

**SKRIPSI**

**ANALISIS PERBANDINGAN KINERJA *FRAMEWORK* REACT JS DAN  
SVELTE UNTUK IMPLEMENTASI PWA**

**Disusun dan diajukan oleh:**

**ANNAS CASMAWAN AHMAD**

**D121171313**



**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**GOWA**

**2024**

# LEMBAR PENGESAHAN SKRIPSI

## ANALISIS PERBANDINGAN KINERJA *FRAMEWORK* REACT JS DAN SVELTE UNTUK IMPLEMENTASI PWA

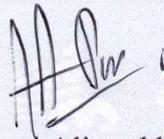
Disusun dan diajukan oleh

**Annas Casmawan Ahmad**  
**D121171313**

Telah dipertahankan dihadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika  
Fakultas Teknik Universitas Hasanuddin  
Pada tanggal 24 Juni 2024  
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama,



A. Ais Prayogi Alimuddin, S.T., M.Eng.  
NIP 198305102014041001

Ketua Program Studi,



Prof. Dr. Indrabayti, S.T., M.T., M.Bus.Sys., IPM, ASEAN. Eng  
NIP 197507162002121004

## PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini ;  
Nama : Annas Casmawan Ahmad  
NIM : D121171313  
Program Studi : Teknik Informatika  
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Analisis Perbandingan Kinerja *Framework* React JS dan Svelte untuk  
Implementasi PWA

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilalihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 05 Agustus 2024

Yang Menyatakan



Annas Casmawan Ahmad

## ABSTRAK

**ANNAS CASMAWAN AHMAD.** *Analisis Perbandingan Kinerja Framework React JS dan Svelte untuk Implementasi PWA* (dibimbing oleh A. Ais Prayogi Alimuddin, S.T., M.Eng)

Perkembangan pesat dalam teknologi web telah mengubah peran situs web dari sekadar penyedia informasi menjadi platform multifungsi yang mencakup komunikasi real-time, streaming media, hingga pengeditan foto. Fokus kualitas dan performa situs web menjadi krusial dalam membangun persepsi positif pelanggan dan memperkuat loyalitas. Dalam konteks ini, penelitian ini membandingkan dua framework frontend React.JS dan Svelte. Fokus utama adalah pada analisis kinerja keduanya yang mengukur metrik seperti *First Contentful Paint (FCP)*, *Speed Index*, *Largest Contentful Paint (LCP)*, *Cumulative Layout Shift (CLS)*, *Time to First Byte (TTFB)*, *Start Render*, dan *Page Weight*. Hasil analisis menunjukkan bahwa Svelte unggul dalam sebagian besar metrik performa, Sedangkan React.JS menonjol dalam metrik *Time to First Byte (TTFB)* Svelte juga menawarkan pendekatan pengembangan yang lebih efisien dengan jumlah baris kode dan ukuran file yang lebih kecil. Kesimpulannya, Svelte menawarkan solusi yang lebih ringkas dan efisien dalam pengembangan aplikasi web, dengan potensi meningkatkan kinerja dan pengalaman pengguna.

Kata kunci: teknologi web, front-end web, React.JS, Svelte, *Virtual DOM*, kompiler, performa web.

## ABSTRACT

**ANNAS CASMAWAN AHMAD.** *Comparison Analysis of Framework React JS and Svelte Performance for PWA Implementation (supervised by A. Ais Prayogi Alimuddin, S.T., M.Eng.)*

The rapid development in web technology has transformed the role of websites from mere information providers to multifunctional platforms encompassing real-time communication, media streaming, and even photo editing. The focus on the quality and *performance* of websites is crucial in building positive customer perceptions and reinforcing loyalty. In this context, this research compares two frameworks frontend React JS and Svelte. The primary focus is on analyzing their performance, measuring metrics such as *First Contentful Paint (FCP)*, *Speed Index*, *Largest Contentful Paint (LCP)*, *Cumulative Layout Shift (CLS)*, *Time to First Byte (TTFB)*, *Start Render*, and *Page Weight*. The results of the analysis indicate that Svelte excels in most *performance* metrics, while React.JS stands out in the *Time to First Byte (TTFB)* metric. Svelte also offers a more efficient development approach with fewer lines of code and smaller file sizes. In conclusion, Svelte provides a more concise and efficient solution in web application development, with the potential to enhance *performance* and user experience.

