

Pengembangan Menu Digital Menggunakan ReactJs Implementasi Hasil Belajar Studi Independen di Frontend Engineering Program Ruangguru CAMP (Career Acceleration Bootcamp)

Kevin Juan^{#1}, Setia Budi^{#2}

#Program Studi S1 Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Maranatha

Jln. Prof. Drg. Surya Sumantri No. 65, Sukawarna, Bandung, Indonesia

¹1972002@maranatha.ac.id

²setia.budi@it.maranatha.edu

Abstract — In the digital era, websites are an important thing for a business. Here the web developer is needed. One part of a web developer's job is a frontend developer. Ruangguru, in collaboration with the Kampus Merdeka, created a boot camp called Ruangguru CAMP (Career Acceleration Bootcamp) to foster the ability to build responsive and interactive website applications that have two learning paths, namely frontend engineers and backend developers. In this digital era, the food industry, especially restaurants, has adopted a lot of technology in the form of moving from menus on paper to digital menus. To display responsive and efficient menus, the development of a digital menu can be done using the ReactJs Javascript library. With ReactJs, developers make the user interface display into several different components. By using React Router, developers can use the concept of a single page application which allows display navigation between various components in React, allows changes to URLs, and keeps the user interface in sync with the URL. Application development using ReactJs and React Router will make a digital menu web application responsive and efficient with the actions taken by the user.

Keywords— Digital Menu, Frontend Engineer, ReactJS, React Router, Single Page Application

I. PENDAHULUAN

Di era digital seperti ini, website memang memiliki peran yang sangat penting bagi sebuah bisnis. Dengan mengembangkan website, perusahaan dapat menjangkau lebih banyak pelanggan. Di sinilah peran web developer dibutuhkan. Salah satu bagian pekerjaan dari web developer adalah frontend developer.

Tugas Frontend Developer adalah mengembangkan tampilan visual atau user interface pada halaman website. Mereka akan bertanggung jawab untuk mengimplementasikan elemen visual dan elemen interaktif yang digunakan oleh pengguna saat mereka mengoperasikan website pada halaman web browser

Ruangguru telah mengadakan Frontend Engineering Bootcamp untuk mahasiswa semester 6 ke atas untuk membina mereka agar dapat membangun aplikasi website yang responsif & interaktif dengan menggunakan teknologi terbaru yang banyak digunakan di industri. Ruangguru juga telah menyediakan para instruktur yang telah berpengalaman untuk membagikan ilmu - ilmu untuk menjadi Frontend Engineer atau Frontend Developer.

Pada program Studi Independen Ruangguru bersama Kampus Merdeka, materi ajar yang akan diberikan kepada mahasiswa akan disusun oleh tim ahli pengembangan konten dari Ruangguru bersama para instruktur berpengalaman dan didukung oleh teknologi canggih untuk mempermudah proses pembelajaran.

Pada era digital ini tempat makan banyak mengadopsi teknologi berupa perpindahan dari menu dalam bentuk kertas menjadi menu digital. Menurut Mark Yi-Cheon Yim dan Chan Yun Yoo, dalam beberapa tahun terakhir sebuah industri semakin banyak untuk mengadopsi menu digital dalam bentuk web dan tablet dengan tujuan untuk meningkatkan efisiensi dalam penelusuran pelanggan [1].

Untuk menampilkan menu yang responsif dan efisien maka dapat dikembangkan dengan menggunakan library javascript reactjs. Dengan reactjs memungkinkan untuk membuat tampilan menu digital dengan menggunakan konsep single page application, menurut A. Mesbah dan A. van Deursen mendefinisikannya sebagai “antarmuka web satu halaman yang terdiri dari komponen individual yang dapat diperbaharui atau diganti secara independen, sehingga seluruh halaman web tidak

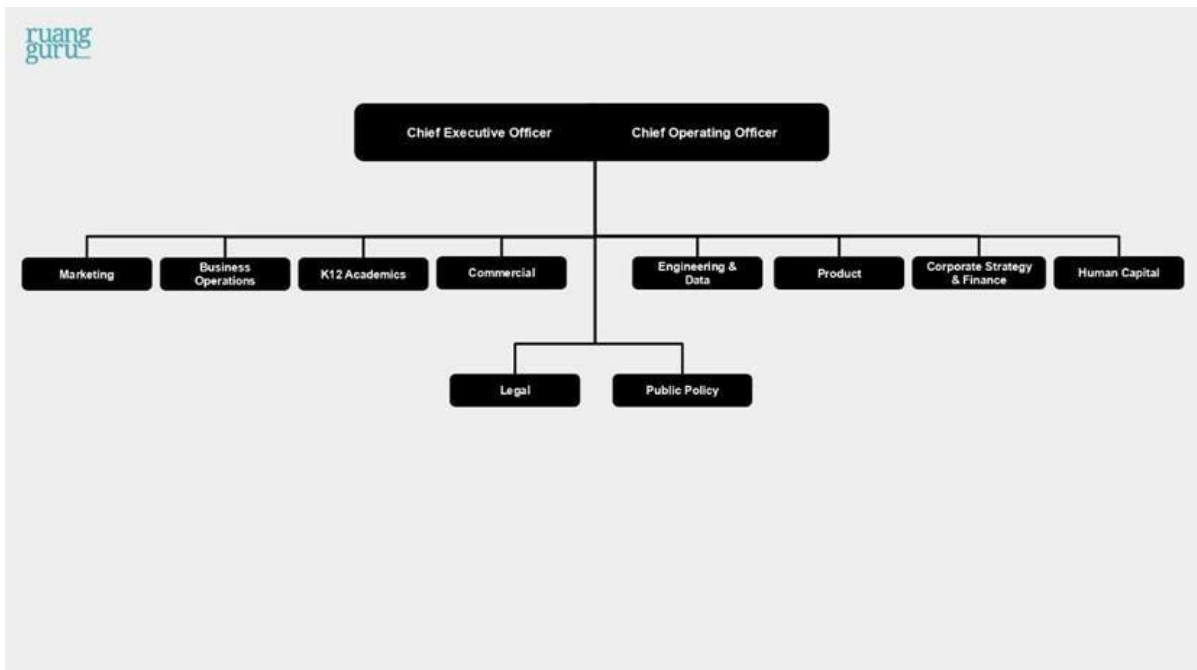
perlu dimuat ulang pada setiap tindakan pengguna” [2]. Dengan adanya konsep single page application pengguna yang berinteraksi dengan menu digital berbasis web akan mendapatkan response yang lebih cepat karena halaman yang ditampilkan tidak akan memuat ulang keseluruhan halaman namun hanya komponen - komponen yang berubah yang akan dimuat ulang. Dengan menggunakan React Router dapat mengaplikasikan konsep single page application. React Router sendiri merupakan sebuah standard library untuk routing di react, yang memungkinkan navigasi tampilan di antara berbagai komponen di react, memungkinkan perubahan pada URL, serta menjaga UI tetap sinkron terhadap URL [3].

