

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan aplikasi web dari tahun ke tahun mengalami kemajuan yang baik dalam hal teknologi yang digunakan. Pada dasarnya, sebuah halaman web dibangun menggunakan bahasa HTML dan CSS sebagai dasar dan penghias. Namun, teknologi terus berkembang dan saat ini membangun sebuah website lebih menitikberatkan pada kecepatan dalam menampilkan halaman web dengan cara mengoptimalkan kode program dan menggunakan teknik seperti *Single Page Application* (SPA) yang hanya memuat satu halaman HTML dan secara dinamis memperbarui halaman sesuai dengan interaksi pengguna. Dengan demikian, website akan lebih cepat dalam menampilkan data [1], [2].

Single Page Application (SPA) adalah sebuah arsitektur yang dapat digunakan untuk membangun aplikasi web yang tidak perlu melakukan *refresh* halaman saat digunakan. SPA hampir sama dengan aplikasi natif yang di-*load* dari server ke browser pengguna. SPA bekerja dengan menampilkan sebagian dari antarmuka pengguna berdasarkan tindakan pengguna. Dalam hal ini, tidak perlu memuat halaman baru secara keseluruhan, cukup bagian yang diperlukan saja yang di-*update*. Sebaliknya, aplikasi web biasa akan *refresh* seluruh halaman ketika terjadi tindakan pengguna, yang menyebabkan tampilan berkedip pada halaman. Waktu loading ini bisa saja memakan waktu beberapa detik jika server sedang dalam kondisi berat. Dalam hal ini SPA bisa menampilkan animasi loading dan menyediakan konten lain yang dapat di-browsing saat itu juga [3] Kelebihan SPA dalam membangun aplikasi web antara lain, waktu respons yang cepat, responsif untuk *mobile* dan *desktop*, pemisahan tampilan dengan logika bisnis aplikasi web, transfer data yang ringan dan cepat, dan memiliki kemungkinan untuk dapat digunakan secara offline [4]. SPA sangat efektif digunakan pada web yang memiliki banyak fitur dan interaksi kepada pengguna.

Dalam mengembangkan aplikasi web dengan arsitektur SPA, dibutuhkan sebuah *Framework* yang akan mempermudah dalam pembuatan aplikasi web.

Framework mengatur alur kerja pengembangan aplikasi, mengurangi waktu pengembangan dan kesalahan yang mungkin terjadi [5]. Dalam perkembangannya, banyak sekali *Framework* yang hadir dan menyuguhkan kekuatannya masing-masing.

Sangat penting untuk dapat mengetahui performa aplikasi web yang dibangun oleh tiap *Framework* yang akan digunakan sebagai pertimbangan pemilihan *Framework* dalam pembangunan aplikasi web. Pemilihan *Framework* yang tepat tidak selalu mudah dilakukan tanpa pengujian mandiri.[6] Beberapa faktor yang perlu dipertimbangkan dalam pemilihan *Framework* termasuk kemudahan penggunaan, dukungan komunitas, dan performansi, yang dapat menentukan tingkat kesulitan untuk menemukan solusi untuk masalah yang mungkin ditemui selama proses pengembangan. Disisi lain pemilihan *Framework* juga dapat digunakan sebagai pilihan strategis untuk perusahaan dan juga untuk para mengembang muda dalam melengkapi atau menambah kemampuannya dalam menguasai *Framework* [5].

Berdasarkan penjelasan diatas tentang pentingnya pemilihan *Framework* dalam pengembangan SPA, menjadi sebuah tantangan para pengembang web dalam pemilihan *Framework* yang sesuai kebutuhan. Masalah timbul ketika para pengembang memilih *Framework* yang akan digunakan untuk pengembangan aplikasi web hanya berdasarkan dari *Framework* yang biasa para pengembang web gunakan, namun ini tidak mendasar karena cenderung subyektif. Dengan begitu banyaknya pilihan *Framework* yang ada, para pengembang seringkali kekurangan informasi dalam menentukan *Framework* mana yang memiliki kinerja yang lebih baik [7]. Dengan permasalahan ini, maka peneliti akan melakukan komparasi terhadap *Framework*, penelitian ini dapat memberikan wawasan yang berharga tentang bagaimana kinerja *Framework* dalam mempengaruhi pengalaman pengguna dan ekfektivitas aplikasi web.

