

Analisa Peformasi Metode Rendering Website: Client Side, Server Side, dan Incremental Static Regeneration

Roni Ardiyanto^{1*}, Eka Ardhianto²

^{1,2}Program Studi Teknik Informatika, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank
Jl. Trilomba Juang No 1, Semarang, Jawa Tengah 5024, Indonesia

e-mail: roniardiyanto@mhs.unisbank.ac.id, ekaardhianto@edu.unisbank.ac.id

(*) Corresponding Author

Artikel Info : Diterima : 15-09-2023 | Direvisi : 01-11-2023 | Disetujui : 05-01-2024

Abstrak - *Website* telah menjadi bagian penting dari kegiatan bisnis, pendidikan, hiburan, maupun media sosial. Optimalisasi website menjadi penting saat proses pembuatannya. Hal ini disebabkan supaya pengguna website merasa nyaman. Akses website yang cepat dan ringan menjadi salah satu daya tarik pengguna. Metode pembuatan website pada bagian *Front-End* diklasifikasikan menjadi 3 macam, yaitu: metode render CSR (*Client Side Rendering*), metode render SSR (*Server Side Rendering*), dan metode render ISR (*Incremental Static Regeneration*). Penelitian ini menggunakan aplikasi website dengan *framework* Next.js sebagai website untuk pengujian. Dibuat dalam website satu halaman berisi list daftar berita dari API CNNIndonesia.com. Tujuan penelitian ini untuk mencari metode mana yang optimal digunakan dalam pengembangan website. Sebagai metrik performansi digunakan kecepatan dan ukuran website. Hasil yang diperoleh adalah bahwa metode ISR sangat cocok digunakan dalam pengembangan web frontend karena lebih ringan dan lebih cepat diakses dibanding metode SSR dan CSR. Dengan rekomendasi untuk pembuatan website, jika menginginkan kecepatan berarti dapat menggunakan metode ISR untuk peramban chrome jika menggunakan mozilla SSR. Jika menginginkan *generate* yang ringan maka menggunakan metode ISR untuk mozilla dan ISR untuk chrome.

Kata Kunci : Website, Rendering, Analisis

Abstracts - *Websites* have become an important part of business, education, entertainment, and social media activities. Website optimization becomes important during the creation process. This is so that website users feel comfortable. Fast and lightweight website access is one of the user attractions. Website creation methods in the *Front-End* section are classified into 3 types, namely: CSR (*Client Side Rendering*) render method, SSR (*Server Side Rendering*) render method, and ISR (*Incremental Static Regeneration*) render method. This research uses a website application with the *Next.js* framework as a website for testing. Created in a one-page website containing a list of news lists from the CNNIndonesia.com API. The purpose of this study is to find which method is optimally used in website development. As a performance metric, website speed and size are used. The result is that the ISR method is very suitable for use in frontend web development because it is lighter and faster to access than SSR and CSR methods. With recommendations for website creation, if you want speed means you can use the ISR method for chrome browsers if you use mozilla SSR. If you want to generate a lightweight then use the ISR method for mozilla and ISR for chrome.

Keywords : Website, Rendering, Analysis

PENDAHULUAN

Interconnection-networking (Internet) merupakan sebuah sistem global jaringan komputer yang saling menghubungkan antara satu dengan yang lain di seluruh penjuru(Wahdaniya & Sariwati, 2022). Internet dapat dianalogikan sebagai perpustakaan besar yang didalamnya terdapat beragam macam informasi yang dibutuhkan masyarakat (Apriyanti et al., 2022). Informasi yang dibagikan melalui internet ini dapat dibagikan melalui laman *website*. *Website* digunakan sebagai wadah untuk menampung informasi yang dapat diakses dan digunakan oleh masyarakat di penjuru dunia menggunakan internet melalui aplikasi peramban (browser). *Website* adalah sebuah perangkat lunak yang terdiri dari halaman-halaman situs yang dapat menyajikan konten berupa teks, audio dan video kepada pengguna yang mana halaman-halaman situs ini memiliki akses domain dalam *worldwide* dalam



internet (Budikusuma & Susanto, 2022) Untuk mengakses laman *website* diperlukan aplikasi peramban web seperti: Google Chrome, Mozilla Firefox, dan perangkat komputer yang terkoneksi dengan internet.

Pada perkembangannya hingga saat ini, secara pendekatan data dan tampilannya, *website* dibagi menjadi 2 macam, yaitu : *website* dinamis dan *website* statis. *Website* dinamis merupakan jenis situs *web* yang secara rutin diperbarui oleh pemilik atau administratornya dengan konten yang terus diperbaharui (Furqani & Muliono, 2021). Sehingga konten dan data yang ada didalamnya bisa berubah-ubah secara terus-menerus jika diperlukan dan dapat dimanfaatkan secara berkelanjutan. *Website* dinamis dalam pemanfaatannya cocok digunakan untuk: aplikasi sistem informasi, *website* berita, *website* sistem manajemen, dan *website* lainnya yang membutuhkan banyak aksi untuk memanipulasi data. Kelebihan dari jenis *website* dinamis ini adalah: cocok digunakan untuk sistem yang kompleks, dan skalabilitas penggunaan lebih luas. Sedangkan, kekurangannya adalah bahwa Pengerjaan yang lebih kompleks. *Website* statis adalah *website* yang bersifat tidak mudah diubah isinya oleh pengguna (Ronaldo & Pasha, 2021). *Website* yang cocok untuk menerapkan jenis ini adalah : *website* profil perusahaan dan *website* informasi undangan. Karena sifatnya statis, jika ingin merubah data yang ditampilkan pada *website* nantinya, perlu untuk merubahnya pada kode program yang sudah dibuat. Kelebihan dari jenis *website* statis ini adalah : 1) Lebih cepat dalam akses *website* di peramban web, 2) Codingan yang dibuat tidak terlalu kompleks, dan 3) Tidak menggunakan database dalam pembuatannya. Sedangkan, kekurangannya adalah : 1) Jika ingin membuat perubahan konten, harus menulis langsung pada kode programnya, dan 2) Tidak interaktif dua arah antara pengguna dan *website*.

