test untuk membandingkan waktu load halaman antara SPA CSR dan SPA SSR. Pengujian dilakukan pada environment local menggunakan browser Chrome pada perangkat Macbook dengan chip M1 Pro. Hasil pengujian menunjukkan bahwa SPA CSR memiliki waktu load halaman yang lebih cepat daripada SPA SSR pada kecepatan jaringan No Throttling/normal yaitu dibawah 5 detik. Namun, pada kecepatan jaringan Fast 3G dan Slow 3G baik SPA CSR ataupun SPA SSR memiliki kecepatan load lebih 5 detik yang artinya tidak sesuai dengan ketentuan NFR-01 pada dokumen perangkat lunak/SKPL. Pada aplikasi Sistem Informasi Tekos, terdapat beberapa halaman tidak dapat menerapkan SPA SSR karena membutuhkan data dinamis berdasarkan ID Owner, yang hanya bisa diimplementasikan pada CSR. Kesimpulannya, penggunaan SPA CSR dengan NextJS pada Sistem Informasi Tekos dapat meningkatkan kecepatan render halaman dan memberikan pengalaman pengguna yang lebih responsif. Namun, perlu pertimbangan khusus saat memilih metode render antara SPA CSR dan SPA SSR tergantung pada kebutuhan aplikasi dan kompleksitas data yang digunakan.

Kata Kunci: Sistem Informasi Tekos, Single Page Application (SPA), NextJS, Client-Side Rendering (CSR), Server-Side Rendering (SSR), Load Time Test

I. PENDAHULUAN

Tekos adalah start-up web yang membantu mahasiswa menemukan kosan atau kontrakan sesuai kebutuhan mereka. Fokus utama pada pengelolaan data yang efisien guna menampilkan informasi kosan dan kontrakan dengan cepat di telyukost.com. Aplikasi ini bertujuan memudahkan pencarian tempat tinggal dan pengelolaan data untuk penggunaannya.

Pada awal munculnya web, aplikasi web cenderung menggunakan model multi-page, namun

memiliki kelemahan pada responsivitas karena perpindahan halaman memengaruhi waktu render [1]. Solusinya adalah menggunakan Single Page Application (SPA), di mana browser hanya melakukan render halaman yang dibutuhkan, menghindari permintaan server berulang saat berinteraksi [2][3]. Penggunaan SPA meningkatkan kecepatan perpindahan halaman dan pengalaman pengguna karena mengurangi waktu loading [4].

Pengembangan aplikasi Sistem Informasi Tekos memanfaatkan framework JavaScript, NextJS, yang mendukung pembuatan website dengan konsep SPA. NextJS memungkinkan komponen yang sama pada halaman lain tidak perlu dimuat ulang, hanya tampilan yang berubah yang diperbarui [5]. Framework ini juga menyediakan fitur Client-Side Rendering (CSR) untuk meningkatkan interaktivitas website dengan kecepatan render yang lebih cepat dibandingkan Server-Side Rendering (SSR) [6] [7]. Dalam pengembangan aplikasi, panduan berupa Spesifikasi Kebutuhan Perangkat Lunak (SKPL) menjadi acuan penting yang mencakup functional dan non-functional requirement [8]. SKPL berperan krusial untuk memastikan perangkat lunak memenuhi persyaratan sebelum dilakukan pengembangan lebih lanjut [8].

II. KAJIAN PUSTAKA

A. Sistem Informasi berbasis Web

Website adalah sebuah halaman atau kumpulan halaman terkait yang dapat diakses melalui internet dan menyediakan berbagai informasi, konten, atau layanan. *Website* biasanya dapat diakses dengan menggunakan perangkat seperti komputer, tablet, atau *smartphone*. Informasi yang terdapat pada *website* dapat berupa teks, gambar, video, audio, atau gabungan dari semuanya. [9]

Menurut Yulia Djahir dan Dewi Pratita dalam bukunya yang berjudul “Bahan Ajar Sistem Informasi Manajemen”, Sistem informasi adalah suatu kumpulan dari elemen-elemen yang berinteraksi untuk menghasilkan informasi yang berguna bagi

penggunanya [10]. Pengembangan sistem informasi ditujukan untuk mengelola suatu informasi dan memudahkan dalam pengolahan informasi tersebut [11]. Pemanfaatan sistem informasi saat ini sudah banyak digunakan, baik dari instansi formal maupun non formal seperti contoh pengembangan aplikasi sistem informasi penjadwalan matakuliah, pengolahan nilai siswa, dan aplikasi lainnya yang berbasis web [11], [12].