II. PROFIL PERUSAHAAN

Ruangguru CAMP (Career Acceleration Bootcamp) adalah salah satu kolaborasi antara Ruangguru dengan Kampus Merdeka, berbentuk kursus studi online berdurasi empat bulan, menyediakan pengalaman pembelajaran yang komprehensif melalui webinar training dengan ahli, serta pembelajaran mandiri.

Ruangguru CAMP memiliki dua program studi yaitu frontend engineering dan backend engineering. Saya dalam mengikuti kegiatan di Ruangguru CAMP ini memilih untuk mengambil program studi frontend engineering.

Kegiatan yang dilakukan selama mengikuti pembelajaran di Ruangguru CAMP ini adalah mentoring yaitu pendampingan peserta Ruangguru CAMP dengan mentor melalui kelompok belajar kecil yang terdiri dari kurang lebih dua puluh lima orang dalam satu kelompok kecil, tujuannya untuk membahas kesulitan yang dialami selama melakukan pembelajaran di Ruangguru CAMP. Kegiatan selanjutnya adalah pembelajaran secara mandiri yaitu kegiatan belajar secara mandiri melalui sumber yang kredibel dari Ruangguru. Ruangguru CAMP juga memberikan tugas akhir dari permasalahan industri saat ini.



Gambar 1. Struktur Organisasi Ruangguru

III. LANDASAN TEORI

A. Hypertext Markup Language

Hypertext Markup Language atau HTML adalah sebuah standard markup language yang digunakan untuk membuat dan menyusun halaman web. HTML bukanlah suatu bahasa pemrograman, melainkan bahasa markup, yang artinya adalah sebuah sistem yang digunakan untuk mengidentifikasi dan mendeskripsikan berbagai komponen dalam dokumen, seperti halnya header, paragraph, dan list, yang menunjukkan struktur dasar dokumen tersebut [5].

B. Cascading Style Sheet

CSS merupakan singkatan dari Cascading Style Sheet, berguna untuk mempercantik tampilan dari sebuah dokumen yang berformat markup language seperti HTML ataupun XML. Ketika di-render pada web browser, CSS ini berperan penting dalam membuat hasil render HTML ataupun XML tersebut menjadi lebih indah untuk dilihat. CSS dapat mengontrol bagaimana konten cara suatu tampilan web dapat terlihat indah, dengan menambahkan beberapa metode dan pengaturan untuk memberikan spesifikasi ilustrasi dari sebuah konten, baik yang bersifat visual maupun non-visual [5].

C. Javascript

Dalam membangun sebuah aplikasi berbasis web, Javascript merupakan salah satu bahasa pemrograman yang sering digunakan untuk membuat tampilan antarmuka dengan animasi, konten yang interaktif dan visual efek yang dinamis. Dengan tingginya penggunaan Javascript dalam pengembangan website, kemudian para pengembang Javascript membuat Javascript yang ditulis dapat digunakan secara berulang untuk menyelesaikan tugas yang sama tanpa menulis kode kembali. [6]. Javascript diperkenalkan pada tahun 1995 sebagai cara untuk menambahkan program ke halaman web pada browser Netscape Navigator [7]. Web browser bukan satu – satunya platform yang menggunakan Javascript. Beberapa database, seperti MongoDB dan CouchDB, menggunakan Javascript sebagai bahasa scripting dan query mereka. Beberapa platform untuk pemrograman desktop dan server, terutama proyek NodeJS, menyediakan environment untuk pemrograman Javascript di luar browser. Javascript sendiri merupakan single-threaded programming language yang artinya Javascript hanya bisa melakukan satu hal dalam satu waktu, namun meskipun Javascript merupakan single-threaded Javascript secara default akan mengeksekusi kode dengan asynchronous [8].

