



Mata Kuliah : Pemograman Web Lanjut
Materi : React JS
Kelas : TI-19-KA
Pengajar : Febri Damatraseta S.T, M.Kom

1. Tools yang diperlukan

Download dan install Node JS [<https://nodejs.org/en/download/>]

IDE dapat menggunakan Notepad++, Sublime Text, atau VS Code

Compiler dapat menggunakan Command Prompt atau Terminal

2. Cara membuat project

Setelah berhasil menginstal Node JS, buka command prompt atau terminal dan pilih directory path dimana anda akan menyimpan atau membuat project react. Setelah menetapkan directory tuliskan sintax dibawah ini untuk membuat project dengan React JS:

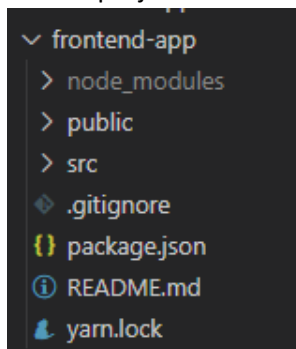
```
npx create-react-app my-app
```

my-app merupakan nama project yang akan kalian buat, dapat kalian ganti dengan nama apapun. Namun ketentuan penamaan project perlu diperhatikan seperti:

1. Tidak boleh menggunakan Space () sebagai pemisah antar kalimat atau penggunaan dua frasa kata
2. Gunakan tanda strip (-) atau garis bawah (_) atau menggunakan argument penggabungan text dari beberapa rentang atau kata.

Setelah berhasil membuat project React JS, maka kalian akan mendapatkan folder project sesuai dengan penamaan yang telah dibuat dan disimpan didalam directory path yang telah ditentukan.

Struktur project dari React JS terdiri dari beberapa bagian seperti gambar dibawah ini:



- **Frontend-app** merupakan nama project yang telah dibuat
- **Node_modules**, folder ini berisi module-module utama dari Node.Js dan react.js. yang sudah datang ketika kamu melakukan penginstalan react.js. Dalam folder node_modules berisi repositori modul/library yang akan digunakan di dalam project. Apa pun yang akan diimpor kedalam project, library itu harus ada di dalam folder node_module. Proses penambahan module library biasanya dilakukan ketika kita memasukan sintax `npm install nama_module` didalam terminal project. Module yang akan ditambahkan kedalam project akan tercatat di file `package.json` sebagai bagian dari dependencies.
- **Public**, folder ini biasanya berisi data tambahan yang diperlukan pada project. Bagian folder ini umumnya digunakan sebagai data penyimpanan untuk template atau bentuk dari website yang akan dibuat. Data-data yang disimpan disini umumnya bersifat tetap atau tidak dapat diubah-ubah.
- **Src**, folder ini berisi data-data proses developing yang akan diolah sebagai proses input-output. Didalam struktur folder ini seluruh data yang disimpan dapat diubah-ubah sesuai dengan business logic yang dibutuhkan. Pada dasarnya saat pertama kali program react dijalankan, react akan memanggil file index.html yang berada dalam folder public. Selanjutnya defaultnya react akan merender file app dan di tampilkan di dalam div dengan id root di dalam file index.html.

3. Cara menjalankan project

Untuk menjalankan project react js diperlukan masuk kedalam directory path project dan dieksekusi dengan menggunakan terminal. Dan memberikan argument seperti dibawah ini untuk dapat menjalankan project react:

```
npm start
```