Keywords: web technology, web front-end, React.JS, Svelte, *Virtual DOM*, compiler, web performance.

## DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI.....	i
PERNYATAAN KEASLIAN.....	ii
ABSTRAK.....	iii
ABSTRACT.....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	vii
DAFTAR TABEL.....	viii
KATA PENGANTAR.....	ix
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	4
1.3. Tujuan.....	4
1.4. Manfaat.....	5
1.5. Ruang Lingkup.....	6
BAB II TINJAUAN PUSTAKA.....	7
2.1. Pengertian <i>Website</i> .....	7
2.2. Bahasa Pemrograman.....	8
2.2.1. Javascript.....	8
2.2.2. Node JS.....	9
2.2.3. React JS.....	9
2.2.2. Svelte.....	11
2.3 Progressive Web App.....	11
2.4. <i>Performance Testing</i> .....	12
2.5. WebPageTest.....	12
2.6. Line dan Size of Code.....	12
BAB III METODE PENELITIAN/PERANCANGAN.....	14
3.1 Lokasi Penelitian.....	14
3.2 Instrumen Penelitian.....	14
3.3 Tahapan Penelitian.....	15
3.4 Metode Pembuatan <i>Website</i> .....	18
3.4.1 Analisis Kebutuhan Sistem.....	19
3.4.2 <i>Use Case Diagram</i> .....	21
3.4.3 Activity Diagram.....	22

3.4.4 Implementasi Sistem .....	28
3.4.5 Skenario Pengujian.....	29
3.4.6 Perhitungan Line dan Size of Code .....	31
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>33</b>
4.1 Hasil.....	33
4.2 Pembahasan .....	58
<b>BAB V KESIMPULAN DAN SARAN.....</b>	<b>61</b>
5.1 Kesimpulan.....	61
5.2 Saran .....	63
<b>DAFTAR PUSTAKA .....</b>	<b>64</b>

## DAFTAR GAMBAR

Gambar 1 Perbedaan cara kerja React dan Svelte.....	3
Gambar 2 Lokasi Penelitian .....	14
Gambar 3 Diagram tahapan penelitian.....	15
Gambar 4 Diagram model waterfall.....	18
Gambar 5 <i>Use Case Diagram</i> Rancangan Sistem.....	22
Gambar 6 <i>Activity Diagram</i> Admin .....	23
Gambar 7 <i>Activity Diagram</i> Mahasiswa .....	25
Gambar 8 <i>Activity Diagram</i> Dosen .....	27
Gambar 9 Hasil pengukuran WebPageTest .....	31
Gambar 10 Extension VScode counter .....	31
Gambar 11 Command VSCode Counter.....	32
Gambar 12 Halaman Login.....	33
Gambar 13 Halaman <i>List User</i> .....	34
Gambar 14 Modal Tambah/Edit User .....	35
Gambar 15 Halaman Profile.....	30
Gambar 16 Halaman Kanban Board Mahasiswa .....	31
Gambar 17 Modal Detail Item Tugas.....	32
Gambar 18 Halaman <i>List Mahasiswa</i> .....	38
Gambar 19 Waktu First Contentful Paint pada setiap test case .....	40
Gambar 20 Waktu Largest Contentful Paint setiap test case .....	43
Gambar 21 <i>Cumulative Layout Shift (CLS)</i> setiap test case.....	45
Gambar 22 Waktu <i>Speed Index</i> setiap test case .....	41
Gambar 23 Ukuran <i>Page Weight</i> setiap test case.....	43
Gambar 24 Waktu <i>Start Render</i> setiap test case .....	51
Gambar 25 Waktu Time to First Byte setiap test case .....	53
Gambar 26 Perbandingan Performa React JS dan Svelte dalam Metrik dengan satuan detik pada React JS dan Svelte .....	54
Gambar 27 Perbandingan Performa React JS dan Svelte di Metrik <i>Page Weight</i> .....	55
Gambar 28 Perbandingan Performa React JS dan Svelte dalam Metrik CLS .....	56

