

TEKNIK SINGLE PAGE APPLICATION (SPA) LAYOUT WEB DENGAN MENGGUNAKAN REACT JS DAN BOOTSTRAP

Miftah Faroq Santoso

Universitas Bina Sarana Informatika
miftah.mfq@bsi.ac.id

INFO ARTIKEL

Diajukan :

18 Agustus 2021

Diterima :

28 Agustus 2021

Diterbitkan:

01 Desember 2021

Kata Kunci :

javascript, library, reactjs, single page application, render

INTISARI

Teknologi yang sangat cepat berkembang, memiliki imbas yang cukup besar khususnya di era sekarang yang semua serba instan dan tentunya tidak terlepas dari penggunaan internet. Peristiwa ini secara tidak langsung ikut mempengaruhi di bidang pengembangan *website*. Sebuah halaman web saat ini tidak hanya mengedepankan desain tampilan layout saja, melainkan dari sisi penyajian konten kepada pengguna (*user*). Semakin banyak isi, data atau konten yang akan ditampilkan, maka hal ini ikut mempengaruhi kinerja kecepatan dari sebuah browser untuk melakukan render sebuah halaman dalam melakukan penyajian. Seorang web desainer dituntut tidak hanya berfokus pada layout desain melainkan harus dapat memaksimalkan kinerja, kecepatan dari *website* yang dibuat. Sebuah metode yang dikenal dengan nama *Single Page Application (SPA)* dengan menggunakan *library JavaScript* atau dikenal dengan nama ReactJS, dapat dikatakan bisa menanggapi kendala dalam penyajian isi atau konten serta dapat memaksimalkan kinerja dari sebuah *website*. Dengan mengadopsi teknik *Single Page Application (SPA)* pada layout web yang dibuat, maka browser akan lebih cepat dalam melakukan rendering halaman secara menyeluruh dan sempurna. Hasil uji coba memperlihatkan bahwa layout web yang dibuat dengan menggunakan teknik SPA akan sangat cepat pada saat melakukan penyajian konten. Dengan diimplementasikannya teknik SPA ke dalam sebuah layout web, browser tidak akan melakukan *re-load* ulang secara terus menerus pada keseluruhan halaman saat bernavigasi, melainkan hanya sebagian saja sesuai inputan dari pengguna (*user*), yang semua itu akan diolah secara otomatis oleh *library ReactJS*, sehingga akan menghasilkan pengalaman pengguna dalam berinteraksi lebih nyaman dalam menggunakan sebuah halaman web.

I. PENDAHULUAN

Perkembangan desain laman (*layout*) web dari tahun ke tahun semakin mengalami perkembangan baik dari sisi desain layout, serta teknologi yang digunakan. Sebuah halaman web pada dasarnya hanya dibangun dengan menggunakan bahasa *HTML (Hypertext Markup Language)* sebagai pondasi utama dan *CSS (Cascading Stylesheet)* yang berfungsi sebagai penghias sekaligus pengatur gaya tampilan layout supaya dapat terlihat menarik dan elegan, dari sana kemudian dipadukan sehingga akan menghasilkan sebuah halaman web yang utuh yang dapat diakses diberbagai perangkat baik itu *mobile* ataupun *desktop PC* (Santoso, 2019). Namun demikian teknologi selalu berkembang, dalam mendesain sebuah web tidak hanya berfokus dari sisi estetika saja melainkan kecepatan web dalam melakukan penyajian data.

Tren teknologi dalam membangun sebuah halaman web mengalami perkembangan, hal ini bertujuan agar *website* yang dibuat tidak saja memperhatikan dari segi estetika namun dari segi kecepatan, berapa lama waktu yang dibutuhkan sebuah halaman web untuk ditampilkan, hal ini sangat krusial jika ditinjau dari sisi pengguna (*user*). Jika ditinjau dari segi estetika, sebuah halaman *website* yang terlihat begitu menarik, baik dari desain hingga kemudahan pengguna (*user-friendly*), maka hal ini akan ikut berimbas pada saat halaman web tersebut akan ditampilkan, dengan tampilan dan fitur yang menarik otomatis skrip kode CSS yang ditulis juga akan semakin bertambah. Hal ini menuntut para web desainer mencoba memaksimalkan serta mengoptimalkan agar sebuah halaman *website* tidak membutuhkan waktu yang lama, ketika akan ditampilkan dan tentunya dari segi waktu akan sangat efisien,