Peneliti memilih dua *Framework* berdasarkan dari popularitas diantara para pengembang web yaitu Svelte dan React untuk dilakukan pengujian performansi. Menurut survey yang dilakukan Sacha Greif dan Eric Burel dalam StateofJs yang membahas mengenai pengembang web yang ingin menggunakan Kembali

Framework yang sudah pernah para pengembang gunakan, Svelte mendapatkan presentase 90% dari 39,472 responden dan React mendapat presentase 83% dari 39,472 responden [8].

Pengujian *Framework* akan berfokus pada performa berdasarkan Matriks Performansi. Parameter performa dipilih karena aplikasi web menggunakan *Framework* yang ideal dapat menyediakan layanan dengan kinerja yang baik. Ada 6 variabel yang dapat digunakan untuk mengukur performa aplikasi web. Variabel tersebut adalah, *First Contentful Paint(FCP)*, *Page Weight*, *Largest Contentful Paint(LCP)*, *Speed Index(SI)*, *Total Blocking Time(TBT)*, dan *Cumulative Layout Shift(CLS)*[7]. Pengujian dilakukan dengan serangkaian *test case* yang akan disusun oleh peneliti dan dengan menggunakan Webpagetest sebagai alat memonitor performa. Diharapkan dengan adanya penelitian ini mampu mengetahui performa *Single Page Application* yang dibangun dari Svelte dan React serta mengetahui *Framework* terbaik ketika digunakan dalam *Single Page Application*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disebutkan di atas, maka penelitian ini memiliki rumusan masalah para pengembang kekurangan informasi dalam menentukan *Framework* mana yang memiliki kinerja yang lebih baik.

1.3 Pertanyaan Penelitian

Berdasarkan rumusan masalah tersebut, penulis merumuskan beberapa pertanyaan terkait penelitian bagaimana performa Svelte dan React dalam pengembangan aplikasi web dan bagaimana perbandingan performa berdasarkan variabel *First Contentful Paint(FCP)*, *Largest Contentful Paint(LCP)*, *Cumulative Layout Shift(CLS)*, *Speed Index(SI)*, *Total Blocking Time(TBT)*, dan *Page Weight* pada *Single Page Application*?

1.4 Batasan Masalah

Berdasarkan rumusan dan tujuan penelitian, maka untuk mewujudkan penelitian yang sesuai dengan masalah yang ada diperoleh batasan-batasan masalah penelitian sebagai berikut:

1. *Framework* Svelte dan React hanya akan dibandingkan satu sama lain dan tidak dibandingkan dengan Javascript murni atau Vanilla Javascript
2. Dalam penelitian ini hanya akan menggunakan Node.js versi 18.16.0 *Framework* Svelte versi 4.0.5 dan *Framework* React versi 18.2.0.
3. Pengujian hanya menggunakan Webpagetest.

1.5 Tujuan Penelitian

Berdasarkan rumusan masalah yang tertera maka dapat diketahui tujuan dari penelitian ini adalah untuk mengetahui *Framework* terbaik berdasarkan variabel *First Contentful Paint(FCP)*, *Largest Contentful Paint(LCP)*, *Speed Index(SI)*, *Total Blocking Time(TBT)*, *Cumulative Layout Shift(CLS)*, dan *Page Weight* antara Svelte dan React untuk digunakan pada arsitektur *Single Page Application*.

1.6 Manfaat Penelitian

Berdasarkan rumusan masalah yang tertera maka dapat diketahui dari penelitian ini adalah:

1. Memberikan informasi mengenai performa Svelte dan React pada *Single Page Application*.
2. Mempermudah para pengembang aplikasi web dalam memilih *Framework* untuk pengembangan *Single Page Application*.