## DAFTAR TABEL

Tabel 1 <i>Test case</i> untuk pengujian performa .....	29
Tabel 2 Pengujian <i>First Contentful Paint</i> pada React JS dan Svelte .....	39
Tabel 3 Pengujian <i>Largest Contentful Paint</i> pada React JS dan Svelte.....	42
Tabel 4 Pengujian <i>Cumulative Layout Shift</i> pada React JS dan Svelte.....	44
Tabel 5 Pengujian <i>Speed Index</i> pada React JS dan Svelte .....	46
Tabel 6 Pengujian <i>Page Weight</i> pada React JS dan Svelte .....	48
Tabel 7 Pengujian <i>Start Render</i> pada React JS dan Svelte .....	50
Tabel 8 Pengujian <i>Time to First Byte (TTFB)</i> pada React JS dan Svelte .....	52
Tabel 9 Hasil hasil pengujian dengan satuan detik pada React JS dan Svelte .....	54
Tabel 10 Hasil Akhir <i>Page Weight</i> pada React JS dan Svelte .....	55
Tabel 11 Hasil Akhir <i>Cumulative Layout Shift</i> pada React JS dan Svelte.....	55
Tabel 12 Hasil perhitungan baris kode pada React JS dan Svelte .....	57
Tabel 13 Hasil perhitungan ukuran kode pada React JS dan Svelte .....	57
Tabel 14 Hasil perbandingan dengan React JS dan Svelte di sistem.....	60

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT yang senantiasa memberikan rahmat serta hidayahnya serta diberi kelancaran dan kesehatan sehingga penulis dapat menyelesaikan karya tulisnya yang berjudul “**Analisis Perbandingan Kinerja Framework React JS dan Svelte untuk Implementasi PWA**” guna memenuhi salah satu persyaratan dalam menyelesaikan jenjang Strata-1 di Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari sepenuhnya bahwa skripsi ini masih jauh dari kesempurnaan karena menyadari segala keterbatasan yang ada. Dalam penulisan skripsi ini penulis menghadapi berbagai kendala dan masalah, namun karena usaha yang maksimal dan kemampuan yang Tuhan berikan kepada penulis serta bantuan dan dukungan dari berbagai pihak, maka penulisan skripsi ini dapat selesai. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada:

1. Allah SWT ,Tuhan pencipta alam semesta yang senantiasa memberikan rahmat serta hidayahnya kepada penulis.
2. Kedua orang tua penulis, Bapak Ahmad Nganro dan Ibu Rukmini yang selalu memberikan kasih sayang, nasehat, motivasi, dukungan, dan doa kepada penulis.
3. Bapak A. Ais Prayogi Alimuddin, S.T., M.Eng., selaku pembimbing utama yang senantiasa menyediakan waktu, tenaga, pikiran, dan perhatian yang luar biasa dalam mengarahkan penulis dalam penyusunan tugas akhir ini.
4. Bapak Robert, Bapak Zainuddin, Ibu Arizah dan Ibu Yuanita serta segenap staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu kelancaran penyelesaian tugas akhir penulis.
5. Segenap Dosen dan Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah banyak membantu semasa perkuliahan hingga penyelesaian tugas akhir penulis.

6. Saudara seperjuangan penulis RECOGN17ER yang telah menemani dan mendukung perjalanan penulis sekaligus tempat berbagi keluh kesah selama menjadi mahasiswa teknik di Departemen Informatika Fakultas Teknik Universitas Hasanuddin.
7. Seluruh pihak yang tidak sempat disebutkan satu persatu yang telah banyak meluangkan tenaga, waktu, dan pikiran selama penyusunan tugas akhir ini.

# BAB I PENDAHULUAN

## 1.1. Latar Belakang

Perkembangan teknologi yang sangat cepat memberikan sejumlah manfaat signifikan di berbagai sektor masyarakat. Di sektor industri, kemajuan teknologi berperan penting dalam meningkatkan kapasitas produksi perusahaan. Sekitar 73,7% dari jumlah total populasi Indonesia adalah pengguna internet, data ini didasarkan pada survei yang dilakukan oleh APJII (Asosiasi Penyelenggara Jasa Internet Indonesia) pada periode 2019-2020 (Gunawan et al., 2021).