karena pengguna tidak perlu harus terlalu lama menunggu pada saat sebuah halaman web dibuka. Salah satu teknik untuk mengoptimalkan sebuah laman web adalah dengan cara membuat satu halaman web sebagai tampilan utama dari aplikasinya, dimana dalam proses penyajian data halaman tidak akan diproses ulang secara menyeluruh, melainkan hanya sebagian saja sesuai dengan permintaan. *Single Page Application (SPA)* adalah teknologi yang bekerja di dalam browser yang tidak membutuhkan reload page saat digunakan. Dengan kata lain, pengguna atau *user* tidak akan berpindah halaman dengan melakukan *request* kepada server setiap kali terjadi interaksi pada aplikasi (Wibowo & Wiguna, 2019). *Single-Page Application (SPA)* merupakan aplikasi web yang hanya memuat satu halaman HTML dan secara dinamis memperbaharui halaman tersebut sesuai dengan interaksi pengguna dengan aplikasi tersebut (Nurpandi & Rosid Liki, 2021). Tentu dalam hal ini website akan lebih cepat pada saat diakses dan akan ditampilkan.

Adapun manfaat dan tujuan menggunakan teknik *Single Page Application (SPA)* adalah agar dalam penyajian sebuah data tidak diperlukan waktu yang lama untuk menampilkan data yang sedang diminta (*request*). Dengan kata lain *Single Page Application (SPA)* menampilkan beberapa halaman dalam 1 (satu) buah halaman. Halamannya tetap, namun konten-nya yang berubah. Teknik ini dapat dikatakan lebih modern jika dibandingkan dengan teknik tradisional dengan teknik *Multi Page Application*.

Dalam penelitian ini digunakan metode atau teknik *Single Page Application (SPA)* supaya didapatkan hasil yang optimal serta efisien dari segi kecepatan (*load time*). Adapun penulisan pemrograman yang digunakan menggunakan *software* pendukung NodeJS menjalankan fungsi Library JavaScript ReactJS yang nanti digunakan untuk menerapkan teknik *Single Page Application (SPA)* dan *CSS Framework* dengan Bootstrap sebagai fitur untuk meresponsifkan *layout website*. Sebagai *tools* pendukung untuk penulisan kode menggunakan editor Visual Code Studio dan Mozilla Firefox serta Google Chrome untuk menjalankan serta mensimulasikan tampilan halaman website.

Hasil yang diharapkan agar dalam proses menampilkan sebuah laman web lebih cepat dan efisien, aplikasi web yang dibuat dalam skala besar yang memerlukan perubahan data setiap saat akan tertanggulangi, karena halaman dapat melakukan perubahan tanpa harus mere-load/

refresh setiap kali terjadi permintaan (*request*) perubahan isi konten atau data.

II. BAHAN DAN METODE

1. Alat dan Bahan

Sebagai bahan (*tools*) pendukung yang digunakan dalam perancangan web aplikasi ini terbagi ke dalam dua bagian, yaitu perangkat lunak (*software*) dan perangkat keras (*hardware*). Dapat dilihat pada tabel 1 untuk spesifikasi *software* dan *hardware* yang dibutuhkan didalam penelitian ini.

Tabel 1. Spesifikasi *Software* dan *Hardware*

No	Alat	Keterangan
1	Perangkat Lunak (<i>software</i>) a. Visual Code Studio b. Mozilla Firefox c. Google Chrome, Firefox d. Nodejs	Editor Pengujian dan simulasi Menjalankan ReactJS
2	Perangkat Keras a. 1 Unit PC/ laptop b. Mouse, Monitor, Keyboard (<i>peripheral</i>) c. Ponsel (HP)	Spesifikasi: RAM 8 G, SSD 128GB Ujicoba di Ponsel Samsung Galaxy A7 2018

Sumber: (Santoso, 2021)

2. Metode

Metodologi penelitian ini dilakukan dengan beberapa tahapan, diantaranya:

A. Studi literatur

Pada tahap awal dilakukan studi terhadap beberapa artikel dan buku, yang berkaitan dengan teknik perancangan web aplikasi dengan metode *Single Page Application (SPA)*. Pada tahap ini juga dilakukan untuk menemukan fokus persoalan serta perumusan solusi yang akan ditawarkan pada penelitian ini.