B. Frontend

Frontend merujuk pada pembangunan visual antarmuka pengguna pada sebuah situs web, yang melibatkan penggunaan HTML, CSS, serta JavaScript. Tujuannya adalah agar pengguna mampu mengamati dan berinteraksi dengan isi situs web tersebut [13]. *Frontend developer* adalah seorang spesialis dalam pengembangan aplikasi web atau mobile yang bertanggung jawab untuk merancang dan mengembangkan bagian depan atau tampilan (*user interface*) dari sebuah *website* atau aplikasi [14].

C. NextJS

Next.js adalah sebuah kerangka kerja (*framework*) atau *platform* pengembangan web yang *open-source* untuk mengembangkan aplikasi web modern berbasis React yang dibuat oleh Zeit [15]. Next.js mempunyai sejumlah fitur yang dapat memudahkan pengembangan aplikasi web yang efisien dan cepat seperti Client-Side Rendering (CSR) dan Server-Side Rendering (SSR). CSR menggunakan *useEffect* pada React Hooks untuk melakukan pengambilan data dari sisi klien, dengan CSR maka data dari API akan diambil setiap kali terjadi permintaan halaman dari sisi klien (Setelah halaman *dirender*, maka data pengambilan dilakukan) [16]. Sedangkan SSR menggunakan suatu fungsi khusus yaitu *getServerSideProps* untuk melakukan render data dinamis dari *server* yang hanya bisa berjalan di sisi *server* [17]. Tidak seperti CSR, SSR tidak dapat menampilkan halaman selama proses pengambilan data dilakukan [18].

D. Single-Page Application (SPA) dan Client-Side Rendering (CSR)

Single Page Application (SPA) adalah pendekatan pengembangan aplikasi web di mana seluruh konten aplikasi dimuat ke dalam satu halaman web yang sama [19]. Dalam SPA, ketika pengguna berinteraksi dengan aplikasi, konten yang berubah diperbarui secara dinamis pada halaman yang sama tanpa memuat ulang seluruh halaman sehingga menciptakan pengalaman pengguna yang lebih responsif karena tidak ada jeda yang terlihat saat pindah halaman [20]. SPA biasanya dibangun menggunakan JavaScript untuk mengelola manipulasi dan perubahan konten di sisi klien [21].

Client-Side Rendering (CSR) adalah suatu pendekatan dalam pengembangan aplikasi web di mana *rendering* (proses mengubah kode menjadi tampilan yang terlihat oleh pengguna) dilakukan di sisi klien, yaitu pada *browser* pengguna [6]. Keuntungan dari CSR adalah pengalaman pengguna

yang lebih cepat setelah halaman awal diunduh [7]. Hal ini memberikan respon yang lebih baik, terutama dalam aplikasi yang sangat interaktif seperti SPA yang sering digunakan saat ini.

E. Load Time Test

Load time Test adalah suatu pengukuran waktu dari inialisasi halaman website hingga memunculkan keseluruhan konten halaman [22]. Kecepatan dalam proses *render* pada halaman ditentukan oleh bagaimana kode *frontend* dibuat, penggunaan *cache browser*, dan jaringan yang digunakan ketika mengakses halaman [23]. Salah satu browser yang memiliki fitur untuk mendukung pengujian load adalah Chrome dimana terdapat fitur *developer* yang terdapat pada Network pada *chrome tools* untuk mengetahui waktu *load* atau *render* terhadap halaman yang diakses [24].

III. PERANCANGAN MODEL

A. Perancangan Model Sistem

GAMBAR 3. 1 merupakan perancangan arsitektur SPA pada aplikasi Sistem Informasi Tekos. Pada GAMBAR 3. 1 menjelaskan beberapa komponen didalamnya antara lain:

- a. Side Navigation (Merah) dan Top Navigation (Hijau), yang digunakan untuk berpindah antar halaman. Side Navigation dan Top Navigation bersifat statis
- b. Halaman Dinamis (Biru), yang digunakan untuk memunculkan halaman yang ingin digunakan oleh *user*.