Dalam perkembangannya *website* telah terbagi menjadi 2 dalam jenis pengembangannya yaitu : *Front-end* dan *Back-end*. Pembagian ini bertujuan untuk memisahkan beban dan pengoptimalan dalam proses pengembangan suatu *website*. *Front-end* adalah bagian dari sistem yang bertanggung jawab untuk menampilkan tampilan aplikasi yang digunakan oleh pengguna untuk berinteraksi dengan sistem (Terttiaavini et al., 2023). Pada *Front-end* inilah bagaimana tampilan dan interaksi pada halaman *website* diimplementasi dan disajikan kepada pengguna pada peramban web. Sedangkan *Back-end* adalah bagian dari sistem yang bertanggung jawab dalam pengolahan data dan informasi yang berasal dari *Front-end*. *Backend* umumnya bertanggung jawab atas segala jenis proses yang tidak bersentuhan langsung dengan pengguna, seperti pengelolaan server dan basis data (Pangestika & Dirgahayu, 2020).

Tanggung jawab dari proses *website rendering* terletak pada *front-end*. *Website rendering* mengacu pada proses menghasilkan tampilan visual dari sebuah halaman *web* di browser atau perangkat lainnya dari konten dan data yang statis ataupun dinamis yang berasal dari *back-end*. Ketika pengguna mengunjungi sebuah situs web, aplikasi peramban *web* perlu mengambil konten HTML, CSS, JavaScript, dan sumber daya lainnya, lalu menggabungkannya untuk membuat tampilan yang dapat dilihat dan diinteraksikan oleh pengguna. Ketika berbicara tentang *website* yang bagus, tentunya kualitas ada hal yang penting dan wajib diperhatikan (Biyaypu et al., 2023).

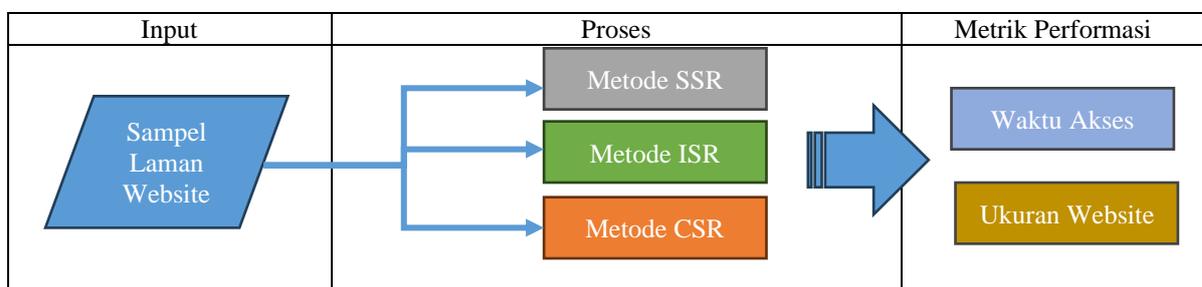
Ada 3 metode yang dalam proses *website rendering* yang digunakan dalam penelitian yaitu : 1) *Client-Side Rendering* adalah salah satu jenis pengembangan perangkat lunak pada aplikasi berbasis web yang menangani setiap halaman di peramban web pengguna (Fadhilah Iskandar et al., 2020). Sehingga segala proses rendering *website* dibebankan pada peramban web pengguna. 2) *Server-side rendering* adalah jenis *rendering* yang proses *rendering* dilakukan oleh server. Dalam konteks ini, kita harus melakukan proses *rendering* semua elemen HTML sebelum tampilan tersebut disajikan di peramban *web*, dan akhirnya, kita perlu membuat sebuah fungsi ekstra dalam aplikasi kita untuk melakukan proses *rendering* ini di sisi *server* (*Server-Side Rendering*) (Yunanto et al., 2022). 3) *Incremental static regeneration* adalah jenis *rendering* yang merupakan kombinasi dari SSG dan SSR. Halaman dihasilkan statis saat pembangunan awal, tetapi bagian-bagian tertentu dapat diperbarui secara dinamis dalam interval waktu tertentu. Ini menggabungkan keuntungan pembuatan situs web statis dengan kemampuan pembaruan terkini, menjaga kecepatan dan kinerja tanpa perlu rendering server setiap saat.

