

VERSION 4.0

JULY 1, 2023



# PEMROGRAMAN WEB

REACT FUNDAMENTAL – MODUL 3

DISUSUN OLEH:  
SYAUQI AMIQ AMRULLAH  
ALIF FATWA RAMADHANI

DIAUDIT OLEH:  
AMINUDIN, S.KOM., M.CS

LAB. INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH MALANG

## PEMROGRAMAN WEB

---

### PERSIAPAN MATERI

- <https://legacy.reactjs.org/docs/getting-started.html>
- <https://www.w3schools.com/REACT/DEFAULT.ASP>

---

### TUJUAN

Mahasiswa mampu memahami konsep dasar dari pemrograman website

---

### TARGET MODUL

1. Mahasiswa dapat memahami cara membuat desain web
2. Mahasiswa mampu mengimplementasikan desain dengan code

---

### PERSIAPAN SOFTWARE/APLIKASI

Hardware dan Infrastruktur

- Laptop/PC.
- Koneksi Internet.

Software

- Text Editor: Visual Studio Code (Recommended).

Extension (Visual Studio Code)

- Live Server | Ritwick dey.
- Prettier - Code Formatter | Prettier.
- HTML CSS Support | Ecmel.
- ES7 React/Redux/GraphQL/React-Native snippets | rodrigovallades.

Programming Language

- Javascript, HTML, CSS.

Installation

- Node.js
- React.js (Framework [CRA])

---

## MATERI

### Pengertian React.js

#### A. Apa itu React.js?

React.js adalah sebuah library JavaScript yang digunakan untuk membangun antarmuka pengguna (UI) dalam pengembangan website. React dikembangkan oleh Facebook dan dirilis sebagai open-source pada tahun 2013.

#### B. Karakteristik utama React.js

##### 1. Komponen

React memungkinkan pengembangan menggunakan pendekatan berbasis komponen. Aplikasi dibagi menjadi komponen-komponen yang dapat digunakan Kembali.

##### 2. Virtual DOM

React menggunakan konsep Virtual DOM, yang merupakan representasi virtual dari struktur DOM actual yang dimana ini membantu mengoptimalkan kinerja, karena react dapat membandingkan perubahan di virtual DOM dan memperbarui hanya bagian-bagian yang diperlukan DOM actual untuk menghindari pembaruan yang tidak perlu.

##### 3. State dan Props

React memungkinkan penggunaan state dan props untuk mengelola data dan mengalirkan data antara komponen. State adalah data yang dimiliki oleh komponen itu sendiri, sementara props adalah data yang diteruskan dari komponen induk ke komponen anak.

##### 4. React Router

React router adalah library tambahan yang digunakan untuk mengelola navigasi dan rute dalam aplikasi react, memungkinkan pengembang untuk membuat aplikasi multipage dengan navigasi yang responsive

##### 5. Komersial dan Open-Source

React adalah proyek open-source yang dapat digunakan secara gratis oleh siapa saja, dan banyak perusahaan besar dan startup menggunakan React dalam proyek komersial karena keunggulan kinerjanya. Contoh perusahaan yang menggunakan react.js



traveloka

## Intro React.js

### A. Membuat aplikasi React menggunakan CRA (create-react-app)

#### 1. Install Node.js

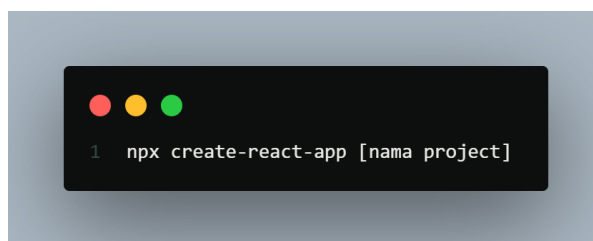
kunjungi halaman website <https://nodejs.org/en/download>, kemudian pilih versi 18.18.0 LTS (Recommended For Most Users) dan pilih sesuai dengan sitem operasi device masing masing.