- 1) *ES6*: Javascript ES6 (juga dikenal sebagai ECMAScript 2015 atau ECMAScript 6) adalah versi yang lebih baru dan ES6 juga tidak dapat langsung dikompilasi langsung di browser, sehingga dibutuhkan compiler dari ES6, untuk itu diperlukan kompilasi menggunakan Babel, yang pada akhirnya menghasilkan Javascript yang kompatibel dengan browser [9].

D. Node JS

Javascript merupakan bahasa pemrograman yang berfokus pada browser, namun Javascript juga bisa juga digunakan untuk membuat aplikasi di luar browser atau server. Agar Javascript dapat dieksekusi di luar browser dibutuhkan sebuah environment, yang salah satunya adalah NodeJs. NodeJs adalah cross-platform runtime environment [9] untuk mengeksekusi kode Javascript di luar browser. NodeJs sendiri memilih V8 untuk menjadi mesin yang menggerakkan Node.js pada tahun 2009. V8 sendiri merupakan Javascript Engine yang digunakan oleh Google Chrome[10].

- 1) *NPM* : NPM adalah singkatan dari Node Package Manager. NPM merupakan sebuah standard package manager untuk Node.js. NPM sendiri merupakan sebuah package manager yang digunakan untuk mengelola package yang dibuat oleh developer, termasuk package yang dibuat oleh kita sendiri [11].

E. Responsive Design

Dalam meningkatkan pengalaman seorang User dalam menggunakan sebuah website pada device yang berbeda beda setiap User makan akan lebih baik untuk menerapkan konsep responsif desain. Responsif desain sendiri merupakan sebuah cara untuk membuat sebuah tampilan halaman website menjadi lebih baik dalam segala device. Hal itu dikarenakan sebuah halaman website akan menyesuaikan sebuah desain dengan viewport pada tiap device yang berbeda [13].

F. ReactJS

ReactJs adalah sebuah *library* Javascript yang dibuat oleh Facebook untuk membangun *user interface*. ReactJs dibuat dengan menggunakan Javascript ES6 dan JSX. ReactJs adalah *library* Javascript yang sangat populer dan banyak digunakan oleh *developer* untuk membuat aplikasi *web*. ReactJs, atau yang biasa dikenal dengan React, merupakan salah satu *library* dari Javascript yang bersifat *open source* yang berarti dapat diakses oleh banyak kalangan untuk memfasilitasi tampilan antarmuka yang bersifat interaktif, *stateful*, dan mudah digunakan. React berupaya untuk menyediakan kecepatan

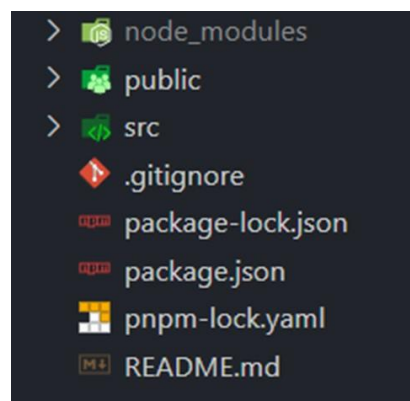
kesederhanaan dan perluasan. Beberapa fitur yang paling mencolok dan sering digunakan adalah komponen *stateful* dan DOM [14].

- 1) *Virtual Document Object Model*: Virtual DOM adalah sebuah konsep yang digunakan ReactJs untuk mengoptimalkan proses rendering. Virtual DOM adalah sebuah representasi dari DOM yang ada di browser, bentuknya adalah sebuah objek mirip dengan DOM. Virtual DOM ini akan di-update setiap kali terjadi perubahan pada state atau props. Setelah itu, ReactJs akan melakukan perbandingan antara Virtual DOM yang baru dengan Virtual DOM yang lama. Jika terdapat perbedaan, maka ReactJs akan melakukan proses rendering ulang. Props sendiri merupakan sebuah objek yang berisi data yang akan digunakan oleh komponen yang bersifat read-only yang artinya tidak dapat diubah props tersebut. Sedangkan state adalah sebuah objek yang berisi data yang akan digunakan oleh komponen namun state bersifat read-write yang artinya developer dapat mengubah state tersebut.
- 2) *Router DOM*: Router adalah proses yang akan mengarahkan user ke halaman yang berbeda berdasarkan action atau request yang dilakukan. Pada ReactJs pengembang dapat melakukan routing dengan React Router DOM. React Router DOM sendiri merupakan package NPM yang digunakan untuk membuat sebuah single page Application (SPA) dengan kata lain aplikasi yang memiliki banyak halaman ataupun komponen, namun perpindahan halamannya tidak akan melakukan reload browser, melainkan konten yang ditampilkan diambil secara dinamis berdasarkan URL-nya. Sehingga saat user mengetik URL tertentu ke browser dan jika path URL ini sesuai dengan Route yang ada di dalam Router, user akan diarahkan ke Route tertentu. Route sendiri merupakan sebuah komponen yang digunakan untuk mendefinisikan dan merender komponen berdasarkan path yang ditentukan. Komponen Route akan menerima komponen lain dan melakukan proses rendering.

IV. HASIL PEKERJAAN

A. Proses Inisialisasi Proyek

Create React App (CRA) adalah sebuah rangkaian alat untuk membangun sebuah single page application dengan menggunakan react. Dengan menggunakan CRA pengembang tidak perlu untuk melakukan konfigurasi yang rumit untuk membangun sebuah proyek ReactJs. Dapat diartikan bahwa CRA merupakan sebuah starter kit untuk membuat proyek ReactJs yang menghasilkan output proyek ReactJs yang berisi file, folder, file konfigurasi dan package dasar (basic package).