Tujuan dari penelitian ini adalah menganalisa dari perbandingan ketiga metode rendering tersebut dari sisi ukuran *website* dan kecepatan akses *website* pada peramban web. Sebagai metrik performansi digunakan waktu loading dan ukuran file dalam mengakses *website*. Penelitian ini bersifat eksperimental dengan melakukan beberapa percobaan. Jumlah percobaan dalam penelitian ini 28 kali untuk setiap metode. Hasil yang diperoleh adalah bahwa metode ISR sangat cocok digunakan dalam pengembangan web frontend karena lebih ringan dan lebih cepat diakses dibanding metode SSR dan CSR. Artikel ini disajikan dalam beberapa bagian. Bagian awal berisi pendahuluan. Bagian kedua berisi metode penelitian. bagian ketiga berisi hasil dan pembahasan, dan pada akhir artikel sebagai kesimpulan.

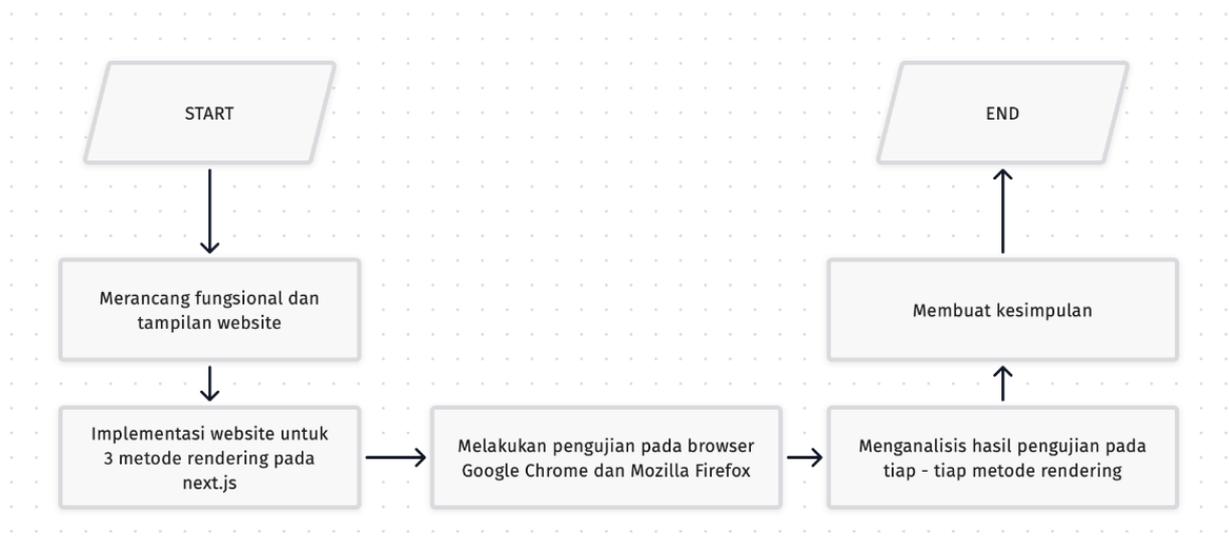
METODE PENELITIAN

Metode penelitian yang peneliti gunakan adalah eksperimental. Penelitian eksperimen adalah metode penelitian yang berusaha mengidentifikasi hubungan sebab-akibat antara variabel yang independen dan variabel yang dependen, dengan mengendalikan serta memanipulasi variabel independen secara sengaja. Percobaan adalah kegiatan yang dimaksudkan untuk mengungkapkan efek atau dampak dari perlakuan tertentu pada suatu fenomena atau perubahan yang terjadi sebagai hasil dari perlakuan tersebut (Abraham & Supriyati, 2022). Sebagai bahan input percobaan, digunakan laman *website* yang berisi berita dari laman CNNIndonesia.com. Laman *website* yang

digunakan sebagai bahan percobaan memuat 10 artikel berita dari CNNIndonesia.com menggunakan API *crawling* yang dikombinasikan dengan metode *rendering website*. Metode *rendering website* yang digunakan dalam percobaan adalah CSR, SSR, dan ISR. Penelitian ini mengukur kecepatan akses dan ukuran *website*. Gambar 1 menunjukkan kerangka penelitian yang dilakukan. Dalam pelaksanaan penelitian, sebagai alat akses *website* digunakan aplikasi browser berupa Google Chrome dan Mozilla Firefox seperti terlihat pada gambar 2. (Muhammad & Muhammad, 2019)



Sumber: Penelitian (2023)
Gambar 1. Kerangka Penelitian.

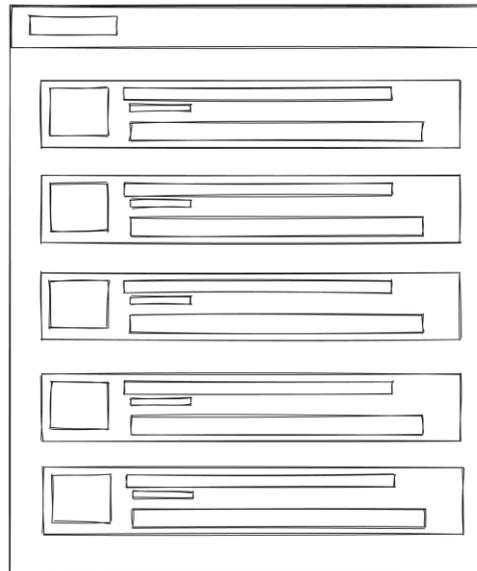


Sumber: Penelitian (2023)
Gambar 2. Tahapan Alur Penelitian.