B. Analisis permasalahan dan perumusan hipotesis.

Setelah didapatkan bahan literatur, pada tahap ini dilakukan analisis untuk merumuskan sebuah solusi yang akan ditawarkan.

C. Perancangan solusi

Pada tahap ini, akan diawali dengan perancangan sebuah desain yang akan dijadikan sebuah solusi dari permasalahan yang diangkat.

D. Implementasi solusi dan uji coba

Penerapan solusi dari perancangan yang telah dilakukan, akan diujikan dengan

mengimplementasikan ke dalam kode pemrograman.

E. Penarikan kesimpulan

Hasil pengujian kemudian dianalisis untuk mendapatkan rumusan beberapa kesimpulan yang dapat diambil dari penelitian.

3. Perancangan dan Desain Aplikasi

Untuk perancangan serta desain aplikasi didalam penelitian ini, aplikasi web dibuat dengan menggunakan pustaka (*library*) JavaScript yaitu ReactJS dan sebagai (*tools*) pendukung tambahan menggunakan *software* NodeJS serta *CSS Framework* Bootstrap.

A. ReactJS

ReactJS/ React adalah pustaka (*library*) JavaScript untuk membangun *User Interface* (Hakim, 2020). ReactJS merupakan kerangka kerja *open source* yang menggunakan *library* javascript untuk membuat *user interface* dan React biasa digunakan untuk menangani pengembangan pada aplikasi *single-page* dan aplikasi *mobile* (Nursaid et al., 2020). ReactJS adalah pustaka JavaScript berbasis komponen yang mengikuti paradigma pemrograman deklaratif. Tampilan deklaratif dapat digunakan untuk membuat UI interaktif kompleks yang berfungsi sebagai komponen presentasi web (Mukhiya & Hoang Hung, 2018). Jika dilihat dari arsitektur antarmuka pengguna, maka React dapat dikatakan hanya sebagai komponen V dalam konsep *MVC (Model-View-Controller)* (Bhalla et al., 2020). *ReactJS* adalah *view rendering JavaScript library* yang sangat bagus untuk menangani *JavaScript view component* yang berbasis *Single Page Application* dengan menggunakan *data-driven UI* dengan cara yang efisien, dinamis, dan menarik secara visual (Bustamin Prodi Teknik Mesin & Teknologi Industri Dewantara Palopo, 2021).

Instalasi ReactJS dapat dilakukan dengan beberapa cara salah satunya adalah dengan menggunakan NodeJS (Hamidin et al., 2018). *Library* atau paket yang diperlukan untuk membangun *layout web* dengan teknik SPA didalam react adalah React Router atau biasa dikenal dengan nama *react-router-dom*. Untuk melakukan instalasi dapat dilakukan melalui terminal, skripnya akan terlihat seperti berikut:

```
npx create-react-app my-app1  
cd my-app2  
npm start3
```

Penjelasan untuk kode skrip diatas adalah: label angka 1 untuk melakukan instalasi react sekaligus pembuatan folder atau direktori proyek web yang akan dibuat, label angka 2 menjelaskan bahwa untuk berpindah ke direktori atau proyek tujuan, dan label dengan angka 3 perintah untuk menjalankan aplikasinya. Pada saat melakukan instalasi penambahan *library* atau paket dapat dilakukan melalui terminal dan harus terhubung dengan koneksi internet.

Di dalam perancangan aplikasi web dengan teknik SPA ini, dibutuhkan beberapa *folder* yang akan dibagi kedalam beberapa kategori, diantaranya: 1) *Assets*, merupakan tempat menyimpan file-file yang berkaitan dengan multimedia seperti gambar, suara atau video. 2) *Layout*, yang didalamnya terdapat skrip kode untuk bagian *Header (Navbar)* dan *Footer*. 3) *Pages*, bagian yang paling penting dimana disini

akan dibuat file-file yang berkaitan dengan isi dari aplikasi web yang dibuat, seperti: *Home.js*, *News.js*, *Details.js*, *Contact.js*. Berikut ini adalah salah satu potongan kode pembuatan *Component* dari skrip dari file *Home.js*

```
import React, { Component } from 'react';1  
  
class Home extends Component{2  
  render(){  
    return(  
      <div>  
        ...  
        <p>Halo dari React JS</p>  
        ...  
      </div>  
    );  
  }  
}  
  
export default Home;3
```