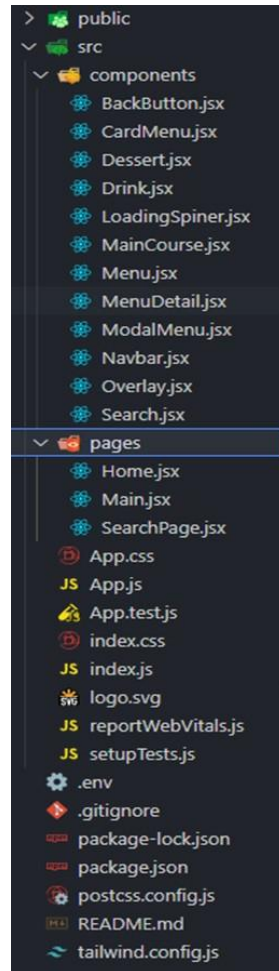


Gambar 2. Struktur Proyek Create React App

Gambar 2 merupakan sebuah struktur hasil output dari Create React App. Struktur ini terdiri dari folder public, node_modules dan file – file package.json yang berisikan informasi mengenai proyek yang dibuat, serta seluruh dependency (package terkait) yang dibutuhkan untuk proyek yang dibuat. Adapun folder src yang merupakan folder inti yang menyimpan file - file Javascript, serta komponen yang digunakan untuk membangun sebuah proyek React.

Pada pengembangan aplikasi web menu digital ini menggunakan Create React App untuk membangun sebuah aplikasi React.

Pada Gambar 3 menampilkan struktur proyek menu digital menggunakan CRA. Pada folder src terdapat sub-folder yang terdiri dari folder components yang berisikan file - file component yang dapat digunakan secara reusable. Adapun sub-folder pages yang berisikan file Javascript yang akan me-render halaman pada user interface.



Gambar 3. Struktur Menu Digital Menggunakan Create React App

B. Proses Pembuatan Komponen

Sebuah website atau aplikasi yang dibangun dapat dipisah menjadi beberapa bagian kecil, setiap bagian kecil yang dibuat tersebut disebut dengan komponen. ReactJs sendiri menggunakan konsep komponen yang akan selalu menghasilkan sebuah element HTML yang akan ditampilkan di browser.

Pada pengembangan aplikasi menu digital ini menggunakan functional component dalam pengembangannya. Functional Component ini merupakan React Component yang dibuat menggunakan function.

```
1.   const BackButton = () => {  
2.   const navigate = useNavigate();  
3.   return (  
4.     <button  
5.       onClick={() => {  
6.         navigate(-1);  
7.       }}  
8.     >  
9.     <TiArrowBack className="text-3xl" />  
10.  </button>  
11.  );  
12.  };
```

Diatas merupakan salah satu *React Component* dalam pembuatan aplikasi *web* menu digital untuk menghasilkan element HTML *Button* untuk navigasi ke halaman sebelumnya.

```

1.     return (
2.         <
3.             {loading && <LoadingSpiner />}
4.             {!loading && (
5.                 <
6.                     {loading2 && <LoadingSpiner />}
7.                     {!loading2 && (
8.                         <ModalMenu
9.                             isOpen={isOpen}
10.                            onClose={onClose}
11.                            overlay={overlay}
12.                            data={dataSatuan}
13.                        />
14.                    )}
15.                </>
16.                <SimpleGrid columns={{[1, 2]} spacing={8} m="10px">
17.                    {data.map((el) => {
18.                        return (
19.                            <CardMenu
20.                                data={el}
21.                                klik={() => {
22.                                    handleklik(el.id);
23.                                }}
24.                                country={menuNames}
25.                                key={el.id}
26.                            />
27.                        );
28.                    })}
29.                </SimpleGrid>
30.            </>
31.        )}
32.    </>
33.    );

```

Kode di atas merupakan salah satu *React Component* dalam pengembangan menu digital untuk menampilkan menu berdasarkan jenis makanan dari wilayah tertentu.

C. Proses Fetching Data dari API

API (Application Programming Interface) merupakan interface yang dapat menghubungkan satu aplikasi dengan aplikasi lainnya. API dapat diartikan sebagai sebuah perantara antara berbagai aplikasi yang berbeda, baik itu dalam satu platform yang sama maupun lintas platform.

Dalam pengembangan sebuah aplikasi biasanya memerlukan data yang didapatkan dari sumber lain. Untuk mendapatkan data dari sumber lain menggunakan API dapat menggunakan fungsi `Fetch()`. Fungsi ini akan mengembalikan sebuah promise yang berisi data yang di-fetch.