*Website* sudah berkembang sangat pesat. Sekarang, *website* digunakan tidak hanya untuk media berbagi informasi atau berita. Layaknya sebuah aplikasi pada komputer dan handphone kita, *website* dapat digunakan untuk komunikasi secara real-time, mendengarkan sebuah lagu, hingga photo editing kini juga dapat dilakukan pada sebuah *website*. 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*. Semakin tinggi kualitas *website*, semakin baik persepsi pelanggan terhadap perusahaan sehingga meningkatkan rasa kepuasan pelanggan, kepuasan pelanggan akan membangun loyalitas pelanggan yang menjadi modal utama untuk menang dan bertahan dalam persaingan bisnis (Putra & Suyanto, 2023).

*Progressive Web Apps (PWA)* telah menjadi terobosan besar dalam dunia teknologi mobile dan web. Aplikasi Web Progresif adalah konsep yang menggabungkan teknologi terkait untuk menghasilkan aplikasi dengan pengalaman seperti aplikasi native. PWA adalah aplikasi yang semakin banyak digunakan, akan semakin kuat, yaitu sejak pengguna pertama kali

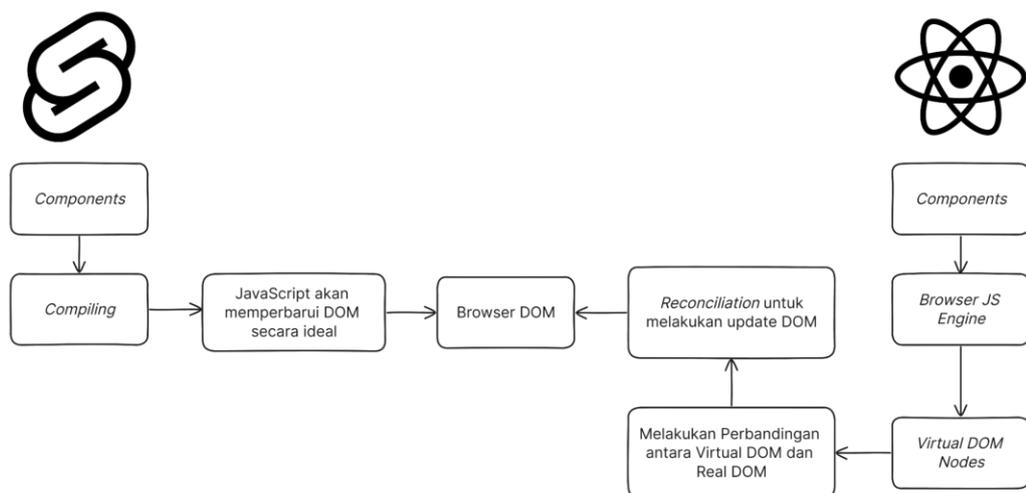
membuka halaman web konsep PWA, sampai pengguna menutup halaman web, aplikasi memiliki kecepatan yang baik, meskipun kecepatan internet tidak baik. Selain itu, Aplikasi Web Progresif memiliki fitur notifikasi push, dapat menyimpan pintasan di layar beranda, dan berjalan di layar penuh. (Syah Lampah & Setiawan, 2019)

Namun, mengintegrasikan PWA bukan satu-satunya hal yang perlu diperhatikan oleh pengembang. Kualitas dan performa sebuah website tetap menjadi kunci sukses. Sebuah website yang berkinerja baik tidak hanya meningkatkan keterlibatan pengguna, tetapi juga meningkatkan visibilitas di mesin pencari. Faktor-faktor seperti kecepatan loading, kesalahan teknis, dan kualitas interaksi pengguna semuanya berkontribusi pada kesuksesan suatu website. Karena itu, pengoptimalan terus-menerus dari website adalah esensial.

Dalam rangka pengoptimalan tersebut, peran front-end web menjadi krusial. Front-end adalah jendela pertama yang dilihat oleh pengguna saat mengakses situs. Sebuah front-end yang responsif, menarik, dan mudah digunakan dapat meningkatkan retensi pengguna dan mendorong keterlibatan yang lebih dalam. Untuk mencapai ini, pemilihan library atau framework yang tepat adalah kunci. Library dan framework JavaScript, telah membantu pengembang dalam menciptakan pengalaman web yang dinamis dan interaktif. Dengan menggunakan teknologi ini, pengembang dapat memastikan bahwa situs mereka tidak hanya berfungsi dengan baik tetapi juga menawarkan pengalaman yang memuaskan bagi pengguna.