1. Perancangan Sampel Website

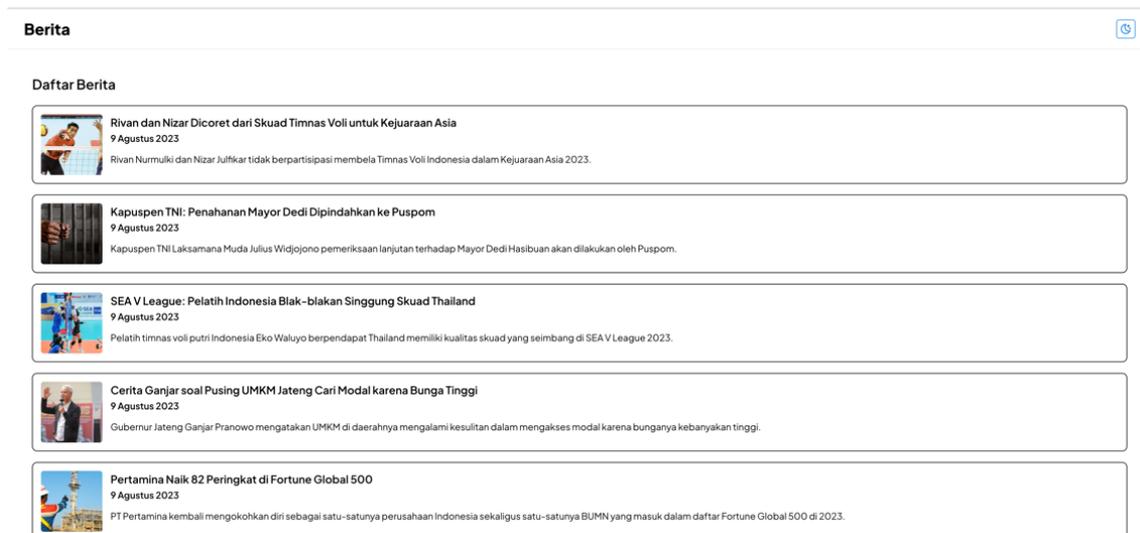
Laman *website* yang digunakan pada penelitian ini dirancang seperti terlihat pada gambar 3. Pada tahap perancangan *website*, sebagai permulaan akan dibuat *wireframe* awal tentang bagaimana layout dan isi dari *website* yang nantinya dibuat. *Wireframe* adalah representasi awal atau kerangka dasar yang digunakan sebelum merancang halaman *web* atau antarmuka aplikasi (Hartawan & Id, 2022). berdasarkan desain pada gambar 3, bentuk implementasi diperlihatkan pada gambar 4, sebagai frontend *website*. Pada sampel *website* ini sudah tertanam API untuk proses *crawling* di laman *website* CNNIndonesia.com dengan tiga metode *rendering* (SSR,

CSR dan ISR).



Sumber: Penelitian (2023)

Gambar 3. Rancangan *Wireframe* sampel laman website.



Sumber: Penelitian (2023)

Gambar 4. Tampilan sampel laman *website*

2. Implementasi Desain

Pada tahap ini, desain UI / UX yang sudah dibuat untuk diimplementasikan dalam bentuk website, dibuat 1 buah website dengan fungsionalitas dan desain yang serupa untuk masing – masing metode : CSR, SSR, dan ISR. Dalam pembuatannya, masing - masing website akan dibuat menggunakan *framework* Next.js. Next.js adalah sebuah kerangka kerja yang memiliki fleksibilitas tinggi (Baehaqi et al., 2023). Alasan penggunaan Next.js karena Next.js secara penuh mendukung metode CSR, SSR, dan ISR dalam pengembangannya, serta merupakan *framework web* modern yang dapat digunakan untuk membuat aplikasi website yang cepat (Jubhari Phie Joarno et al., 2022).

a. Implementasi Metode CSR

Pada metode CSR, proses mengambil data berita yang tampil pada website, dibuat dalam fungsi *handleFetch()* yang terlampir pada gambar 5. Fungsi ini dieksekusi pada peramban internet pengguna dengan memanfaatkan fungsi *useEffect()*. Fungsi *useEffect* adalah fungsi bawaan ReactJS yang akan dijalankan setiap kali terjadi perubahan pada dependensinya atau hanya pada beberapa tahap siklus hidup komponennya (Mashhadi Mohammad & Valentina, 2023). Pada fungsi *handleFetch()* data berita diambil secara asinkronus,

kemudian setelah data dari API berita sudah terambil akan diberikan ke variabel *res* yang nantinya data – data berita digunakan untuk pada proses *render* website.