Berikut ini adalah contoh cara menjalankan project react:

```
Windows PowerShell
C:\xampp\htdocs\program\IBIK\2020-2021\Pemograman-Web-Lanjut\frontend-app>npm start
> frontend-app@0.1.0 start C:\xampp\htdocs\program\IBIK\2020-2021\Pemograman-Web-Lanjut\frontend-app
> react-scripts start

i @wdwds: Project is running at http://192.168.8.134/
i @wdwds: webpack output is served from
i @wdwds: Content not from webpack is served from C:\xampp\htdocs\program\IBIK\2020-2021\Pemograman-Web-Lanjut\frontend-a
pp\public
i @wdwds: 404s will fallback to /
Starting the development server...

Browserslist: caniuse-lite is outdated. Please run:
npx browserslist@latest --update-db

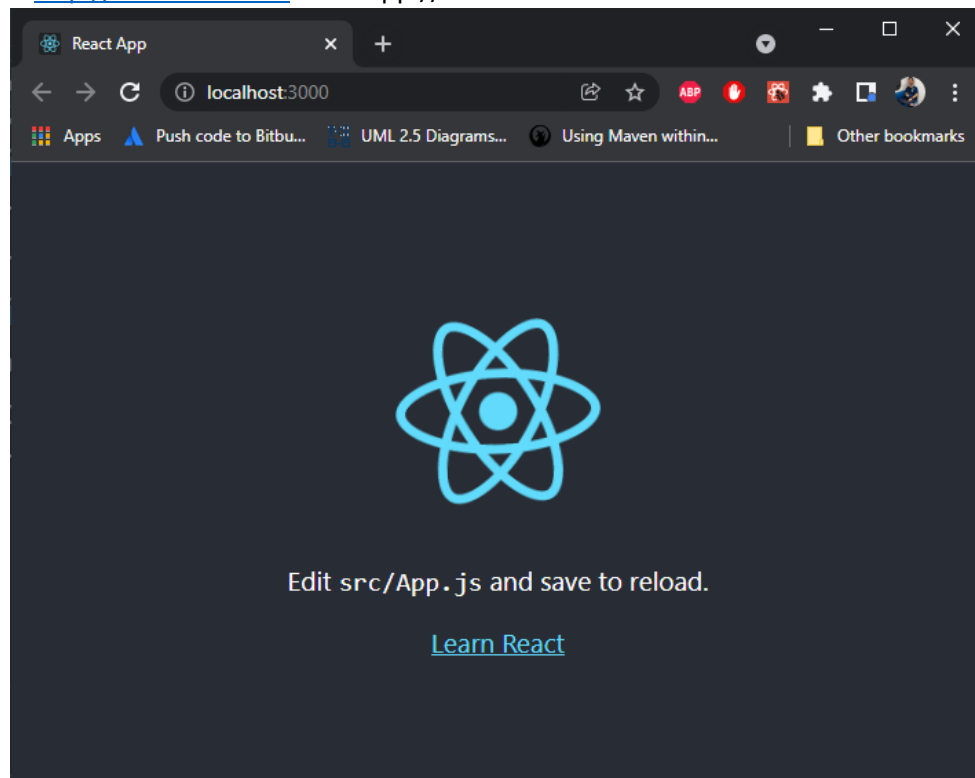
Why you should do it regularly:
https://github.com/browserslist/browserslist#browsers-data-updating
Compiled successfully!

You can now view frontend-app in the browser.

Local:      http://localhost:3000
On Your Network:  http://192.168.8.134:3000

Note that the development build is not optimized.
To create a production build, use yarn build.
```

Pada gambar diatas, setelah proses compiling berhasil maka kita dapat mengakses project react dengan cara membuka browser dan memasukan address local network yang telah ditentukan. Contoh pada gambar diatas, untuk mengakses address project react dengan cara mengakses local network di <http://localhost:3000> atau <http://192.168.8.134:3000>



4. Penggunaan React Class Component (RCC)

```
class Welcome extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>;  
  }  
}
```

Komponen RCC memungkinkan untuk membagi UI menjadi bagian-bagian independen yang dapat digunakan kembali, dan memikirkan setiap bagian secara terpisah. Halaman ini memberikan pengenalan ide komponen.

Dalam mendeklarasikan sebuah variable dengan RCC dapat menggunakan teknik **Component State**.

```
this.setState({name:"Febri"})
```

5. Penggunaan React Function Component (RFC)

```
import React from 'react';  
  
function App() {  
  const greeting = 'Hello Function Component!';  
  
  return <h1>{greeting}</h1>;  
}  
  
export default App;
```

RFC adalah status *quo* dari penulisan aplikasi React modern. Di masa lalu, ada berbagai Jenis Komponen React, tetapi dengan diperkenalkannya React Hooks, Anda dapat menulis seluruh aplikasi Anda hanya dengan fungsi sebagai komponen React.

Komponen RFC yang pada dasarnya hanyalah Fungsi JavaScript sebagai Komponen React yang mengembalikan JSX (Sintaks React).

Untuk mendeklarasikan variable dengan RFC dapat menggunakan teknik **Component Hooks**.

```
const [name, setName] = useState("Febri");
```

6. JSX

```
const element = <h1>Hello world!</h1>;
```

Sintaksis tag aneh ini bukanlah sebuah string ataupun HTML.

Sintaksis ini di kenal dengan sebutan JSX, dan sintaksis ini adalah sebuah sintaksis ekstensi untuk JavaScript. Digunakan sintaks ini dengan React untuk mendeskripsikan bagaimana antarmuka pengguna seharusnya terlihat. JSX mungkin akan mengingatkan dengan sebuah bahasa templat, bedanya adalah JSX telah dilengkapi dengan kekuatan penuh dari JavaScript.

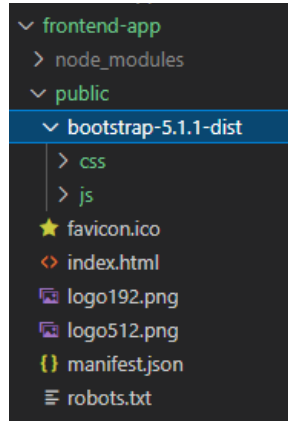
JSX akan menghasilkan “elemen” React. “Element” akan mengeksplor bagaimana me-render mereka ke dalam DOM di bagian berikutnya.