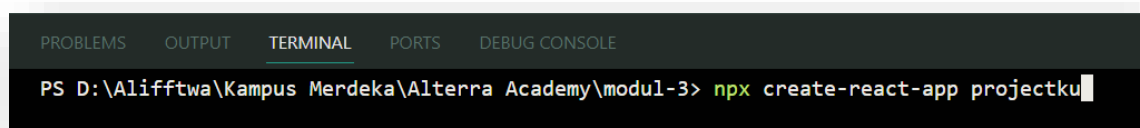
Youtube - how to install node.js:

<https://youtu.be/06X51c6WHsQ?si=9d26MuXfOYQ435d8>

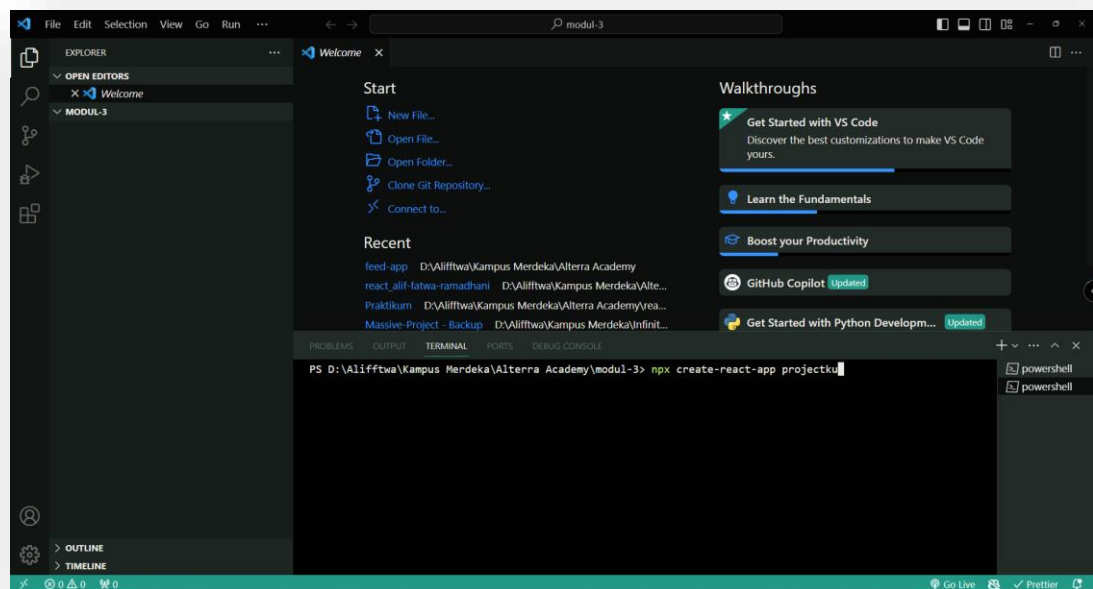
#### 2. Buat aplikasi react.js:



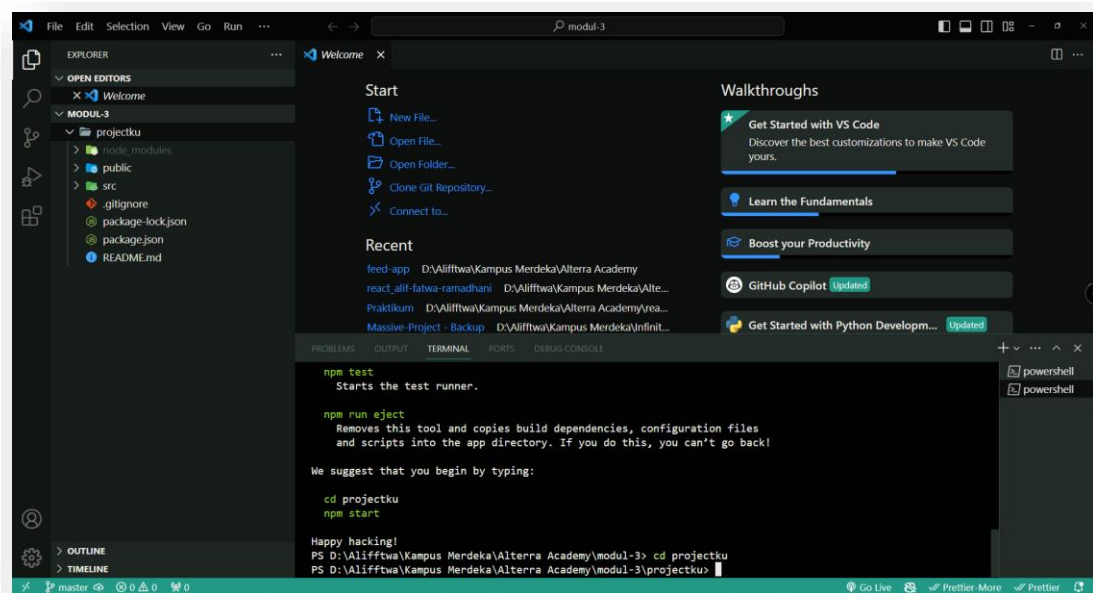
Contoh:



(Sebelum membuat app)

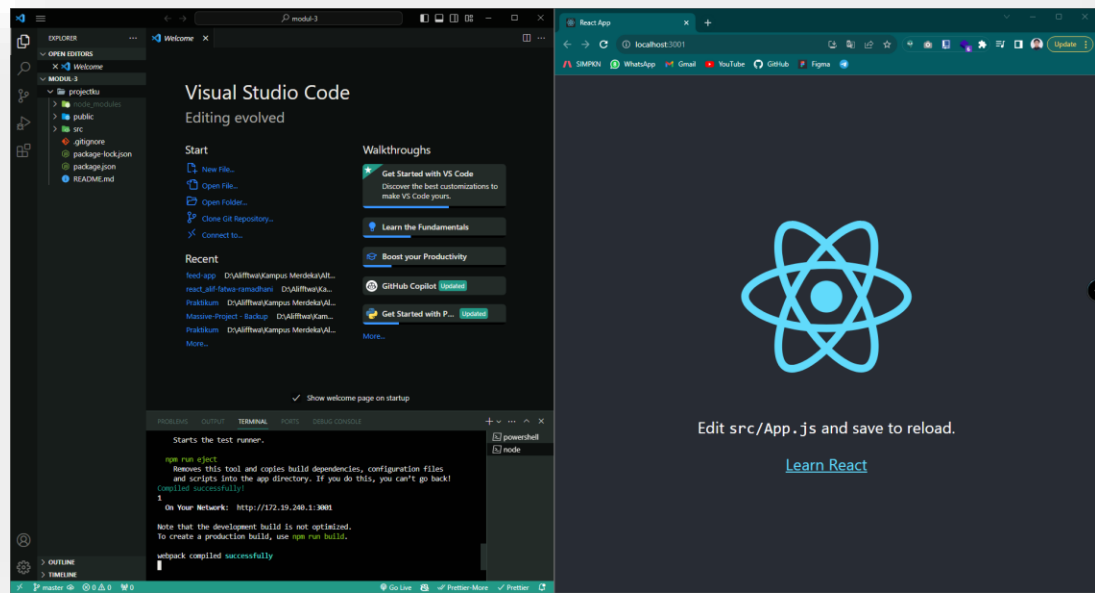


(Sesudah membuat app)



- **Choose Destination Folder**

Run app dengan menggunakan command **npm start** pada terminal



## Dasar React.js

### A. JavaScript XML (JSX)

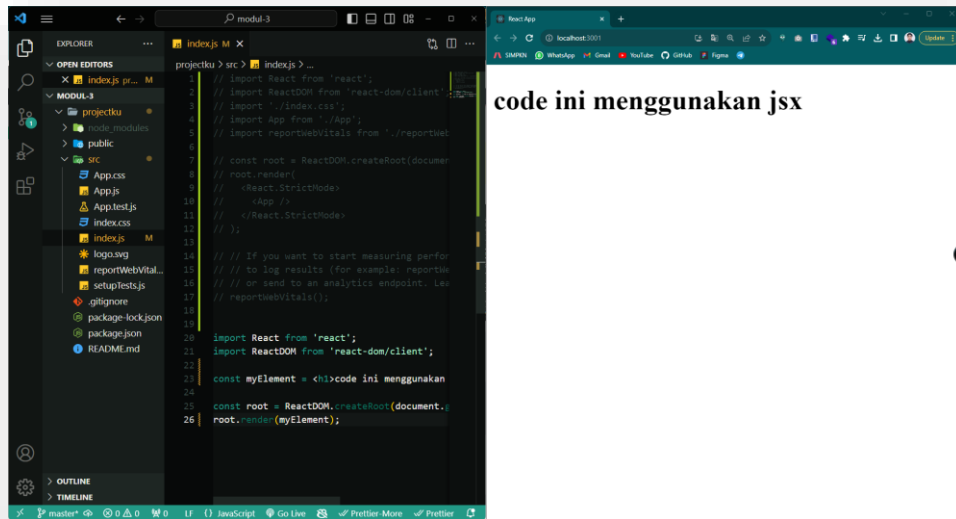
JSX atau JavaScript XML adalah ekstensi sintaksis yang digunakan dalam React, sebuah library JavaScript untuk membangun antarmuka pengguna (UI) dalam pengembangan web.