Umumnya ada dua cara yang dilakukan framework frontend dalam melakukan rendering dan memanipulasi DOM, yaitu dengan menggunakan *Virtual DOM (VDOM)* dan melakukan kompilasi secara langsung ketika proses *build time*. Framework seperti React.JS menggunakan pendekatan VDOM, sementara framework seperti Svelte memanfaatkan pendekatan

kompiler. Framework lain yang menggunakan VDOM termasuk Vue.JS dan Preact, sedangkan Solid.JS menggunakan pendekatan kompiler. Meskipun ada berbagai framework yang menerapkan kedua pendekatan ini, fokus penelitian ini akan terbatas pada perbandingan antara React.JS dan Svelte. Kedua framework ini menggunakan pendekatan yang dikenal sebagai *component-based architecture*, yang memungkinkan pengembang untuk membangun aplikasi dengan cara memecahnya menjadi komponen-komponen yang lebih kecil dan dapat digunakan kembali.



Gambar 1 Perbedaan cara kerja React dan Svelte

Pendekatan VDOM yang digunakan dalam React.JS mengandalkan struktur *Virtual DOM*, yang menciptakan representasi *virtual* dari tampilan komponen. Ini memungkinkan React untuk memperbarui hanya bagian-bagian yang diperlukan pada tampilan tanpa merender ulang seluruh halaman, menghasilkan penghematan sumber daya dan peningkatan performa. Di sisi lain, Svelte adalah pendekatan yang lebih dekat dengan kompiler, yang memproses kode komponen Svelte menjadi JavaScript murni pada tahap kompilasi. Hal ini menghilangkan overhead VDOM dan

menghasilkan kode yang lebih efisien saat dijalankan di browser. Dengan melakukan perbandingan ini, penelitian ini diharapkan dapat memberikan panduan yang lebih jelas untuk pengembang dalam memilih pendekatan yang sesuai dengan kebutuhan proyek, dengan penekanan pada efisiensi dan performa situs web.

Dalam penelitian ini, akan diuji dan dibandingkan kinerja pengembangan web menggunakan framework frontend React.JS dan Svelte, termasuk implementasinya dalam konteks *Progressive Web Apps (PWA)*. Analisis ini tidak hanya akan berfokus pada aspek-aspek kinerja tetapi juga pada kemampuan dan efektivitas kedua *framework* dalam menyediakan fitur dan fungsi PWA yang efisien dan handal. Dengan membedah kelebihan dan kekurangan dari masing-masing, penelitian ini diharapkan dapat memberikan wawasan yang membantu pengembang dalam membuat keputusan yang tepat saat merancang dan mengembangkan situs web dan aplikasi web progresif yang mengutamakan efisiensi, kinerja, serta pengalaman pengguna yang setara dengan aplikasi native.

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang telah dijabarkan sebelumnya, berikut merupakan masalah-masalah yang ingin diatasi dalam penelitian ini:

1. Bagaimana cara menerapkan penggunaan framework React JS dan Svelte dalam pengembangan web?
2. Bagaimana melakukan perbandingan performa antara framework React JS dan Svelte dalam pengembangan web?
3. Bagaimana implementasi PWA pada aplikasi web yang dibangun menggunakan React JS dan Svelte?

## **1.3. Tujuan**

Berdasarkan rumusan masalah yang telah dijelaskan sebelumnya, maka tujuan dari penelitian ini yaitu:

1. Untuk menerapkan penggunaan framework React JS dan Svelte dalam pengembangan web.
2. Untuk mengukur dan membandingkan kinerja (performa) antara framework React JS dan Svelte dalam pengembangan web.
3. Untuk mengimplementasikan *Progressive Web App* (PWA) dalam aplikasi web yang dibangun menggunakan framework React JS dan Svelte.