Pada konstanta *fetch* dilakukan pengambilan data dari API CNNIndonesia.com. Kemudian hasilnya disimpan dalam konstanta *res* dengan bentuk data JSON (JavaScript Object Notation). Dari hasil pada konstanta *res*, data akan dimasukkan kedalam variabel *state res* dengan fungsi *setRes()*.

```
const handleFetch = async () => {
  const fetch = await fetch(`${url}/cnn-news/`);
  const res = await fetch.json();
  setRes(res?.data);
};

useEffect(() => {
  handleFetch();
}, []);
```

Sumber: Penelitian (2023)

Gambar 5. Kode implementasi *Client Side Rendering*

b. Implementasi Metode SSR

Pada metode *rendering* website SSR, dibuat dengan memanfaatkan fungsi khusus dari *Next.js* yang disebut *getServerSideProps* yang terlampir pada gambar 6. Fungsi *getServerSideProps* hanya berjalan di sisi server dan tidak pernah berjalan di peramban (Fariz et al., 2022). Pengambilan data pada metode *rendering* ini dilakukan oleh server dan langsung diberikan kepada pengguna (Jartarghar et al., 2022).

Pada konstanta *fetch* dilakukan pengambilan data dari API CNNIndonesia.com. Kemudian hasilnya disimpan dalam konstanta *res* dengan bentuk data JSON (JavaScript Object Notation). Dari hasil pada konstanta *res*, data akan di balikkan / *return* kedalam atribut *res* pada object *props*.

```
export async function getServerSideProps() {
  const fetch = await fetch(`${url}/cnn-news/`);
  const res = await fetch.json();
  return {
    props: {
      res: res.data,
    },
  };
}
```

Sumber: Penelitian (2023)

Gambar 6. Kode Implementasi *Server Side Rendering*

c. Implementasi Metode ISR

Pada metode ISR terdapat fungsi khusus dari *Next.js* yaitu *getStaticProps()* seperti yang terdapat pada gambar 7. Metode ini berjalan pada sisi *server* dalam pengambilan data dan proses *render* datanya. Proses *render* ini tidak dilakukan setiap ada permintaan dari pengguna. Tetapi hanya akan dirender ulang ketika masa kadaluarsa yang dinyatakan dalam satuan detik pada nilai *revalidate* telah terpenuhi. Tidak seperti metode SSR yang hasil *render* diperuntukkan untuk khusus untuk pengguna yang mengakses. Metode ISR hasil dari *render*nya digunakan oleh semua pengguna, dan hanya dipicu proses *render* ini oleh satu pengguna.

Pada konstanta *fetch* dilakukan pengambilan data dari API CNNIndonesia.com. Kemudian hasilnya disimpan dalam konstanta *res* dengan bentuk data JSON (JavaScript Object Notation). Dari hasil pada konstanta *res*, data akan di balikkan / *return* kedalam atribut *res* pada object *props*.

```
export async function getStaticProps() {
  const fetch = await fetch(`${url}/cnn-news/`);
  const res = await fetch.json();
  return {
    props: {
      res: res.data,
    },
  };
}
```

```

    revalidate: 30,
  };
}

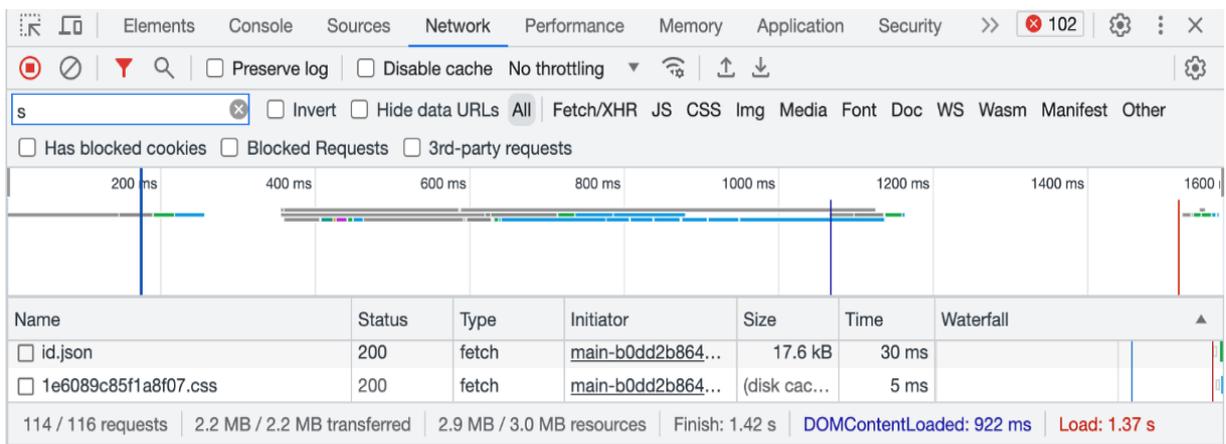
```

Sumber: Penelitian (2023)