Penjelasan dari potongan skrip diatas adalah sebagai berikut: 1) label dengan angka 1 merupakan perintah untuk meng-*import library built-in* dari React JS, diikuti dengan pembuatan *Component* dengan nama *Home* pada label 2, dan pada label 3 adalah perintah untuk menjalankan *Component Home* tersebut. Pembuatan aplikasi SPA ini membutuhkan 6 *Component* untuk total secara keseluruhan. Untuk menghubungkan ke-6 *Component* dibutuhkan *library* tambahan yaitu *react-router-dom*, dimana *library* ini berfungsi sebagai penghubung antar tiap-tiap *Component* dengan cara meng-*import*. Berikut ini adalah skrip untuk melakukan *import* untuk *react-router-dom*, perlu diketahui untuk melakukan *import* paket atau *library*, dibutuhkan koneksi internet serta penempatan skrip berada pada posisi baris diatas, sebelum pembuatan *Component* dibuat.

```
import { Switch, Route } from 'react-router-dom'
```

Konsep dasar dari perancangan layout web dengan teknik SPA dengan menggunakan React JS ini adalah memisahkan halaman-halaman (*pages*) yang akan ditampilkan, kedalam beberapa bagian dalam hal ini adalah file sebagai *Component*, kemudian dihubungkan dengan menggunakan *library* React Router (Lestari & Belluano, 2018). Jika dilihat pada gambar 1 ilustrasi tersebut menjelaskan bahwa teknik ini sedikit berbeda dengan teknik pembuatan web yang ada pada umumnya, dimana pembuatan halaman web dengan teknik tradisional dibuat murni secara terpisah, yang akan mengakibatkan laman web pada saat di-render oleh *browser* memerlukan waktu yang cukup lama, karena halaman akan di-render ulang setiap halaman akan dibuka (Elektronik Ilmu Komputer Udayana et al., 2020). *Single Page Application (SPA)* merupakan konsep arsitektur aplikasi web dimana tujuannya membuat cara kerja suatu web bekerja layaknya seperti aplikasi *desktop* (Azhari & Tanone, n.d.).

Cara kerja serta pembuatan layout web dengan React JS jika diperhatikan sedikit berbeda, ada beberapa folder dan file yang sudah tersedia (*built-in*), disini akan dilakukan perubahan sesuai dengan kebutuhan. Perubahan serta penambahan dapat dilakukan dengan melakukan pembuatan sebuah *Component*. Ada 3 file utama didalam React JS yang saling berhubungan, diantaranya: 1) *index.html*, pada bagian ini akan ditampilkan isi dari keseluruhan web itu sendiri yang didapatkan dari sumber kode yang berada di file *index.js* dan *App.js*. 2) *index.js*, didalam file ini terdapat kode yang mengarahkan ke *index.html*, sekaligus merujuk dari file sebelumnya dengan melakukan *import* terlebih dahulu dari file *App.js* dan 3) *App.js*, pada bagian ini dapat dilakukan penulisan kode program sesuai dengan kebutuhan, ini merupakan induk dari beberapa

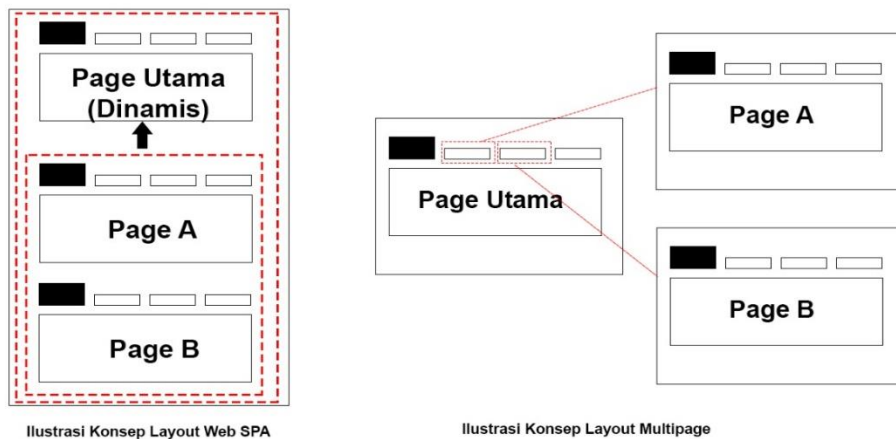
Component yang dibuat secara terpisah kemudian akan dipanggil pada file *App.js*