Dokumentasi selengkapnya bisa dilihat di <https://reactjs.org/docs/getting-started.html>

Latihan:

✚ Memasang template pada project react js.

- Pada latihan kali ini hal yang perlu disiapkan ialah source dari bootstrap. Dan masukan source bootstrap kedalam struktur folder public.



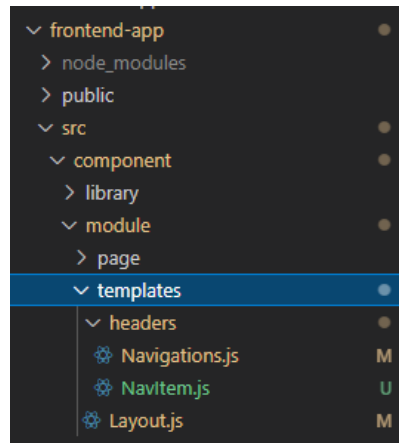
- Buka file index.html pada folder public, dan silakan kalian masukan file css dan js di file tersebut. Dan silakan membuat template desain website yang diinginkan.

```
index.html M X
frontend-app > public > index.html > html > body.d-flex.flex-column.h-100
14  manifest.json provides metadata used when your web app is installed on a
15  user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
16  -->
17  <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
18  <!--
19  Notice the use of %PUBLIC_URL% in the tags above.
20  It will be replaced with the URL of the 'public' folder during the build.
21  Only files inside the 'public' folder can be referenced from the HTML.
22
23  Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
24  work correctly both with client-side routing and a non-root public URL.
25  Learn how to configure a non-root public URL by running 'npm run build'.
26  -->
27  <title>Pemograman Web Lanjut</title>
28
29  <link href="%PUBLIC_URL%/bootstrap-5.1.1-dist/css/bootstrap.min.css" rel="stylesheet" type="text/css" />
30  <script src="%PUBLIC_URL%/bootstrap-5.1.1-dist/js/bootstrap.bundle.min.js"></script>
31
32  </head>
33
34  <body class="d-flex flex-column h-100">
35    <noscript>You need to enable JavaScript to run this app.</noscript>
36    <div id="root"></div>
37  <!--
38  This HTML file is a template.
```

Untuk memanggil file yang ada di dalam folder public, diperlukan syntax argument `'%PUBLIC_URL%/...'` dilanjutkan dengan path dari file yang akan digunakan.

- Setelah menambahkan data-data dan memodifikasi index.html di folder public, selanjutnya kita akan membuat sebuah process developing di folder src. Disini akan dicontohkan bagaimana membuat sebuah layout seperti ini <https://getbootstrap.com/docs/5.1/examples/sticky-footer-navbar/>

- Buatlah file bernama Layout.js, headers/Navigations.js dan NavItem.js di path src seperti gambar dibawah ini:



Pada file NavItem.js masukan scripting dibawah ini:

```
import React from 'react'

const NavItem = (props) =>{
  return (
    <ul className="navbar-nav me-auto mb-2 mb-md-0">
      {(props.data) ? (
        props.data.map((v)=>{
          <li className="nav-item px-1" title={v.name}>
            <a className="nav-link fs-4" href={v.url}>
              <i className={v.icon}></i>
              <span className="d-sm-none">{v.name}</span>
            </a>
          </li>
        })
      ) : ''}
    </ul>
  )
}

export default NavItem
```

Pada file Navigations.js masukan scripting dibawah ini:

```
import React, {useState} from 'react'
import NavItem from './NavItem';

export function Navigations() {
  const [menu, setMenu] = useState([
    {name: "Notifications", icon: "fa-globe", url: "/notifications"},
    {name: "Messages", icon: "fa-envelope-o", url: "/messages"},
    {name: "Profile", icon: "fa-user-o", url: "/profile"},
  ]);

  return (
    <nav className="navbar navbar-expand-md navbar-dark fixed-top" style={{backgroundColor: "#7952b3"}}>
      <div className="container-fluid">
        <a className="navbar-brand" href="#">IBIK</a>
        <button className="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse" aria-controls="navbarCollapse" aria-expanded="false" aria-label="Toggle navigation">
          <span className="navbar-toggler-icon"></span>
        </button>
        <div className="collapse navbar-collapse" id="navbarCollapse">
          <form className="d-flex w-100 px-5">
            <div className="input-group">
              <input className="form-control" type="search" placeholder="Search" aria-label="Search" />
              <span class="input-group-text" id="basic-addon1"><i className="fa fa-search"></i></span>
            </div>
          </form>
          <NavItem data={menu} />
        </div>
      </div>
    </nav>
  )
}

export default Navigations;
```