Gambar 7. Kode Implementasi *Incremental Static Regeneration*

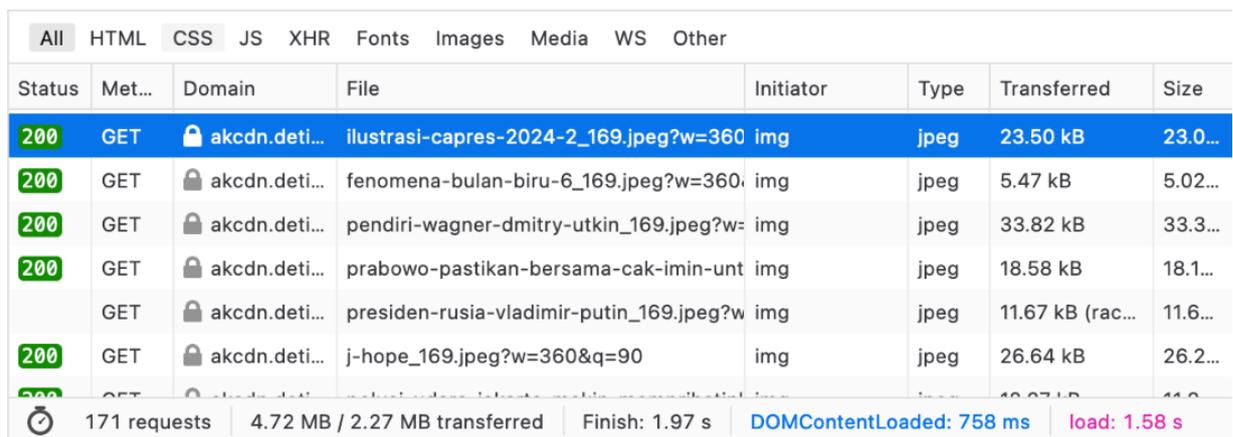
3. Pengujian

Pengujian dilakukan menggunakan peramban internet Google Chrome dan Mozilla Firefox dengan memanfaatkan fitur developer tools. Pada pengujian yang nanti dilakukan, akan memanfaatkan fitur Networks dan melihat ukuran website pada informasi “resources” dan kecepatan akses website pada informasi “finish”. Tampilan *developer tools* diperlihatkan pada gambar 8 untuk aplikasi peramban (browser) Google Chrome dan pada gambar 9 untuk aplikasi peramban (browser) Mozilla Firefox.



Sumber: Penelitian (2023)

Gambar 8. Tampilan *developer tools* pada aplikasi peramban Google Chrome



Sumber: Penelitian (2023)

Gambar 9. Tampilan *developer tools* pada aplikasi peramban Mozilla Firefox

Pada pengujiannya peneliti melakukan pengujian selama 7 hari, dalam tiap harinya dilakukan 4 kali pengujian yang dilaksanakan pada jam (WIB): 09.00, 12.00, 15.00, dan 21.00 untuk tiap metode rendering website pada browser Google Chrome dan Mozilla Firefox. Dari hasil yang diperoleh dalam pengujian dalam 1 hari, diambil rata – rata. Pada gambar 8 yang menunjukkan *developer tools* Google Chrome, nilai ukuran website didapatkan pada indikator *resources*. Sedangkan nilai kecepatan aksesnya didapatkan pada indikator *Finish*. Pada gambar 9 yang menunjukkan *developer* pada Mozilla Firefox, nilai ukuran website diambil dari indikator *transferred* dan nilai kecepatan website diambil dari indikator *Finish*. Dalam pengujian yang peneliti laksanakan

tidak menghitung ukuran *disk cache*, karena *cache* hanya akan muncul jika dilakukan *load* aplikasi website lebih dari satu kali. Untuk memastikan bahwa pengujian dilakukan tanpa *disk cache*, peneliti memanfaatkan fitur *incognito* pada aplikasi peramban browser untuk tiap pengujian, agar tidak terjadi *caching* pada halaman website.

HASIL DAN PEMBAHASAN

Hasil yang diperoleh dari hasil pengujian menggunakan *developer tools* pada aplikasi peramban internet Google Chrome dan Mozilla Firefox, berupa : waktu dari berapa lama website dimuat oleh peramban internet, yang dinyatakan dalam satuan detik / *second*, dan berapa ukuran website ketika sudah dimuat oleh peramban internet, yang dinyatakan dalam ukuran *MegaByte*. Dipresentasikan dalam tabel 10 sebagai hasil dari eksperimen yang telah peneliti laksanakan. Penelitian dilaksanakan menggunakan Macbook Air M1 (2020) sebagai perangkat untuk melakukan eksperimen. Dengan spesifikasi RAM 8GB, processor Apple Silicon M1, dan penyimpanan berjenis SSD dengan ukuran 256GB. Dalam melakukan eksperimen, akses internet yang digunakan menggunakan jaringan LAN yang sama untuk tiap eksperimen dan memiliki bandwidth yang tetap. Dengan layanan provider internet biznet yang memiliki *bandwith* 100 Mbps.

Tabel 1. Hasil Eksperimen

Hari ke-	Ukuran (<i>MegaByte</i>)						Waktu (<i>Second</i>)					
	Mozilla			Chrome			Mozilla			Chrome		
	SSR	CSR	ISR	SSR	CSR	ISR	SSR	CSR	ISR	SSR	CSR	ISR
1	6.5	3.1	2.4	7.4	1.2	7.4	3.0	4.0	3.0	3.0	3.2	3.0
2	4.2	2.9	2.2	4.8	3.7	4.3	3.4	3.4	3.0	3.1	3.2	2.8
3	7.1	3.1	2.4	2.4	1.6	1.2	2.9	3.0	3.8	2.8	3.0	2.9
4	3.5	4.1	1.2	6.4	4.7	1.4	2.8	3.2	3.0	2.9	3.1	2.9
5	6.5	3.1	2.4	7.4	1.2	7.4	3.0	4.0	3.0	3.0	3.2	3.0
6	5.1	1.8	1.9	5.4	3.6	1.5	3.0	3.7	2.9	3.0	3.2	3.1
7	4.5	2.3	1.9	5.9	3.0	1.5	2.9	3.1	2.9	2.9	3.1	3.0
Rata - Rata	5.3	2.9	1.9	5.1	2.9	2.8	3.0	3.4	3.2	3.0	3.1	2.9