Dalam *React* biasanya mengambil data pada saat pertama kali komponen dibuat dan ditampilkan di browser.

```

1.     const getFoods = async (menuName) => {
2.         const chek = localStorage.getItem(menuName);

```

```
3.
4.     if (chek) {
5.         setData(JSON.parse(chek));
6.         setLoading(false);
7.     } else {
8.         const req = await fetch(
9.     `https://api.spoonacular.com/recipes/complexSearch?apiKey=${process.env.REACT_APP_API_KEY}&cuisine=${
10.     menuName}`
11.     );
12.     const resp = await req.json();
13.     const foods = resp.results.map(async (el) => {
14.         const url =
15.     `https://api.spoonacular.com/recipes/${el.id}/information?includeNutrition=false&apiKey=${process.env.REACT_
16.     APP_API_KEY}`;
17.         const req = await fetch(url);
18.         const resp = await req.json();
19.         return resp;
20.     });
21.     const temp = await Promise.all(foods);
22.     localStorage.setItem(menuName, JSON.stringify(temp));
23.     setData(temp);
24.     setLoading(false);
25. }
```

Kode di atas merupakan sebuah function untuk mengambil data dari API dan menyimpannya pada sebuah state untuk menyimpan data yang sudah diambil dari API.

D. Proses Menggabungkan Komponen pada Suatu Halaman

Bagian bagian yang telah dibagi menjadi beberapa bagian kecil yang disebut dengan komponen akan disatukan pada satu tempat yang nantinya akan di-render dan ditampilkan di browser.

```
1.     function Search() {
2.         const { isOpen, onOpen, onClose } = useDisclosure();
3.         const [overlay] = useState(<Overlay />);
4.         const [searchValue, setSerachValue] = useState("");
5.         const navigate = useNavigate();
6.
7.         const handleSearch = (e) => {
8.             e.preventDefault();
9.             navigate("/menu/search/" + searchValue);
10.            onClose(setSerachValue(""));
11.        };
12.        return (
13.            <
14.            <Modal isOpen={isOpen} onClose={onClose}>
```

```

15.     {overlay}
16.     <ModalContent m="5">
17.       <ModalBody px="2">
18.         <form
19.           onSubmit={ (e) => {
20.             handleSearch(e);
21.           }}
22.         >
23.           <HStack m="2" justifyContent="space-around">
24.             <AiOutlineSearch className="text-2xl mx-2" />
25.             <Input
26.               value={ searchValue }
27.               onChange={ (e) => setSerachValue(e.target.value)}
28.               type="text"
29.               variant="unstyled"
30.               placeholder="Search Menu..."
31.             ></Input>
32.           </HStack>
33.           <h1> {searchValue}</h1>
34.         </form>
35.       </ModalBody>
36.     </ModalContent>
37.   </Modal>
38.
39.   <button onClick={onOpen}>
40.     <AiOutlineSearch className="hover:text-amber-700" />
41.   </button>
42. </>
43. );
44. }

```

Kode di atas merupakan sebuah *function search* pada menu digital yang menggabungkan dari beberapa komponen - komponen yang telah dipecah menjadi bagian - bagian kecil sehingga kode pada halaman *search* dapat lebih ringkas dan mudah untuk diatur.

E. Proses Routing

Routing adalah suatu bagian penting dalam menciptakan sebuah aplikasi single page application. Routing sendiri merupakan sebuah proses yang akan mengarahkan user ke halaman yang berbeda berdasarkan action atau request yang dilakukan. Pada ReactJs kita dapat melakukan routing menggunakan package NPM yang bernama React Router DOM.

```

1.   function App() {
2.     return (
3.       <>
4.         <Routes>
5.           <Route path="/" element={<Home/>} />
6.           <Route path='menu' element={<Main/>}>
7.             <Route path=':menuNames'>
8.               <Route index element={<Menu/>} />
9.               <Route path=":id" element={<MenuDetail/>} />
10.            </Route>
11.            <Route path='search'>
12.              <Route path=":query" element={<SearchPage/>} />
13.            </Route>

```

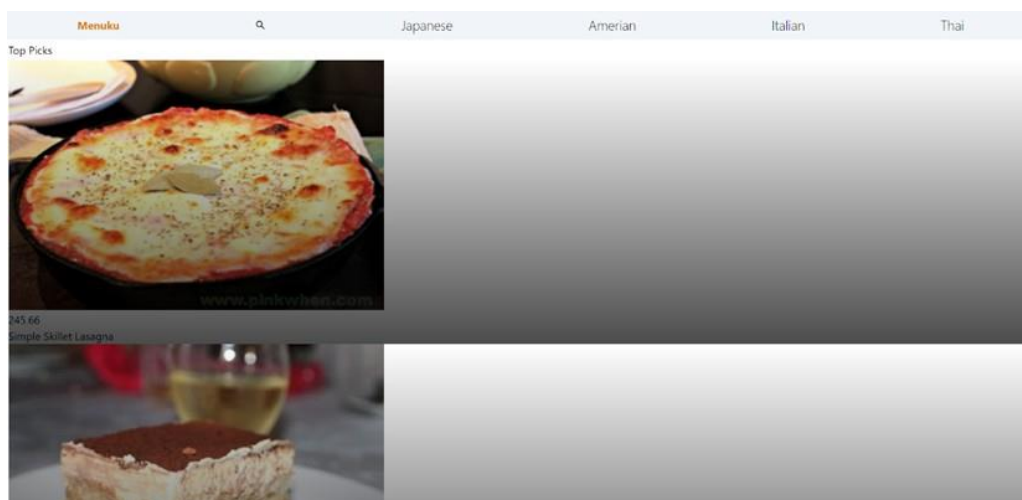


```
14.     </Route>
15.     <Route path="*" element={<h2> not found</h2>} />
16.     </Routes>
17.     </>
18.     );
19.     }
```

Kode di atas merupakan proses routing pada aplikasi web digital menu. Route sendiri merupakan sebuah komponen yang digunakan untuk mendefinisikan dan merender komponen berdasarkan path yang sudah ditentukan. Komponen Route akan menerima komponen lain dan me-render-nya.