Ketika user berpindah halaman misal dari Halaman A ke Halaman B yang mempunyai komponen Side Navigation dan Top Navigation yang sama, browser hanya akan memuat halaman yang bersifat dinamis saja tanpa memuat ulang secara keseluruhan tampilan (tidak memuat ulang Side Navigation dan Top Navigation).

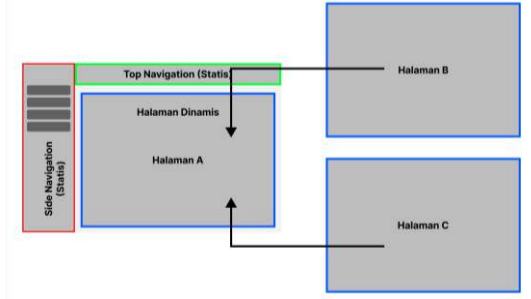
Penggunaan Client-Side Rendering (CSR) pada arsitektur SPA terletak pada halaman dinamis yang ditampilkan. Lihat

GAMBAR 3. 2. Berikut merupakan penjelasan cara kerja CSR pada halaman dinamis:

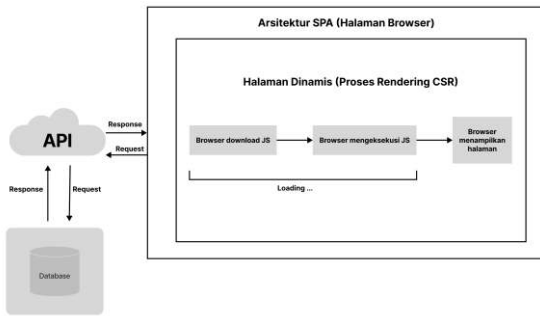
1. *Request* dan *Response*
Halaman yang membutuhkan data dinamis akan melakukan *request* melalui API pada database. Database mengirimkan *response* berupa data JSON melalui API dan dikirim ke *browser*. Ketika proses ini browser akan mengalami *loading*.
2. Halaman Dinamis (Proses Rendering CSR)
 - a. Browser mendownload JS
Setelah *server* mengirim respon berisi data, Browser akan mendownload javascript yang dibutuhkan untuk dieksekusi pada halaman tersebut. Tahap ini *browser* masih mengalami *loading*.
 - b. Browser mengeksekusi JS

Browser mengeksekusi javascript yang telah siap dan berisi data pada halaman tersebut. Tahap ini browser masih mengalami *loading*.

c. Browser menampilkan Halaman
 Browser menampilkan halaman yang mempunyai data dari server dan dapat digunakan untuk berinteraksi dengan user.



GAMBAR 3.1
 Arsitektur SPA



GAMBAR 3.2
 Cara Kerja CSR

IV. HASIL DAN PEMBAHASAN

A. Hasil

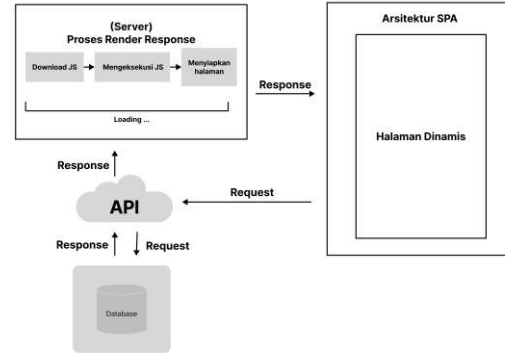
Pengembangan aplikasi Sistem Informasi Tekos yang memenuhi *functional requirement* pada dokumen SKPL terdapat pada tinyurl.com/DokumenPengembanganSISFOTEKOS yang terdiri dari penerapan fungsional *role* Owner yang ditandai dengan kode FR2-01 s/d FR2-14 dan *role* Admin yang ditandai dengan kode FR3-01 s/d FR3-08. Di dalam dokumen tersebut telah dijelaskan mengenai pengembangan fungsionalitas serta pengujian unit yang dilakukan, sehingga selanjutnya pengujian terhadap kebutuhan non-fungsionalitas yang berkaitan dengan proses pengembangan kode dan *asset* yang memengaruhi kecepatan *render* aplikasi yaitu NFR-01.