B. NodeJS

NodeJS merupakan bahasa skrip yang menawarkan pengintegrasian antara klien dan server, yang memungkinkan penggunaan kode kembali (*reusability*) dalam pengembangan aplikasi web serta alat yang sempurna untuk mengembangkan sebuah aplikasi yang cepat dan dapat diskalakan (Mukhiya & Hoang Hung, 2018). NodeJS adalah perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis web dan ditulis dalam sintaks bahasa pemrograman JavaScript (Wibowo & Wiguna, 2019). NodeJS adalah sistem perangkat lunak yang didesain untuk pengembangan aplikasi web. NodeJS disebut juga runtime environment (Rosman et al., n.d.). JavaScript merupakan bahasa pemrograman yang berjalan disisi klien (*browser*), maka dengan menggunakan NodeJS peran dari JavaScript dapat difungsikan sebagai bahasa pemrograman yang berjalan pada sisi server, layaknya seperti bahasa PHP, Ruby dan sebagainya. Umumnya dalam pembuatan web NodeJS berperan sebagai *back-end* sedangkan React sebagai *front-end*. JavaScript merupakan salah satu bahasa script website yang paling banyak digunakan untuk menambah manipulasi *script HTML* dan *CSS* pada sisi *client/browser* (Firdaus et al., 2019). Tidak seperti kebanyakan bahasa JavaScript yang dijalankan pada web browser, Node.js dieksekusi sebagai aplikasi server (Aldo & Munir, 2021).

C. Bootstrap

Fluid Grid Layout atau layout relatif konsep responsif web design dengan menjaga ukuran lebar (*width*) agar tetap relatif terhadap *viewport*, hal ini berdampak pada layout yang memungkinkan untuk beradaptasi serta menyesuaikan terhadap ukuran layar (Santoso, 2019). Kali pertama diperkenalkan oleh dua orang karyawan Twitter yaitu Mark Otto dan



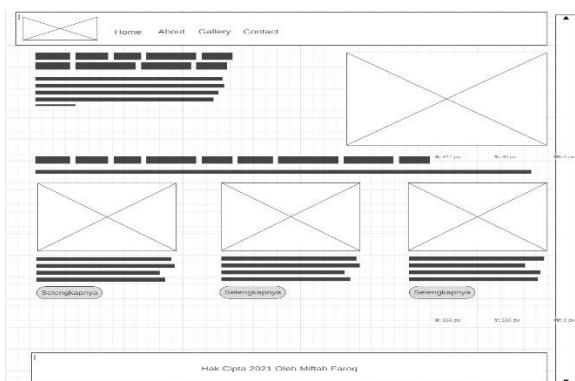
Sumber: (Santoso, 2021)
Gambar1. Ilustrasi Single Page Application (SPA)

Jacob Thornton, hal ini bertujuan agar dalam pengembangan sebuah web aplikasi akan lebih cepat dan sekaligus *layout web* dapat secara otomatis mengadopsi ukuran perangkat pengguna (*user*) baik pada saat diakses menggunakan layar besar komputer *Desktop PC* atau layar yang berukuran kecil seperti di *Tablet* dan *Smartphone*.

D. Desain Layout

Desain mengacu kepada struktur sebuah website pada umumnya, yang terdiri dari beberapa bagian, diantaranya: *Header*, *Menu* atau *Navigation*, *Content* dan *Footer*.

Untuk perancangan serta desain aplikasi didalam penelitian ini, aplikasi web dibuat dengan menggunakan pustaka (*library*) JavaScript yaitu *ReactJS* dan sebagai (*tools*) pendukung tambahan menggunakan software *NodeJS* serta *CSS Framework Bootstrap*.