F. Tampilan Sebelum Menggunakan Chakra UI dan Tailwind CSS

Sebelum menerapkan sebuah style pada tampilan alangkah baiknya untuk membuat tampilan sebuah halaman website secara keseluruhan tanpa menggunakan CSS, setelah semua data dapat ditampilkan dengan baik maka selanjutnya pengembang atau developer dapat menggunakan framework CSS diantaranya yaitu Chakra UI dan Tailwind CSS. Dengan menggunakan framework CSS akan memudahkan dalam melakukan styling pada halaman website dan akan lebih mudah untuk menerapkan konsep responsif desain pada halaman website yang akan dibuat.



Gambar 4. Tampilan Menu Digital Sebelum Menggunakan Chakra UI dan Tailwind CSS

Gambar 4 merupakan sebuah tampilan home pada saat sebelum menggunakan framework Chakra UI dan Tailwind CSS. Gambar yang ditampilkan sangat besar dan tidak indah untuk dilihat. Tampilan juga belum menggunakan konsep responsive design sehingga akan menampilkan hasil yang tidak responsif ketika ada perubahan device yang berbeda.

G. Proses Styling Menggunakan Chakra UI dan Tailwind CSS

CSS pada pembuatan sebuah aplikasi web itu memiliki peran penting untuk mempercantik hasil render HTML. CSS mempunyai suatu konsep Responsive Web Design, yaitu sebuah konsep pengaturan layout dari tiap HTML element akan menyesuaikan dengan viewport yang digunakan. Responsive web design akan membuat halaman web menjadi lebih rapi di semua ukuran layar device. Secara umum viewport merupakan daerah pada layar yang menampilkan suatu konten.

Tailwind CSS dan Chakra UI merupakan sebuah framework CSS yang dapat memudahkan developer dalam styling sebuah halaman web. Chakra UI merupakan sebuah component library yang akan menyediakan seperangkat komponen yang sederhana dan mudah diakses untuk membangun sebuah aplikasi React.

Menu digital pada umumnya diakses melalui smartphone maupun tablet. User yang pada umumnya menggunakan beberapa device yang berbeda maka dalam pengembangan aplikasi menu digital menggunakan konsep responsive web design yang akan menyesuaikan layout pada smartphone dan tablet.

Untuk menggunakan Chakra UI maka aplikasi yang dikembangkan harus dibungkus dengan komponen ChakraProvider untuk dapat berfungsi dalam aplikasi React.

```

1.   const root = ReactDOM.createRoot(document.getElementById('root'));
2.   root.render(
3.     <React.StrictMode>
4.     <ChakraProvider>
5.       <BrowserRouter>
6.         <App />
7.       </BrowserRouter>
8.     </ChakraProvider>
9.   </React.StrictMode>
10.  );

```

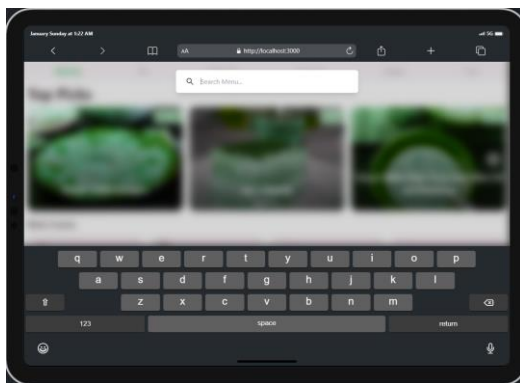
Kode di atas merupakan tampilan aplikasi *web* menu digital dibungkus oleh komponen ChakraProvider agar Charka UI dapat digunakan.

```

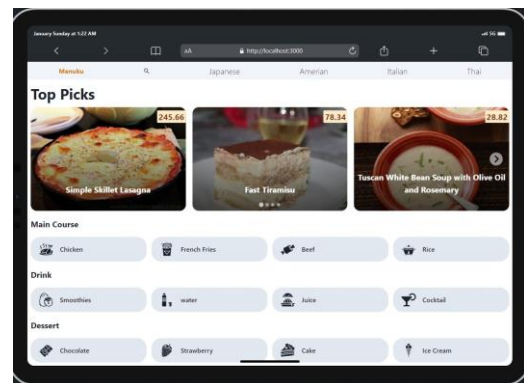
1.   <SimpleGrid columns={{[1, 2]} spacing={8} m="10px">
2.     {data.map((el) => {
3.       return (
4.         <CardMenu
5.           data={el}
6.           klik={() => {
7.             handleklik(el.id);
8.           }}
9.           country={menuNames}
10.          key={el.id}
11.        />
12.      );
13.    })}
14.  </SimpleGrid>

```

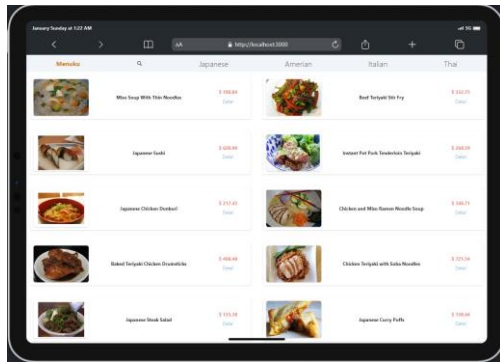
Kode di atas merupakan salah satu contoh penerapan konsep responsive web design menggunakan Chakra UI. Dikarenakan Chakra-UI menggunakan pendekatan mobile-first design, maka seorang developer akan membuat sebuah desain untuk tampilan mobile terlebih dahulu lalu selanjutnya akan disesuaikan untuk tampilan yang lebih besar pada tablet maupun desktop.