Untuk menganalisis berapa lama waktu dalam melakukan *render* terhadap tampilan halaman yang di akses oleh *user*, dilakukan pengujian *load time test* pada aplikasi Sistem Informasi Tekos yang menerapkan Single Page Application dan Client-Side Rendering (CSR). Dalam melakukan pengujian terhadap waktu *load* pada aplikasi ini, perlu pembandingan dimana aplikasi ini menerapkan

kebalikan dari CSR yaitu Server-Side Rendering (SSR).

Cara kerja SSR dapat dilihat pada GAMBAR 4. 1:

1. Ketika aplikasi melakukan *request* data pada API, API akan mengirimkan *response* pada *server*
2. *Server* melakukan semua proses pembuatan konten, dan melakukan *render* halaman web lengkap sebagai *respons* untuk ketika proses *request* terjadi
3. *Browser* menerima *response* dari *server* dan menampilkan halaman web yang sudah jadi kepada *pengguna*.



GAMBAR 4.1
 Cara Kerja SSR

TABEL 4. 1 merupakan hasil uji waktu *load* halaman yang menerapkan SPA CSR. Sedangkan

TABEL 4. 2 merupakan hasil uji *load* halaman yang menerapkan SPA SSR.

TABEL 4. 1
 Load Time Test SPA CSR

Nama Halaman	No Thorotting/nor mal	Fast 3G	Slow 3G
Dashboard Admin	4.81s	41s	>60s
Lihat Kosan Admin	2.34s	20.63s	>60s
Lihat Kontrakan Admin	2.56s	21.14s	>60s
Lihat Nearby	82ms	919ms	3.24s
Dashboard Owner	2.24s	38.51ms	>60s
Lihat Kosan Owner	1.17s	19.65s	>60s
Lihat Kontrakan Owner	3.4s	19.60s	>60s
Lihat Detail Kosan Owner	3.4s	37.4s	>60s

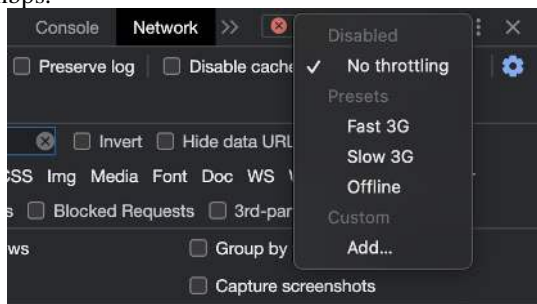
TABEL 4. 2
 Load Time Test SPA SSR

Nama Halaman	No Thorotting/normal	Fast 3G	Slow 3G
Dashboard Admin	7.53s	>60s	>60s
Lihat Kosan Admin	6.77s	48.68s	>60s

Lihat Kontrakan Admin	6.93s	50.95s	>60s
Lihat Nearby	2.72s	33.71s	>60s
Dashboard Owner	SPA CSR	SPA CSR	SPA CSR
Lihat Kosan Owner	SPA CSR	SPA CSR	SPA CSR
Lihat Kontrakan Owner	SPA CSR	SPA CSR	SPA CSR
Lihat Detail Kosan Owner	SPA CSR	SPA CSR	SPA CSR

B. Pembahasan

Pengujian SPA CSR dan SPA SSR menggunakan browser Chrome Version 115.0.5790.102 (Official Build) (arm64) yang memiliki fitur Networking. Pada fitur Networking ini terdapat konfigurasi untuk mengatur kecepatan suatu internet yaitu No Throttling/normal, Fast 3G, dan slow 3G. Lihat GAMBAR 4. 2. Pengujian dilakukan pada device Macbook dengan chip M1 Pro 2023 tanggal 23 Juli 2023 dengan menggunakan kecepatan internet 10 mbps.



GAMBAR 4. 2 Konfigurasi Jaringan Chrome

Dapat dilihat pada pengujian sebelumnya, penggunaan metode SPA CSR dalam melakukan render terhadap halaman website Sistem Informasi Tekos pada kondisi No Throttling mempunyai waktu load kurang dari 5 detik. Sedangkan pada kondisi lainnya melebihi ekspektasi pada Non-Functional Requirement pada NFR-01. Namun pada

TABEL 4. 2, terdapat beberapa halaman yang hanya bisa menerapkan SPA CSR. Kondisi ini dikarenakan bahwa pada halaman-halaman tersebut membutuhkan ID Owner untuk menyeleksi data berdasarkan ID Owner pada data master yang hanya bisa diimplementasikan pada CSR. Lihat GAMBAR 4. 3.

```
useEffect(() => {
  if (userDetails) {
    dispatch(getKosansByOwnerId(userDetails?.data?._id));
  }
}, [dispatch, userDetails?.data?._id]);
```