Contoh code yang menggunakan JSX:

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const myElement = <h1>code ini menggunakan jsx</h1>;

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```

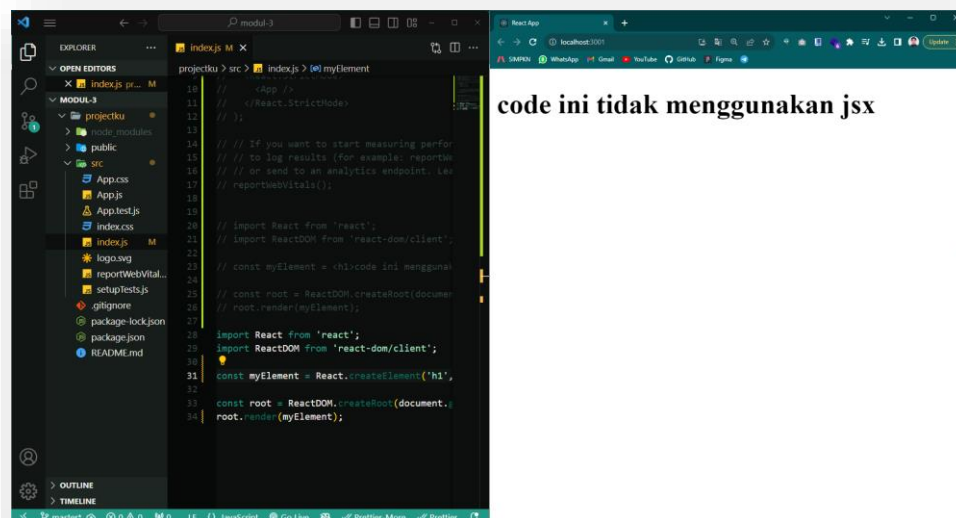


Contoh code yang tidak menggunakan JSX:

```
import React from 'react';
import ReactDOM from 'react-dom/client';

const myElement = React.createElement('h1', {}, 'code ini tidak menggunakan jsx');

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(myElement);
```



## B. Component

Komponen React adalah bagian code yang dapat digunakan Kembali yang digunakan untuk menentukan tampilan, behavior, dan state Sebagian UI.

```
function Car() {
  return <h2>Hi, I am a Car!</h2>;
}
```

Render Component:

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<Car />);
```

## C. Props

Dalam React, "props" adalah singkatan dari "properties" (properti). Props adalah mekanisme yang digunakan untuk mengirim data dari komponen induk (parent component) ke komponen anak (child component). Props adalah objek JavaScript yang berisi data atau nilai yang ingin Anda kirimkan dari satu komponen ke komponen lainnya. Contoh:

Child Component:

```
import React from 'react';

function ChildComponent(props) {
  return <p>Hello, {props.name}!</p>;
}

export default ChildComponent;
```

Parent Component:

```
import React from 'react';
import ChildComponent from './ChildComponent';

function ParentComponent() {
  const name = 'Oki';

  return (
    <div>
      <h1>Welcome to My App</h1>
      <ChildComponent name={name} />
    </div>
  );
}

export default ParentComponent;
```



*Child Component* akan menerima prop bernama *name* dari *Parent Component*. Prop ini digunakan untuk merender pesan "Hello, Oki!" di dalam *Child Component*.

#### D. Event

Dalam React, sebuah "event" adalah tindakan atau interaksi yang terjadi di dalam komponen, seperti mengklik tombol, mengisi formulir, atau menekan tombol keyboard. React memungkinkan Anda untuk menangani event ini dengan mudah menggunakan event handlers (penangan event) yang dikenal sebagai "event handlers" atau "event listeners."

Contoh event yang umum digunakan dalam React:

1. **onClick**, event yang terjadi ketika elemen di klik.

```
<button onClick={handleClick}>Klik Saya</button>
```

2. **onChange**, event yang sering digunakan untuk handle perubahan pada input elemen seperti input teks atau input checkbox.

```
<input type="text" onChange={handleChange} />
```

3. **onSubmit**, Event ini terjadi ketika pengguna mengirimkan formulir.