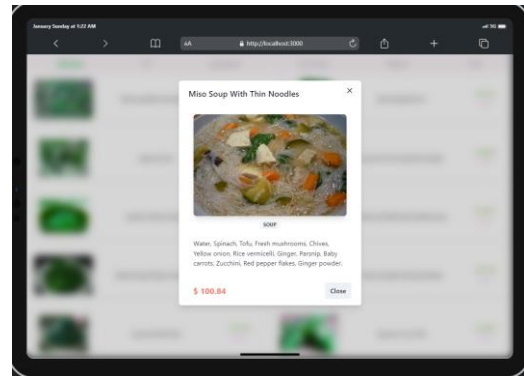
Gambar 5. Search pada Tablet



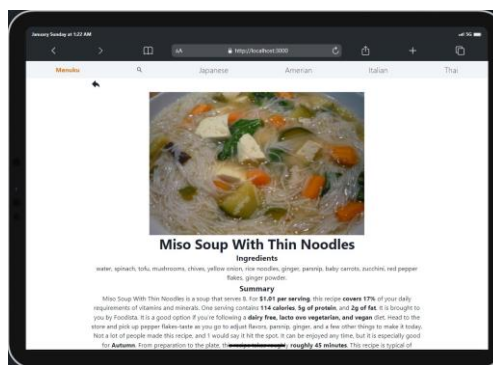
Gambar 6. Tampilan Menu Digital pada Tablet



Gambar 7. Tampilan Menu Khas Negara Jepang pada Tablet

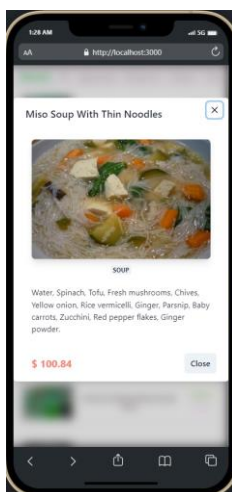


Gambar 8. Pop-Up Modal dalam Tablet

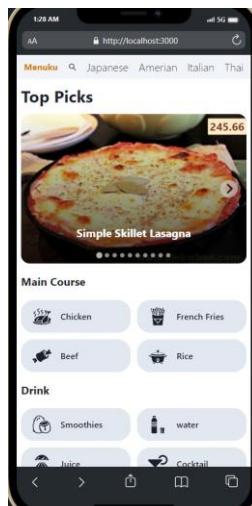


Gambar 9. Detail Menu dalam Tablet

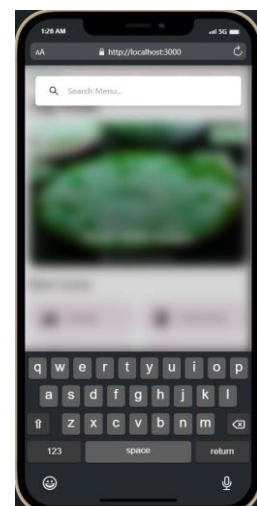
Gambar 5, gambar 6, gambar7, gambar 8 dan gambar 9 merupakan tampilan menu digital pada tablet. Dalam halaman home tersebut menampilkan Top Picks dari menu yang tersedia dalam bentuk carousel. Dalam tampilan tablet untuk menu Top Picks ditampilkan menjadi tiga column serta untuk main course, drink maupun dessert ditampilkan menjadi empat column.