GAMBAR 4. 3 CSR dengan ID Owner

Ketika pengambilan data dari API pada kondisi owner, dilakukan request terhadap server yang mengikutsertakan ID Owner sebagai parameter. Pengimplementasian ini hanya bisa dilakukan menggunakan CSR (ditandai dengan useEffect). Pada SSR proses untuk mendapatkan ID Owner tidak dapat dilakukan karena sisatem kerja SSR akan dieksekusi terlebih dahulu sebelum halaman dimunculkan, sedangkan pada CSR, proses pengambilan ID Owner dan halaman dieksekusi sekuensial pada satu fungsi.

Kecepatan dalam melakukan render pada halaman menggunakan SPA CSR lebih cepat. Hal ini dikarenakan CSR hanya melakukan render halaman atau komponen dinamis (Biru) ketika pengguna berpindah halaman yang memiliki arsitektur yang sama statis. Dalam hal ini adalah Side Navigation (Merah) dan Top Navigation (Hijau). Lihat GAMBAR 4. 4.



GAMBAR 4. 4 Halaman SPA

Sedangkan, kecepatan render halaman menggunakan SSR lebih lama dikarenakan SSR melakukan render keseluruhan halaman dari awal termasuk komponen statis. Proses ini terjadi pada sisi server dimana server akan menyiapkan tampilan yang di dalamnya terdapat data dinamis sehingga setelah siap, tampilan tersebut menjadi statis.

V. KESIMPULAN

Aplikasi Sistem Informasi Tekos merupakan suatu aplikasi yang dapat membantu pemilik kosan atau kontrakan untuk mengelola data kosan atau kontrakan yang dimiliki. Dari hasil uji dan analisis, didapatkan:

- A. Aplikasi Sistem Informasi Tekos dikembangkan menggunakan framework Next.js yang menerapkan konsep Single Page Application (SPA) dan Client-Side Rendering (CSR). SPA memungkinkan pengguna untuk berinteraksi dengan aplikasi tanpa harus melakukan refresh halaman secara keseluruhan, sementara CSR memungkinkan rendering konten dilakukan di sisi klien (browser) sehingga meningkatkan responsivitas dan pengalaman pengguna..
- B. Dari hasil load time test, terlihat bahwa SPA CSR mampu memberikan waktu load halaman yang cukup cepat ketika dalam kondisi No Throttling, yaitu di bawah 5 detik sesuai dengan non-

functional requirement. Namun, saat menghadapi kecepatan internet yang lebih lambat seperti Fast 3G atau Slow 3G, waktu load halaman menjadi lebih lambat dan ada beberapa halaman yang bahkan melebihi batas waktu yang diinginkan.