Sumber : Penelitian (2023)

Berdasarkan hasil pada tabel 10, dapat dilihat bahwa nilai performansi dihitung rata-rata dari pengamatan selama 7 hari. Performansi yang diukur adalah performansi ukuran laman website yang dihasilkan dan kecepatan waktu menghasilkan laman website tersebut. Dalam menggenerate laman website, metode ISR menunjukkan nilai terendah adalah 1.9 Megabyte dan nilai tertinggi adalah 5.3 Megabyte pada aplikasi peramban mozilla. Sedangkan untuk perangkat chrome menunjukkan nilai terendah adalah 2.8 Megabyte dan nilai tertinggi adalah 5.1 Megabyte. Hal ini dapat diartikan bahwa perangkat mozilla dengan metode ISR mampu memberikan ukuran yang paling ringan dalam melakukan generation lama website.

Jika dibandingkan antar metode, metode SSR pada mozilla menunjukkan nilai 5.3 Megabyte dan SSR pada perangkat chrome menunjukkan 5.1 Megabyte. Laman web yang *digenerate* menggunakan metode CSR pada mozilla dan chrome menunjukkan ukuran 2.9 megabyte. Hal ini aman web yang digenerate dengan metode ISR pada mozilla menunjukkan nilai 1.9 megabyte dan perangkat chrome menunjukkan 2.8 Megabyte. Perbedaan ini disebabkan karena: 1). Perangkat yang digunakan, tentunya untuk metode CSR sangat bergantung pada spesifikasi kemampuan perangkat, karena render website nya dilakukan oleh pengguna, 2). Kecepatan akses dan *bandwith* internet.

Dari hasil eksperimen, metode CSR sangat bergantung pada aplikasi peramban browser yang digunakan oleh pengguna. Spesifikasi perangkat juga memengaruhi cara website ditampilkan. Jadi, setiap pengguna bisa melihat website dengan tampilan yang berbeda, tergantung pada perangkat yang mereka pakai. Dalam eksperimen ini, peneliti menggunakan Macbook Air M1 (2020) sebagai perangkat untuk melakukan eksperimen. Dengan spesifikasi RAM 8GB, processor Apple Silicon M1, dan penyimpanan bertipe SSD dengan ukuran 256GB. Tentunya akses dan *bandwith* internet juga berpengaruh terhadap kecepatan akses website.

Pada implementasinya, kinerja server memiliki dampak besar pada kecepatan akses website, terutama dalam metode ISR dan SSR di mana rendering website terjadi di sisi server. Faktor lain yang memengaruhi adalah lokasi server tempat website diakses, semakin dekat dengan pengguna akan semakin cepat aksesnya. Oleh karena itu, efisiensi server dan lokasi server juga menjadi kunci dalam pengalaman akses website yang cepat.

Rekomendasi untuk pembuatan website, jika menginginkan kecepatan berarti dapat menggunakan metode ISR untuk peramban chrome jika menggunakan mozilla SSR. Jika menginginkan generate yang ringan maka menggunakan metode ISR untuk mozilla dan ISR untuk chrome.

KESIMPULAN

Berdasarkan eksperimen dan penelitian yang sudah peneliti laksanakan. Didapatkan hasil eksperimen dan diambil dari rata – rata dari 7 hari eksperimen. Dapat diketahui bahwa metode ISR (*Incremental Static Regeneration*) membuat aplikasi website memiliki ukuran yang ringan pada pengujian yang dilakukan baik di aplikasi peramban *web* Google Chrome maupun aplikasi peramban web Mozilla Firefox. Untuk kecepatan akses yang didapat pada aplikasi peramban Google Chrome, metode ISR (*Incremental Static Regeneration*) memiliki kecepatan akses lebih cepat dibandingkan dengan metode rendering lainnya, sedangkan pada aplikasi peramban Mozilla Firefox, metode SSR (*Server Side Rendering*) mengungguli metode ISR (*Incremental Static Regeneration*) sebesar 10% dalam kecepatan aksesnya. Oleh karena itu, dapat disimpulkan bahwa metode ISR (*Incremental Static Regeneration*) memiliki ukuran yang lebih ringan dan kecepatan akses yang lebih cepat dari hasil proses *rendering*-nya, sehingga cocok digunakan dalam pengembangan aplikasi *website* dalam sisi *frontend*-nya. Dalam pengembangan aplikasi website, terutama pada *front-end* banyak metode *rendering* yang dapat digunakan. Metode *rendering website* sangat berpengaruh pada aspek kecepatan akses dan ukuran suatu *website*. Untuk penelitian lanjutan dari penelitian ini, dapat digunakan metode *rendering website* lainnya yang tersedia. Tentunya setiap metode *rendering website* memiliki kelebihan dan kekurangan sesuai dengan peruntukannya. Dan dengan seiring perkembangan teknologi kedepannya, tentunya akan lebih *modern* dalam proses pengembangan suatu website.