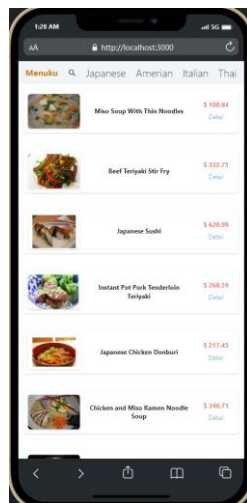
Gambar 10.Pop-Up Menu dalam Mobile



Gambar 11. Tampilan Menu Dalam Mobile



Gambar 12. Tampilan Search dalam Mobile



Gambar 13. Tampilan Menu Khas Negara Jepang dalam Mobile

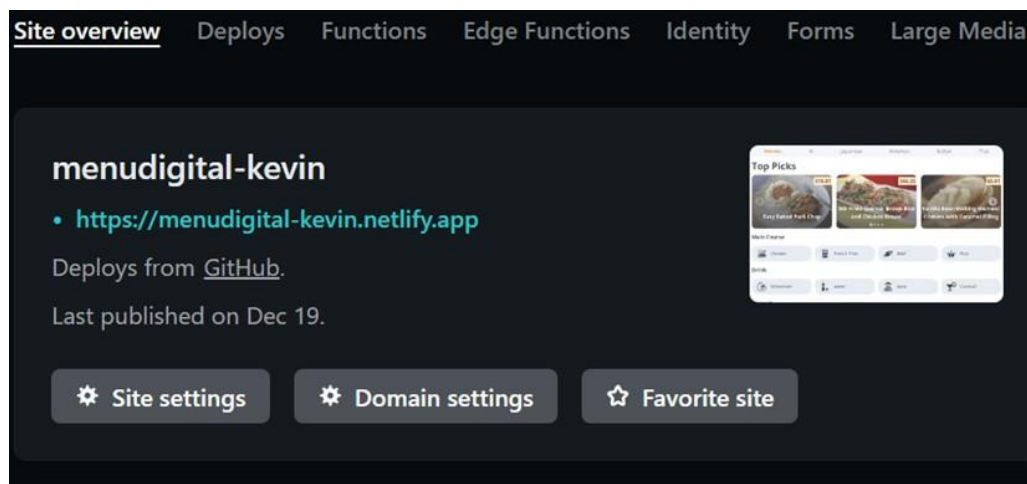


Gambar 14. Detail Menu dalam Mobile

Pada gambar 10, gambar 11, gambar 12, gambar 13 dan gambar 14 menampilkan tampilan pada menu digital yang memiliki layout yang menyesuaikan dengan device mobile. Pada device mobile tampilan menu digital dibuat dengan layout grid satu column agar user dapat melihat menu dengan lebih jelas.

H. Deployment to Netlify

Hasil pengembangan aplikasi web menu digital selanjutnya di-deploy melalui layanan unggah Netlify. Netlify sendiri merupakan sebuah layanan hosting dan pengembangan static website yang menyediakan layanan continuous deployment yang memungkinkan untuk mengunggah perubahan website secara otomatis setiap kali ada perubahan kode di git repository.



Gambar 15. Tampilan Website yang Sudah Diunggah di Netlify

Gambar 15 menunjukkan sebuah tampilan hasil deployment di Netlify. Dengan men-deploy menu digital dapat digunakan dengan mudah oleh siapapun.

V. SIMPULAN DAN SARAN

Kesimpulan yang diperoleh dari Pengembangan Menu Digital Menggunakan ReactJs Implementasi Hasil Belajar Studi Independen di Frontend Engineering Program Ruangguru CAMP (Career Acceleration Bootcamp) adalah dengan menggunakan React Js dan React Router DOM serta dengan bantuan framework CSS Chakra UI dapat membuat sebuah aplikasi yang responsif dan efisien dalam pengembangan menu digital. Adapun materi yang diberikan sudah cukup untuk mengenal lebih dalam menjadi seorang Frontend Engineer, materi yang diberikan juga menggunakan teknologi yang banyak digunakan oleh Frontend Engineer di masa sekarang.

Berdasarkan hasil studi independen di Ruangguru CAMP ini, dapat diperoleh saran sebagai berikut:

1. Metode belajar synchronous mungkin dapat ditingkatkan rentang waktu belajar dalam seminggu untuk memastikan mentee dapat mengerti secara keseluruhan materi.
2. Mengadakan proyek akhir dengan kolaborasi dari Frontend dan Backend dari mentee Ruangguru CAMP.

DAFTAR PUSTAKA

- [1] M. Y.-C. Yim dan C. Y. Yoo, "Are Digital Menus Really Better than Traditional Menus? The Mediating Role of Consumption Vision and Menu Enjoyment," *Journal of Interactive Marketing*.
- [2] A. Mesbah dan A. v. Deursen, "Migrating multi-page web applications to single-page ajax interface.," dalam *In Software Maintenance and Reengineering, 11th European Conference, 2007*, pp. 181-190.
- [3] Geek for Geek, "ReactJS | Router," 22 Aug 2022. [Online]. Available: <https://www.geeksforgeeks.org/reactjs-router/>. [Diakses 2022 Des 17].
- [4] PT RUANG RAYA INDONESIA, "Tentang Ruangguru," [Online]. Available: <https://www.ruangguru.com/about-us>. [Diakses 17 Des 2022].
- [5] J. N. Robbins, *Learning Web Design "A Beginners Guide to HTML, CSS, JAVASCRIPT and Web Graphics"*, Canada: O'Reilly Media, Inc, 2012.
- [6] D. S. McFarland, *Javascript & jQuery: The Missing Manual, Second Edition*, United States of America: O'Reilly Media, Inc., 2012.
- [7] "Netscape and Sun announce JavaScript," PR Newswire, 4 December 1995. [Online]. Available: <https://web.archive.org/web/20070916144913/http://wp.netscape.com/newsref/pr/newsrelease67.html>. [Diakses 17 December 2022].
- [8] MDN Web Docs, "Javascript," MDN Contributors, 14 December 2022. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. [Diakses 17 December 2022].
- [9] K. Simpson, *You Don't Know JS: ES6 & Beyond*, United States of America: O'Reilly Media, Inc., 2016.
- [10] techopedia, "Cross Platform," Techopedia Inc, 17 August 2017. [Online]. Available: <https://www.techopedia.com/definition/17056/cross-platform>. [Diakses 17 December 2022].
- [11] Node JS , "The V8 JavaScript Engine," Node JS, [Online]. Available: <https://nodejs.dev/en/learn/the-v8-javascript-engine/>. [Diakses 17 December 2022].
- [12] NPMJS, "About NPM," npm docs, [Online]. Available: <https://docs.npmjs.com/about-npm>. [Diakses 17 December 2022].
- [13] W3Schools, "HTML Responsive Web Design," [Online]. Available: https://www.w3schools.com/html/html_responsive.asp. [Diakses 10 01 2023].
- [14] T. Khuat, *Developing A Frontend Application using ReactJS and Redux*, Laura University of Applied Sciences, 2018.