Pada file Layout.js masukan scripting dibawah ini:

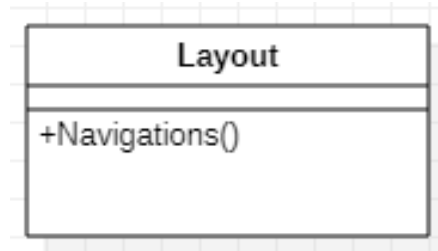
```
import React, { Component } from 'react'
import Navigations from "../headers/Navigations";

export class Layout extends Component {
  render() {
    return (
      <div>
        <header>
          <Navigations />
        </header>

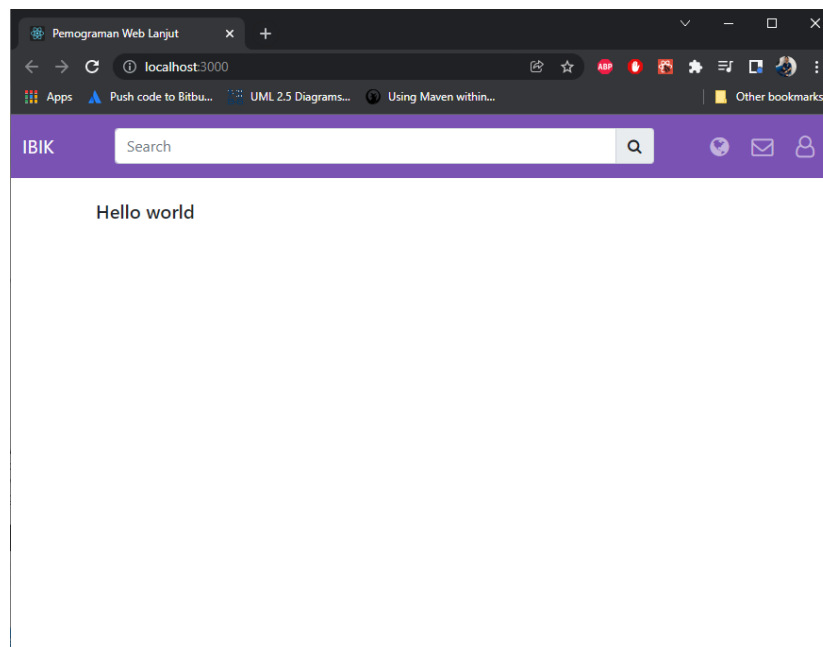
        <main className="flex-shrink-0" style={{marginTop:"80px",paddingTop:"10px"}}>
          <div className="container">
            <h5 className="">Hello world</h5>
          </div>
        </main>
      </div>
    )
  }
}

export default Layout;
```

Dari bentuk scripting diatas telah terbentuk sebuah class diagram seperti dibawah ini:



Dimana class Layout nantinya akan terus berkembang dengan memiliki banyak Attribute dan Operation sesuai dengan kebutuhan module yang akan dibuat. Dan berikut adalah hasil output dari scripting yang telah dibuat.



Pertanyaan:

1. Apa perbedaan antara *React Class Component* dan *React Function Component* ?
2. Pada file `Layout.js` terdapat sebuah syntax JSX dengan tag `<Navigations />`. Jelaskan fungsi penggunaan tag `<Navigations />` pada file tersebut. Dan bagaimana cara kerjanya sehingga tag tersebut dapat berjalan.
3. Pada file `Navigations.js` mengirimkan sebuah nilai array dalam bentuk object yang disimpan pada variabel `menu`. Jelaskan teknik apa yang digunakan dalam file ini untuk mengirimkan variabel `menu` kedalam tag `<NavItem />`.
4. Pada file `NavItem.js`, apakah teknik scripting yang digunakan dapat diubah seperti teknik scripting `Navigation.js` ? Berikan contohnya dalam bentuk implementasi code.
5. Pada scripting dibawah ini khususnya untuk baris code ke-6 hingga ke-15, structure programming structure programming yang digunakan ada dua, pertama Selection dan Repetition. Jelaskan bagian mana yang termasuk kedalam Selection dan Repetition pada script dibawah ini.

```
1 import React from 'react'
2
3 const NavItem = (props) =>{
4   return (
5     <ul className="navbar-nav me-auto mb-2 mb-md-0">
6       {(props.data) ? (
7         props.data.map((v)=>{
8           <li className="nav-item px-1" title={v.name}>
9             <a className="nav-link fs-4" href={v.url}>
10               <i className={`fa "+v.icon}></i>
11               <span className="d-sm-none">{v.name}</span>
12             </a>
13           </li>
14         ))
15       ) : ''}
16     </ul>
17   )
18 }
19
20 export default NavItem
```