#### **1.4. Manfaat**

Berdasarkan latar belakang yang telah dijabarkan sebelumnya, berikut merupakan manfaat dalam penelitian ini yaitu:

1. Memberikan wawasan yang lebih mendalam tentang penerapan dan perbandingan antara penggunaan framework React.JS dan Svelte dalam pengembangan web.
2. Memberikan panduan yang berguna kepada pengembang dalam memilih framework yang sesuai dengan proyek pengembangan web mereka, dengan fokus pada efisiensi dan performa situs web.
3. Meningkatkan pemahaman tentang manfaat dan tantangan implementasi PWA dalam konteks pengembangan web modern, serta memperjelas bagaimana teknologi ini dapat diintegrasikan dengan React.JS dan Svelte untuk meningkatkan pengalaman pengguna.

## 1.5. Ruang Lingkup

Ruang lingkup dari pengerjaan tugas akhir ini adalah :

1. Pembuatan website menggunakan Java Script dan library React dan Svelte.
2. Web browser yang dipilih yaitu Google Chrome, Edge dan Mozilla Firefox
3. Parameter yang dijadikan sebagai acuan validasi untuk mengukur kecepatan sisi klien menggunakan webpagetest.com.
4. Evaluasi *Developer Experience* berdasarkan *Lines of Code (LOC)*, dan *Size of Code (SOC)* dari kode yang dihasilkan saat menggunakan library React dan Svelte.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Pengertian *Website***

*Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Hubungan antara satu halaman web dengan halaman web yang lainnya disebut hyperlink, sedangkan teks yang dijadikan media penghubung disebut hypertext (Batubara 2015).

Dalam beberapa dekade, website telah menjadi lebih besar dan kompleks. Pada awalnya website digunakan hanya untuk mempermudah tukar menukar dan melakukan perbaruan informasi kepada sesama peneliti yang dilakukan oleh Sir Timothy John. namun dalam perkembangannya website dapat melakukan manajemen konten seperti video dan gambar (Butkiewicz 2011).

Secara teknis, web adalah sebuah sistem dimana informasi dalam bentuk teks, gambar, suara, dan lain-lain yang tersimpan dalam sebuah internet web server dipresentasikan dalam bentuk hypertext. Informasi di web dalam bentuk teks umumnya ditulis dalam format HTML. Informasi lainnya disajikan dalam bentuk grafis (dalam format GIF, JPG, PNG), suara (dalam format AU, WAV), dan objek multimedia lainnya (seperti MIDI, Shockwave, Quicktime Movie, 3D World). Web dapat diakses oleh perangkat lunak web client yang secara populer disebut sebagai browser. Browser membaca halaman-halaman yang tersimpan dalam webserver melalui protokol yang disebut HTTP (Hypertext Transfer Protocol). Sebagai dokumen hypertext, dokumen-dokumen di web dapat memiliki link dengan dokumen lain, baik yang tersimpan dalam web server yang sama maupun di web server lainnya. Link memudahkan para pengakses web berpindah dari satu halaman ke halaman lainnya, dan "berkelana" dari satu server ke server lain. Kegiatan

penelusuran halaman web ini biasa diistilahkan sebagai browsing, ada juga yang menyebutnya sebagai surfing (berselancar). Seiring dengan semakin berkembangnya jaringan internet di seluruh dunia, maka jumlah situs web yang tersedia juga semakin meningkat. Hingga saat ini, jumlah halaman web yang bisa diakses melalui internet telah mencapai angka miliaran. Untuk memudahkan penelusuran halaman web, terutama untuk menemukan halaman yang memuat topik-topik yang spesifik, maka para pengakses web dapat menggunakan suatu mesin pencari (search engine). Penelusuran berdasarkan search engine dilakukan berdasarkan kata kunci (keyword) yang kemudian akan dicocokkan oleh search engine dengan basis data miliknya (Batubara 2015).