REFERENSI

- [1] J. Voutilainen, "Evaluation of Front-end JavaScript Frameworks for Master Data Management Application Development," *Metropolia*, no. 12, 2017.
- [2] S. Deshmukh, D. Mane, and A. Retawade, "Building a single page application web front-end for e-learning site," in *Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019*, 2019. doi: 10.1109/ICCMC.2019.8819703.
- [3] E. Molin, "Comparison of Single Page Application Frameworks," *KTH Computer Science and Communication*, 2016.
- [4] M. F. Santoso, "TEKNIK SINGLE PAGE APPLICATION (SPA) LAYOUT WEB DENGAN MENGGUNAKAN REACT JS DAN BOOTSTRAP," *Jurnal Khatulistiwa Informatika*, vol. 9, no. 2, 2021, doi: 10.31294/jki.v9i2.11357.
- [5] "Routing: Pages and Layouts | Next.js." <https://nextjs.org/docs/pages/building-your-application/routing/pages-and-layouts> (accessed Jul. 21, 2023).
- [6] "Rendering: Client-side Rendering (CSR) | Next.js." <https://nextjs.org/docs/pages/building-your-application/rendering/client-side-rendering> (accessed Jul. 08, 2023).
- [7] T. Fadhilah Iskandar, M. Lubis, T. Fabrianti Kusumasari, and A. Ridho Lubis, "Comparison between client-side and server-side rendering in the web development," in *IOP Conference Series: Materials Science and Engineering*, 2020. doi: 10.1088/1757-899X/801/1/012136.
- [8] T. Hovorushchenko and O. Pomorova, "Methodology of evaluating the sufficiency of information on quality in the software requirements specifications," *Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies, DESSERT 2018*, pp. 370–374, Jul. 2018, doi: 10.1109/DESSERT.2018.8409161.
- [9] S. G. M. Gumolung, B. N. N. Xaverius, and A. S. M. Lumenta, "Analisa Teknologi Hyper Text Markup Language (HTML) Versi 5," *Jurnal Teknik Informatika*, vol. 15, no. 3, 2020.
- [10] Y. Djahir and D. Pratita, *bahan Ajar Sistem Informasi Manajemen*. 2014.
- [11] F. Suryandani, B. Basori, and D. Maryono, "PENGEMBANGAN SISTEM INFORMASI AKADEMIK BERBASIS WEB SEBAGAI SISTEM PENGOLAHAN NILAI SISWA DI SMK NEGERI 1 KUDUS," *Jurnal Ilmiah Pendidikan Teknik dan Kejuruan*, vol. 10, no. 1, 2017, doi: 10.20961/jiptek.v10i1.14976.
- [12] I. A. Ramadhani, "Pengembangan Sistem Informasi Penjadwalan Mata Kuliah Berbasis Web Di Fakultas Teknik Universitas Negeri Makassar," *Jurnal Pendidikan*, vol. 6, no. 2, 2018, doi: 10.36232/pendidikan.v6i2.36.
- [13] S. Mufti Prasetyo, M. Ivan Prayogi Nugroho, R. Lima Putri, and O. Fauzi, "Pembahasan Mengenai Front-End Web Developer dalam Ruang Lingkup Web Development," *Jurnal Multidisiplin Ilmu*, vol. 1, no. 6, 2022.
- [14] P. P. Arhandi, Y. Pramitarini, and R. Alviandra, "Desain Prototype Frontend Auto Generator Based On REST API," in *Seminar Informatika Aplikatif Polinema (SIAP)*, 2019.
- [15] "Next.js by Vercel - The React Framework for the Web." <https://nextjs.org/> (accessed May 09, 2023).
- [16] M. Fariz, S. Lazuardy, and D. Anggraini, "Modern Front End Web Architectures with React.Js and Next.Js," *International Research Journal of Advanced Engineering and Science*, vol. 7, no. 1, 2022.
- [17] H. A. Jartarghar, G. R. Salanke, A. K. A.R, S. G.S, and S. Dalali, "React Apps with Server-Side Rendering: Next.js," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 14, no. 4, 2022.
- [18] Herman and A. Geovanny, "ANALISIS RENDERING PERFORMA ANTARA SERVER SIDE DAN CLIENT SIDE PADA WEB APPLICATION," *JURNAL ILMIAH BETRIK : Besemah Teknologi Informasi dan Komputer*, vol. 13, no. 3, 2022.
- [19] M. A. Jadhav, B. R. Sawant, A. Deshmukh, and N. Mumbai, "Single Page Application using AngularJS," *International Journal of Computer Science and Information Technologies*, 2015.

- [20] M. Kaluža and B. Vukelić, "Comparison of front-end frameworks for web applications development," *Zbornik Veleučilišta u Rijeci*, vol. 6, no. 1, 2018, doi: 10.31784/zvr.6.1.19.
- [21] R. John Joseph, "Single Page Application and Canvas Drawing," *International journal of Web & Semantic Technology*, vol. 6, no. 1, 2015, doi: 10.5121/ijwest.2015.6103.
- [22] B. Pourghassemi, A. Amiri Sani, and A. Chandramowliswaran, "What-If Analysis of Page Load Time in Web Browsers Using Causal Profiling," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 3, no. 2, 2019, doi: 10.1145/3341617.3326142.
- [23] R. Oktrifianto, D. Adhipta, and W. Najib, "Page Load Time Speed Increase on Disease Outbreak Investigation Information System Website," *IJITEE (International Journal of Information Technology and Electrical Engineering)*, vol. 2, no. 4, 2019, doi: 10.22146/ijitee.46599.
- [24] "Chrome Developers." <https://developer.chrome.com/> (accessed Aug. 10, 2023).