```
<form onSubmit={handleSubmit}>
  /* elemen-elemen formulir */
  <button type="submit">Kirim</button>
</form>
```

Untuk menangani event dalam react, anda mendefinisikan fungsi (event handler) yang akan dijalankan ketika event terjadi. Contoh:

```
function handleClick() {
  alert('Tombol telah diklik!');
}
```

## E. Condition

Condition atau pengkondisian mengacu pada penggunaan pernyataan logika untuk mengatur perilaku komponen berdasarkan kondisi tertentu. Kita dapat menggunakan berbagai teknik pengkondisian dalam React untuk mengubah tampilan atau perilaku komponen berdasarkan keadaan atau data tertentu.

1. **Penggunaan if-else dalam JSX:** Anda dapat menggunakan pernyataan if-else atau ternary operator dalam JSX untuk menggantikan elemen yang dirender berdasarkan kondisi tertentu.

```
function MyComponent({ isLoggedIn }) {
  return (
    <div>
      {isLoggedIn ? <p>Selamat datang, Pengguna!</p> : <p>Silakan
masuk terlebih dahulu.</p>}
    </div>
  );
}
```

2. **Menggunakan variabel atau state:** Anda dapat menggunakan variabel atau state dalam komponen React untuk mengontrol perilaku komponen.

```
function MyComponent() {
  const [showMessage, setShowMessage] = useState(false);

  const toggleMessage = () => {
    setShowMessage(!showMessage);
  };

  return (
    <div>
      <button onClick={toggleMessage}>Toggle Pesan</button>
      {showMessage && <p>Pesan ini akan ditampilkan jika tombol di
atas ditekan.</p>}
    </div>
  );
}
```

3. **Menggunakan pernyataan if di dalam komponen:** Anda juga dapat menggunakan pernyataan if di dalam komponen untuk mengendalikan logika dan perilaku komponen.

```
function MyComponent({ condition }) {
  if (condition) {
    return <p>Kondisi benar</p>;
  } else {
    return <p>Kondisi salah</p>;
  }
}
```

## F. Form

Membuat formulir (form) dalam React adalah salah satu tugas umum dalam pengembangan aplikasi web. Formulir digunakan untuk mengumpulkan data dari pengguna seperti input teks, checkbox, radio button, atau elemen-elemen lainnya.

1. **Impor React dan Component:** Pastikan Anda telah mengimpor React dan Component dari pustaka React.

```
import React, { useState } from 'react';
```

2. **Membuat Komponen Form:** Buat sebuah komponen yang akan menjadi formulir Anda. Ini bisa menjadi komponen yang terpisah atau bagian dari komponen yang lebih besar.

```
import React, { useState } from 'react';

function MyForm() {
  const [formData, setFormData] = useState({
    name: "",
    email: "",
  });

  const handleInputChange = (event) => {
    const { name, value } = event.target;
    setFormData({ ...formData, [name]: value });
  };

  const handleSubmit = (event) => {
    event.preventDefault();
    // Lakukan sesuatu dengan data formulir, seperti mengirimkannya ke server
  };
}
```

```
return (  
  <form onSubmit={handleSubmit}>  
    <div>  
      <label htmlFor="name">Nama:</label>  
      <input  
        type="text"  
        id="name"  
        name="name"  
        value={formData.name}  
        onChange={handleInputChange}  
      />  
    </div>  
    <div>  
      <label htmlFor="email">Email:</label>  
      <input  
        type="email"  
        id="email"  
        name="email"  
        value={formData.email}  
        onChange={handleInputChange}  
      />  
    </div>  
    <button type="submit">Kirim</button>  
  </form>  
}  
  
export default MyForm;
```

## G. Router

React Router adalah sebuah pustaka (library) JavaScript yang digunakan untuk menangani navigasi dalam aplikasi web berbasis React. Dengan React Router, Anda dapat membuat aplikasi web yang memiliki banyak halaman atau tampilan yang berbeda dan mengelola navigasi antara halaman-halaman tersebut dengan mudah.

Install react-router-dom:

```
npm i react-router-dom
```

Berikut adalah beberapa komponen utama dari React Router:

1. **BrowserRouter:** Ini adalah komponen yang digunakan untuk membungkus aplikasi React Anda. Ini menggunakan HTML5 History API untuk menangani perubahan URL di browser.
2. **Route:** Komponen ini digunakan untuk mendefinisikan rute atau halaman tertentu dalam aplikasi Anda. Setiap Route mungkin atau mungkin tidak memiliki path (alamat URL) yang sesuai dan komponen yang akan dirender jika path tersebut cocok.
3. **Link:** Ini adalah komponen yang digunakan untuk membuat tautan atau link di aplikasi Anda. Ketika tautan ini diklik, aplikasi akan menavigasi ke rute yang sesuai.
4. **Switch:** Komponen ini digunakan untuk mengelompokkan beberapa Route bersama-sama. Ketika aplikasi Anda mencoba mencocokkan path URL dengan Route, ia akan mencari Route pertama yang cocok dalam Switch dan hanya merender komponen dari rute itu.
5. **Redirect:** Komponen ini digunakan untuk mengalihkan pengguna dari satu halaman ke halaman lain.

Contoh React Router:

```
import React from 'react';
import { BrowserRouter as Router, Route, Link, Switch } from 'react-router-dom';

const Home = () => <h1>Halaman Utama</h1>;
const About = () => <h1>Tentang Kami</h1>;
const Contact = () => <h1>Hubungi Kami</h1>;

function App() {
  return (
    <Router>
      <nav>
        <ul>
          <li>
            <Link to="/">Beranda</Link>
          </li>
          <li>
            <Link to="/about">Tentang</Link>
          </li>
          <li>
            <Link to="/contact">Kontak</Link>
          </li>
        </ul>
      </nav>
      <Switch>
        <Route path="/about" component={About} />
        <Route path="/contact" component={Contact} />
        <Route path="/" component={Home} />
      </Switch>
    </Router>
  );
}