Sumber: (Santoso, 2021)
Gambar 2. Wireframe Layout Web

Gambar 2 menjelaskan untuk desain layout web yang dibuat terdiri dari beberapa bagian (*section*), bagian utama *Header*, *Navigation*,

Content, dan *Footer*. Halaman utama (*Content*) akan dibuat secara dinamis, dimana isi konten akan berubah-ubah secara dinamis sesuai masukan dari luar, berdasarkan *user-interface* yang ada di *front-end*. Isi dari konten akan diambil atau di-render dari *library react*, file terkait diantaranya: *Home.js*, *News.js*, *Contact.js*, *Details.js*, *Navbar.js* dan *Footer.js*.

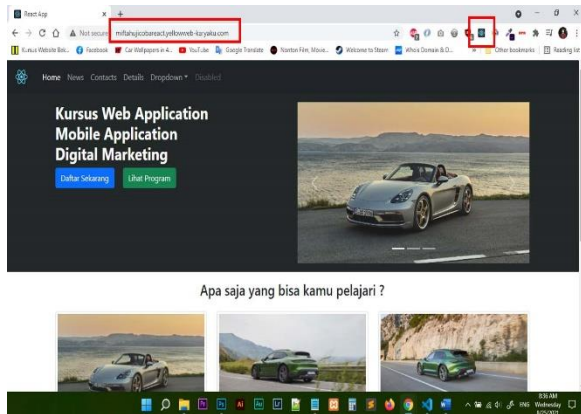
III. HASIL DAN PEMBAHASAN

Pada bagian ini akan dijelaskan seputar implementasi model dan pengujian.

Penelitian ini mencoba melakukan penambahan metode atau teknik didalam melakukan perancangan desain sebuah website, dengan mengacu pada penelitian terdahulu yang hanya mengandalkan *CSS Framework Bootstrap* untuk merancang *layout* atau tampilan sebuah *website*. Teknik *Responsive Web Design (RWD)* Serta Penerapannya Dalam Rancang Bangun *Layout* (Santoso, 2019).

A. Implementasi Model

Implementasi dalam pembuatan layout web dengan teknik *Single Page Application (SPA)*, akan terlihat secara keseluruhan pada Gambar 3

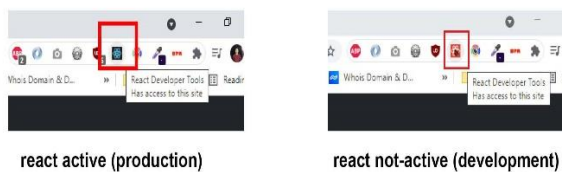


Sumber: (Santoso, 2021)
Gambar 3. Desain *Layout Web*

Penelitian dilakukan dengan melakukan uji coba membuat 2 (dua) buah halaman website yang sudah dihosting dengan sub-domain yang berbeda, *website* dengan konsep penerapan *layout* teknik *Single Page Application (SPA)* dan web tanpa teknik SPA atau tradisional. Domain dapat diakses

di <http://miftahujicobareact.yellowweb-karyaku.com/> dan di <http://miftahujicobanoreact.yellowweb-karyaku.com/> sebagai pembandingan untuk uji coba.

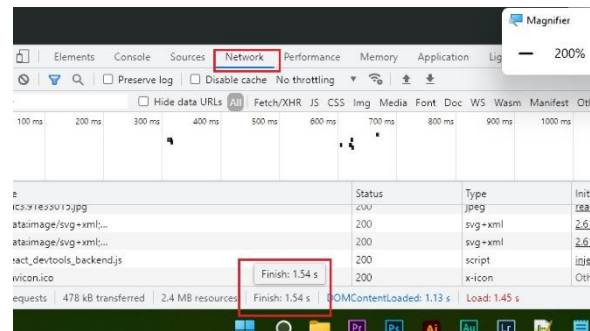
Untuk mengetahui apakah teknik SPA sudah berjalan dengan baik, maka perlu dilakukan fitur tambahan atau ekstensi ke dalam *browser*, didalam penelitian ini menggunakan ekstensi *React Developer Tools for Chrome*, jika indikator menampilkan logo ReactJS berwarna merah mengartikan bahwa *website* dibuat dengan menerapkan teknik *layout* SPA namun kondisi masih dalam tahap pengembangan (*development*) dan posisi masih berada dilokal komputer atau belum dihosting (*deploy*), sedangkan jika indikator menampilkan logo ReactJS warna hitam dan warna hijau tosca, maka dapat dikatakan web menerapkan teknik *layout* SPA sudah jadi (*fully production*) dapat dilihat pada gambar 4.