REFERENSI

- Abraham, I., & Supriyati, Y. (2022). DESAIN KUASI EKSPERIMEN DALAM PENDIDIKAN: LITERATUR REVIEW. *Jurnal Ilmiah Mandala Education (JIME)*, 8(3), 2442–9511. <https://doi.org/10.36312/jime.v8i3.3800/http>
- Apriyanti, W., Syahlanisyiam, M., Anggraini, Y., Gunawan, S., Tyas Arinanto, R., Adnan Fauzan, R., Tuto Suban, N., & Agung, A. L. (2022). SOSIALISASI PENGGUNAAN INTERNET YANG SEHAT BAGI ANAK-ANAK DI YAYASAN DOMYADHU. *Abdi Jurnal Publikasi*, 1(1). <https://jurnal.portalpublikasi.id/index.php/AJP/index>
- Baehaqi, A., Subhi Bashit, M., Eko Indrajit, R., & Dwi Kurniawan, R. (2023). FRONT END LEARNING MANAGEMENT SYSTEM DEVELOPMENT USING THE NEXTJS FRAMEWORK. *Jurnal Teknik Informatika*, 4(4), 899–911. <https://doi.org/10.52436/1.jutif.2023.4.4.1273>
- Biyaypu, V. P., Jammalamadaka, S. K. R., Jammalamadaka, S. B., Chokara, B., Duvvuri, B. K. K., & Budaraju, R. R. (2023). Building an Expert System through Machine Learning for Predicting the Quality of a Website Based on Its Completion. *Computers*, 12(9). <https://doi.org/10.3390/computers12090181>
- Budikusuma, I., & Susanto, S. (2022). PENGEMBANGAN APLIKASI TOEFL PRACTICE EXAM BERBASIS WEBSITE PADA UNIVERSITAS TEKNOLOGI SUMBAWA. *Jurnal Mahasiswa Teknik Informatika*, 6(1). <https://kbbi.kemdikbud.go.id/entri/pendidikan>
- Fadhilah Iskandar, T., Lubis, M., Fabrianti Kusumasari, T., & Ridho Lubis, A. (2020). Comparison between client-side and server-side rendering in the web development. *IOP Conference Series: Materials Science and Engineering*, 801(1). <https://doi.org/10.1088/1757-899X/801/1/012136>
- Fariz, M., Lazuardy, S., & Anggraini, D. (2022). Modern Front End Web Architectures with React.Js and Next.Js. *International Research Journal of Advanced Engineering and Science*, 7(1), 132–141.
- Furqani, N. El, & Muliono, R. (2021). Web-Based Library Information System Design at SDN 056004 Basiram. *Journal of Research Computer Science*, 1(1), 2770–1800. <http://journal.station-it.org/index.php/jrcs>
- Hartawan, M. S., & Id, J. (2022). PENERAPAN USER CENTERED DESIGN (UCD) PADA WIREFRAME DESAIN USER INTERFACE DAN USER EXPERIENCE APLIKASI SINOPSIS FILM. *Jurnal Elektro & Informatika Swadharma (JEIS)*, 02(01).
- Jartarghar, H. A., Rao Salanke, G., Kumar, A. A., & Dalali, S. (2022). React Apps with Server-Side Rendering: Next.js. *Journal of Telecommunication*, 14(04).

- Jubhari Phie Joarno, R., Fajar, M., & Yunus, A. (2022). IMPLEMENTASI PROGRESSIVE WEB APPS PADA WEBSITE GETHELP MENGGUNAKAN NEXT.JS. *Jurnal KHARISMA Tech*, 17, 1–15. <https://jurnal.kharisma.ac.id/kharismatech/>
- Mashhadi Mohammad, H., & Valentina, G. (2023). *Visual analytics for job candidates*. Politecnico Di Torino.
- Muhammad, F. S., & Muhammad, N. F. (2019). Review Tools Web Browser Forensics untuk Mendukung Pencarian Bukti Digital. *JEPIN (Jurnal Edukasi Dan Penelitian Informatika)*, 05(01).
- Pangestika, R., & Dirgahayu, R. T. (2020). Pengembangan Back-end Sistem Informasi Pendataan Sekolah Desa Komunitas Pendar Foundation Yogyakarta. *Jurnal UII Automata*, 1(2).
- Ronaldo, M., & Pasha, D. (2021). SISTEM INFORMASI PENGELOLAAN DATA SANTRI PONDOK PESANTREN AN-AHL BERBASIS WEBSITE. *Telefortech*, 2(1), 2774–5384.
- Tertiaavini, I Made Agus Oka Gunawan, Kraugusteeliana, Winarno, E., & Rony Sandra Yofa Zebua. (2023). Perancangan dan Implementasi Frontend Web untuk Sistem Pengaduan Masyarakat. *Jurnal Informasi Dan Teknologi*, 112–126. <https://doi.org/10.37034/jidt.v5i1.290>
- Wahdaniya, N., & Sariwati, R. (2022). Legal Aspects of Using Digital Signatures in Online Business Transactions. In *International Journal of Latest Research in Humanities and Social Science (IJLRHSS)*. www.ijlrhss.com
- Yunanto, A. A., Wildan, M. A., Khoirunnisa, B., Lestari, R. D., Yogi, A., & Nugraha, D. (2022). PENGEMBANGAN SISTEM MONITORING KINERJA GURU DI SMK MASKUMAMBANG 1 SURABAYA. *Print) Jurnal POROS TEKNIK*, 14(1), 10–17.