export default App;
```

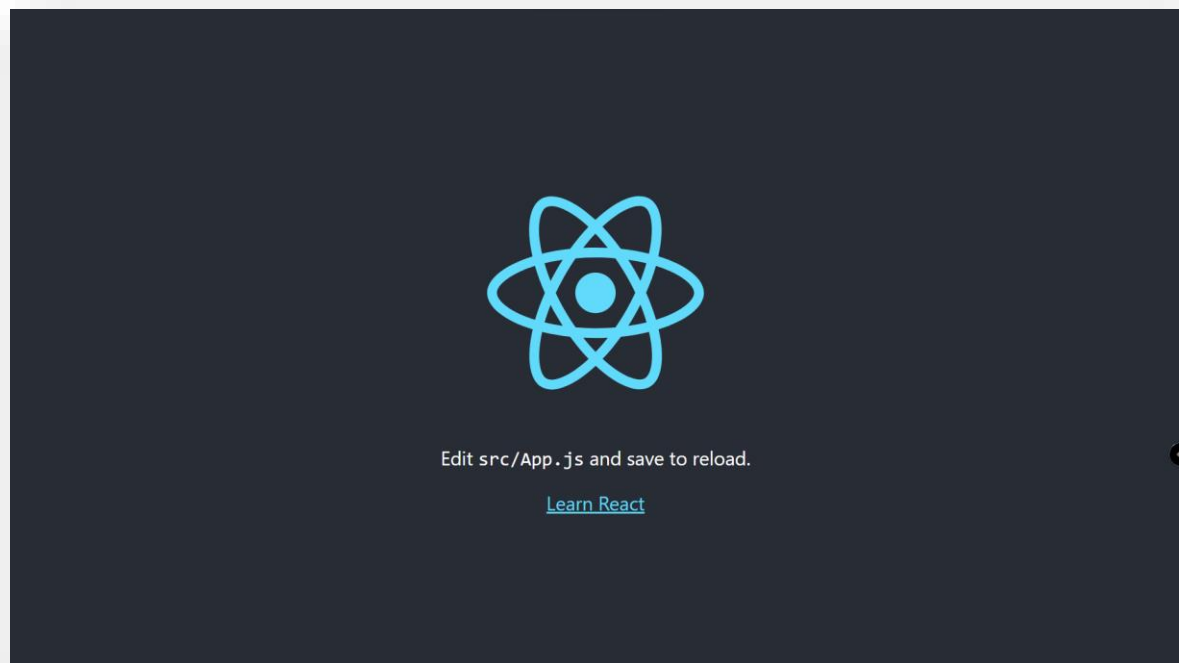
React Router digunakan untuk membuat aplikasi dengan tiga halaman (Beranda, Tentang, dan Kontak) yang dapat diakses melalui tautan. React Router memungkinkan Anda untuk memisahkan tampilan menjadi komponen-komponen yang berbeda dan mengelola navigasi antara halaman-halaman tersebut dengan nyaman.

---

## CODELAB

### TUGAS 1

Buatlah new project react.js dan lakukan *run build* hingga menampilkan tampilan halaman seperti gambar dibawah. **(Khusus tugas 1 dilakukan secara live demo)**



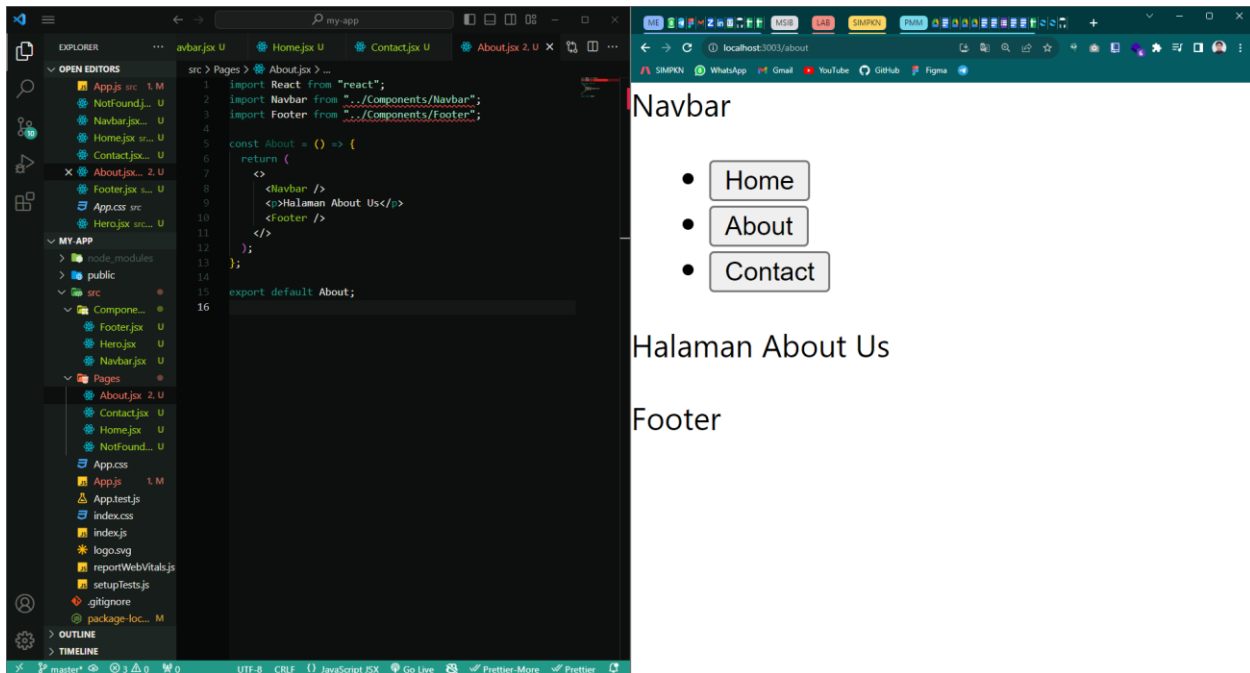
### TUGAS 2

Buatlah rangkaian website sederhana dengan ketentuan seperti dibawah

- Membuat components: Navbar (Home, About us, Contact), Hero, Footer
- Membuat pages: Menu, About us, Contact

Lakukan routing pada setiap opsi yang terdapat pada navbar, keberhasilan routing dapat dilihat dari perubahan pada subdomain.

## Contoh tugas 2:





---

## TUGAS PRAKTIKUM

Ubahlah tugas praktikum modul 1 (slicing homepage) menggunakan React.js dengan ketentuan sebagai berikut.

- Components: Navbar, Hero, Footer
- Pages: Homepage, About, Contact
- Buatlah tambahan pages/halaman untuk about dan contact (konten atau isi halaman **bebas tidak ada** ketentuan khusus)
- Lakukan routing pada setiap pages/halaman melalui navbar ke semua pages/halaman

---

## KRITERIA & DETAIL PENILAIAN

Kriteria	Persentase Penilaian
Dapat membuat dan menjelaskan program yang telah dibuat	40%
Dapat menjelaskan & mengimplementasikan Materi Modul	50%
Dapat menjawab pertanyaan dari asisten laboratorium	10%