## **2.2. Bahasa Pemrograman**

### **2.2.1. Javascript**

Javascript Adalah Bahasa pemrograman yang populer. Javascript adalah Bahasa pemrograman yang digunakan untuk HTML dan web, untuk Server, PC, laptop, tablet, dan lebih banyak lagi. Kode pemrograman Javascript dapat disisipkan kedalam halaman HTML Pada awalnya, Javascript mulai diperkenalkan di browser Netscape Navigator 2. Namun waktu itu Namanya bukan Javascript, namun LiveScript. Mengingat pada 7 waktu itu teknologi Java sedang panas-panasnya atau sedang tren, maka pihak Netscape memutuskan untuk mengganti Namanya menjadi Javascript, yang sepertinya nama tersebut lebih marketable dibandingkan LiveScript. Selanjutnya pihak Microsoft (rival Netscape) pun mulai ikut ikutan memfasilitasi web browser buatannya, 'Internet Explorer', supaya bisa mendukung Javascript. Namun mungkin karena gengsi, pihak Microsoft memberi nama Bahasa yang lain, yaitu Javascript. Mulai saat itu, Netscape dan Microsoft mulai berlomba-lomba mengembangkan bahasa tersebut dalam versi yang berlainan. Oleh sebab persaingan itulah terkadang suatu Javascript mungkin bisa bekerja dengan baik di browser Netscape,

tapi tidak demikian halnya di Internet Explorer, begitu pula sebaliknya (Permana 2016).

Javascript merupakan bahasa pemrograman yang bersifat *Client Side Programming Language*. *Client Side Programming Language* merupakan tipe bahasa pemrograman yang pemrosesannya dilakukan oleh web browser atau disebut dengan client seperti Google Chrome, Opera Mini, Mozilla Firefox dan sebagainya. Pada pertengahan dekade 90'an Javascript pertama kali dikembangkan. Javascript berbeda dengan bahasa pemrograman Java, meskipun memiliki nama yang hampir sama. Javascript dapat disisipkan kedalam dokumen HTML ataupun dipisah menjadi dokumen sendiri yang kemudian dihubungkan dengan dokumen lain yang dituju. (Omar dkk, 2018)

### **2.2.2. Node JS**

Node JS merupakan sebuah sistem perangkat lunak yang didesain untuk pengembangan aplikasi web Node JS, yang biasa juga disebut sebagai runtime environment. Node Js merupakan gabungan Bahasa pemrograman C++ dan juga Javascript yang mempunyai model event driven (basis event) dan asynchronous I/O. Node js dieksekusi sebagai aplikasi server karena dukungan V8 Engine buatan Google dan beberapa modul bawaan. (Fajrin, 2017)

### **2.2.3. React JS**

ReactJS, juga dikenal dengan nama React atau ReactJS, adalah pustaka java script bersifat open source digunakan untuk membangun antarmuka pengguna. ReactJS digunakan untuk mengatasi tampilan pada aplikasi berhalaman tunggal (single page application) dan pengembangan aplikasi mobile (Khuat 2018) .

React dikembangkan oleh facebook dengan berusaha untuk menyediakan kecepatan, kesederhanaan, dan skalabilitas. Beberapa fitur utama react adalah JSX, komponen stateful, dan Virtual Document Object Model.

### **2.2.3.1. *Virtual DOM***

*Virtual DOM* adalah abstraksi yang memungkinkan pengembang React.js untuk mengelola JavaScript sebagai reaktif. Ini adalah replika cepat, di memori dari DOM nyata. *Virtual DOM* memungkinkan untuk merender ulang setiap perubahan state, mencerminkan aktivitas pengguna dan menyediakan pengalaman pengguna yang lancar. (Zerihun Dinku, 2022)

Bekerja pada *Virtual DOM*, di sisi lain, berbeda dari bekerja pada DOM nyata karena layouting tidak diperlukan. React.js membuat patches untuk DOM browser dengan menghitung perbedaan antara subtrees *virtual* dan yang telah dimodifikasi. *Virtual DOM* dapat menggunakan rerendering subtree selektif dengan memanggil `setState()` pada komponen tertentu. Sebuah komponen dari metode `setState` React.js menandainya sebagai tidak murni. (Zerihun Dinku, 2022)

### **2.2.3.2. JSX**

JSX bukanlah sebuah string atau HTML. Ini adalah ekstensi sintaks yang diketik secara statis untuk JavaScript. Ini mirip dengan bahasa berorientasi objek yang dirancang untuk dijalankan pada browser web modern. JSX direkomendasikan untuk digunakan bersama React untuk merancang dan membangun antarmuka pengguna. Meskipun datang dengan seluruh kemampuan JavaScript, pada pandangan pertama, ini bahkan mungkin terlihat seperti bahasa

templat, meskipun sebenarnya tidak. Elemen React dihasilkan oleh JSX. Ini dapat di-render ke React *Virtual DOM* (Islam Naim, 2017)