Sumber: (Santoso, 2021)
Gambar 4. Indikator Ekstensi ReactJS

Uji coba lokal ataupun sudah dalam keadaan di *deploy* dengan menerapkan teknik *layout* SPA sudah dapat dikatakan berjalan dengan baik,

tahapan selanjutnya adalah membandingkan dari segi kecepatan, perpindahan antara halaman yang satu dengan yang lain, jika membuka halaman web yang mengadopsi teknik SPA, ternyata lebih cepat karena halaman tidak dire-load ulang setiap kali bernavigasi atau berpindah-pindah. Jika membuka halaman web tanpa menerapkan teknik SPA, akan terasa lebih lama, karena setiap kali halaman dibuka, browser akan membaca ulang kembali, hal ini tentunya sangat tidak efektif dan efisien, mengingat waktu akan banyak terbuang, karena harus menunggu keseluruhan halaman ditampilkan atau di-load secara keseluruhan.



Sumber: (Santoso, 2021)
Gambar 5. *Network Indikator Inspect Element*

Pada gambar 5 menjelaskan bahwa indikator yang digunakan untuk mengetahui berapa lama *browser* membutuhkan waktu untuk menampilkan halaman secara sempurna, dipenelitian ini menggunakan fitur pada tab *network* di menu *inspect element built-in* dari browser.

Tabel 2. Hasil Uji Coba Implementasi SPA

No	Halaman (Pages)	Uji Coba			
		Google Chrome		Firefox	
		ReactJS	Non-ReactJS	ReactJS	Non-React JS
1	Home (index)	2.16 s	1.46 s	2.53 s	1.69 s
2	News	0 ms	273 ms	0 ms	518 ms
3	Contacts	0 ms	2.87 s	0 ms	5.33 s
4	Details	0 ms	1.98 s	0 ms	8.29 s

Sumber: (Santoso, 2021)

Jika dilihat dari tabel 2 perbandingan diatas antara halaman atau *layout web* yang mengadopsi teknik SPA dan yang tidak, terlihat jelas perbedaan yang sangat signifikan. Uji coba dengan menggunakan *browser* Chrome didapati pada halaman utama (*index*) dengan ReactJS waktu yang diperlukan untuk merender keseluruhan halaman dengan sempurna membutuhkan sekitar 2.16 detik, sedikit lebih lama jika dibandingkan dengan halaman utama Non-ReactJS yang memerlukan waktu 1.46 detik atau selisih 0.7

detik. Pada browser Firefox dapat dilihat dengan menggunakan ReactJS waktu yang diperlukan sebesar 2.53 detik dan Non-ReactJS sebesar 1.69 detik terdapat selisih sebesar 0.84 detik. Hal ini diakibatkan karena halaman web yang menggunakan ReactJS harus terlebih dahulu *me-load library* atau paket yang nanti akan digunakan untuk implementasi dari konsep SPA dan dapat diperhatikan untuk perpindahan antar halaman (*page*), tidak ada sama sekali perubahan waktu yang dibutuhkan untuk menampilkan keseluruhan isi halaman, berbeda dengan halaman (*page*) Non-ReactJS, waktu yang diperlukan selalu berubah sesuai dengan *assets* seperti gambar, *styling* css, yang nanti akan ikut mempengaruhi kinerja dari browser tersebut pada saat rendering halaman.

Hasil dari penelitian ini didapatkan bahwa untuk memaksimalkan agar mesin perambah (*browser*) lebih cepat dalam menampilkan konten atau isi dari keseluruhan halaman secara sempurna, dapat menggunakan *library* ReactJS dengan teknik *Single Page Application (SPA)*. ReactJS dapat dipadukan dengan *libray* atau paket yang lain, salah satunya adalah *CSS Framework Bootstrap* agar tampilan dapat menyesuaikan dibanyak perangkat sekaligus (*Mobile Friendly*) sehingga tampilan akan terlihat elegan, tidak hanya berfokus kepada kecepatan saja, melainkan juga tetap mengedepankan dari segi estetika sebuah desain layout web.