### **2.2.2. Svelte**

Svelte.js adalah alat untuk membangun aplikasi web yang cepat. Ini mirip dengan framework JavaScript seperti React.js dan Vue.js, yang berbagi tujuan umum dalam memfasilitasi pembuatan antarmuka pengguna yang mulus dan interaktif. Namun, ada satu perbedaan kunci, yaitu Svelte.js mengkompilasi aplikasi menjadi JavaScript optimal pada saat build, bukan melakukan interpretasi kode aplikasi pada saat runtime. Ini berarti tidak ada biaya kinerja terkait dengan abstraksi framework dan tidak ada penalti saat aplikasi pertama kali di-load. (Wenqing, 2021)

#### **2.2.2.1. Compiler**

*Compiler* adalah perangkat lunak yang menerjemahkan kode dalam satu bahasa pemrograman ke bahasa pemrograman lain. Biasanya ini adalah dari bahasa tingkat tinggi (seperti Go atau Java) ke bahasa tingkat rendah (seperti kode mesin atau bytecode). (Volkman, 2020)

*Compiler* Svelte mengkompilasi file .svelte menjadi JavaScript dan CSS. Ini memiliki banyak manfaat, salah satunya adalah fitur baru dapat ditambahkan ke Svelte tanpa membesarkan ukuran bundle aplikasi yang diterapkan. *Compiler* hanya menyertakan kode untuk fitur-fitur Svelte yang benar-benar digunakan.

## **2.3 Progressive Web App**

Progressive Web Apps(PWA) adalah teknologi baru yang dirancang dan dikembangkan oleh Google Untuk mengatasi keterbatasan jelajah seluler dan aplikasi native, dengan memilih ikon di layar beranda perangkat seperti halnya

aplikasi native dan dapat bekerja tanpa konektivitas yang tersedia. PWA menerapkan teknologi service worker, web manifest dan cache dimana membuat konsep PWA tersebut dapat berjalan diberbagai platform seperti website, desktop dan mobile. (Intan dkk, 2023)

#### **2.4. Performance Testing**

Menurut Wisnuadhi et al., (2020), *Performance testing* adalah teknik pengujian non-fungsional yang dilakukan untuk menentukan parameter sistem seperti kecepatan atau stabilitas sistem. Banyak faktor yang mempengaruhi performa sebuah perangkat lunak. Dalam pengujian perangkat lunak, metrics adalah ukuran kuantitatif dari tingkatan suatu sistem, komponen sistem, atau atribut proses. Metrics diperlukan untuk memahami kualitas dan efektivitas pengujian performa.

#### **2.5. WebPageTest**

WebPageTest adalah tools yang digunakan untuk menguji performa website jika diakses oleh banyak orang sekaligus. Hasilnya berupa waktu yang dibagi menjadi beberapa bagian yaitu waktu saat DNS lookup, waktu koneksi TCP, waktu respon HTTP dan lainnya. Selain waktu, bandwidth juga diukur selama proses pengujian website. (Jimmy dkk., 2020)

#### **2.6. Line dan Size of Code**

Menurut Kaur (2016), LOC (Line of Code) merupakan salah satu langkah termudah untuk mengukur ukuran program perangkat lunak dalam proses pengembangan perangkat lunak. Sedangkan pada laman Geeksforgeeks ("*Lines of Code (LOC) in Software Engineering,*" 2021), LOC adalah semua baris teks dalam kode yang bukan merupakan komentar, baris kosong. Karena LOC hanya menghitung volume kode, jadi hanya dapat digunakan untuk membandingkan atau memperkirakan proyek yang menggunakan bahasa yang sama dan menggunakan standar kode yang sama.

LOC merupakan metrik yang paling banyak digunakan dalam estimasi biaya karena sangat mudah untuk dilakukan. Selain daripada itu, berhubungan dengan LOC ini akan didapatkan juga hasil pengukuran berupa ukuran kode atau SOC (Size of Code) dari suatu file program.