REFERENSI

- Aldo, D., & Munir, Z. (2021). Aplikasi Pelayanan Pada Klinik Kemina Dental Care Berbasis React.js dan Database NoSQL. *Jurnal Ilmiah Ilmu Komputer*, 16(2). <https://doi.org/10.30872/jim.v16i2.6532>
- Bhalla, A., Garg, S., & Singh, P. (2020). PRESENT DAY WEB-DEVELOPMENT USING REACTJS. *International Research Journal of Engineering and Technology*. www.irjet.net
- Bustamin Prodi Teknik Mesin, S., & Teknologi Industri Dewantara Palopo, A. (2021). Aplikasi Dekstop Multi Platform untuk Redis Client Framework Electron JS dan React JS. *DEWANTARA. J. Tech*, 02(01).
- Elektronik Ilmu Komputer Udayana, J., Wikardiyan, A., Made Widiartha, I., Arida Ayu Rahning Putri, L., Informatika, T., & Matematika dan Ilmu Pengetahuan Alam, F. (2020). *Perancangan dan Implementasi Sistem Manajemen Proyek Perangkat Lunak Menggunakan Teknologi Single Page Application*. 8(4), 2654-5101.
- Firdaus, A., Widodo, S., Sutrisman, A., Gading, S., Nasution, F., Mardiana, R., Komputer, J. T., Negeri, P., & Palembang, S. (2019). RANCANG BANGUN SISTEM INFORMASI PERPUSTAKAAN MENGGUNAKAN WEB SERVICE PADA JURUSAN TEKNIK KOMPUTER POLSRI. *Jurnal Informanika*, 5(2). www.kursuswebsite.org
- Hakim, L. (2020). *Jalan Pintas Menjadi Master ReactJS* (L. Hakim, Ed.; 2nd ed.). Penerbit Lokomedia.
- Hamidin, D., Made, I., Dharma, Y., & Luthfiah, A. N. (2018). SariAsih No.54-Bandung 40151. In *Jurnal Teknik Informatika* (Vol. 10, Issue 2).
- Lestari, P., & Belluano, L. (2018). PENGEMBANGAN SINGLE PAGE APPLICATION PADA SISTEM INFORMASI AKADEMIK. *ILKOM Jurnal Ilmiah*, 10.
- Azhari, A. L., & Tanone, R. (n.d.). *Analisis Penerapan Single Page Application Menggunakan Teknologi AJAX dan REST API (Studi Kasus: Sistem Informasi Reservasi Wisma Tamu UKSW)*.
- Mukhiya, S. K., & Hoang Hung, K. (2018). *An Architectural Style for Single Page Scalable Modern Web Application*. 5(4), 6-13.
- Nurpandi, F., & Rosid Liki, D. M. (2021). Perancangan dan Pembangunan Aplikasi Electronic Menu Restoran. *Media Jurnal Informatika*, 13(1). <http://jurnal.unsur.ac.id/mjinformatika>
- Nursaid, F. F., Hendra Brata, A., & Kharisma, A. P. (2020). Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus: Toko Uda Fajri). *J-Ptiik.Ub.Ac.Id*, 4(1), 46-55.
- Rosman, J. A., Imron, & Prasetyo, M. H. (n.d.). *Sistem Informasi Bimbingan Tugas Akhir dan Skripsi Online Berbasis Web*.
- Santoso, M. F. (2019). *TEKNIK RESPONSIVE WEB DESIGN (RWD) SERTA PENERAPANNYA DALAM RANCANG BANGUN LAYOUT WEB* (p. 8).
- Santoso, M. F. (2021). *TEKNIK SINGLE PAGE APPLICATION (SPA) LAYOUT WEB DENGAN MENGGUNAKAN REACT JS DAN BOOTSTRAP* (p. 6).

Wibowo, A. T., & Wiguna, A. S. (2019).
Pemanfaatan Teknologi Single Page
Application (SPA) dalam Pembuatan Aplikasi
Feedback Dosen dari Mahasiswa Sebagai
Bentuk Pengawasan Lembaga Terhadap
Kinerja Dosen di Bidang Pengajaran.
SMARTICS Journal, 5(1), 34-43.
<https://doi.org/10.21067/smartics.v5i1.3327>