**LAPORAN PRAKTIKUM**

**PEMROGRAMAN BERBASIS FRAMEWORK**

**Modul 3 : React Component**



**Oleh :**

**Nama : Hanin Syafiq R**

**Nim : 1841720183**

**Kelas : TI-3E**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**KOTA MALANG**

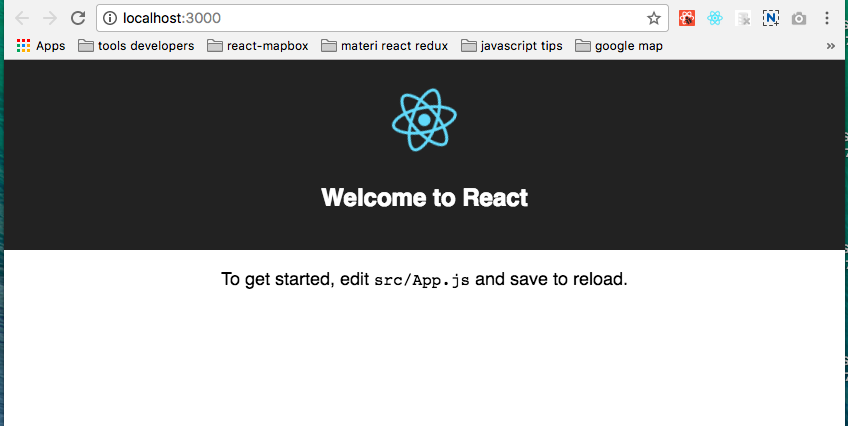
**2021**

1. **PRAKTIKUM**
2. Membuat Stateless component & statefull component

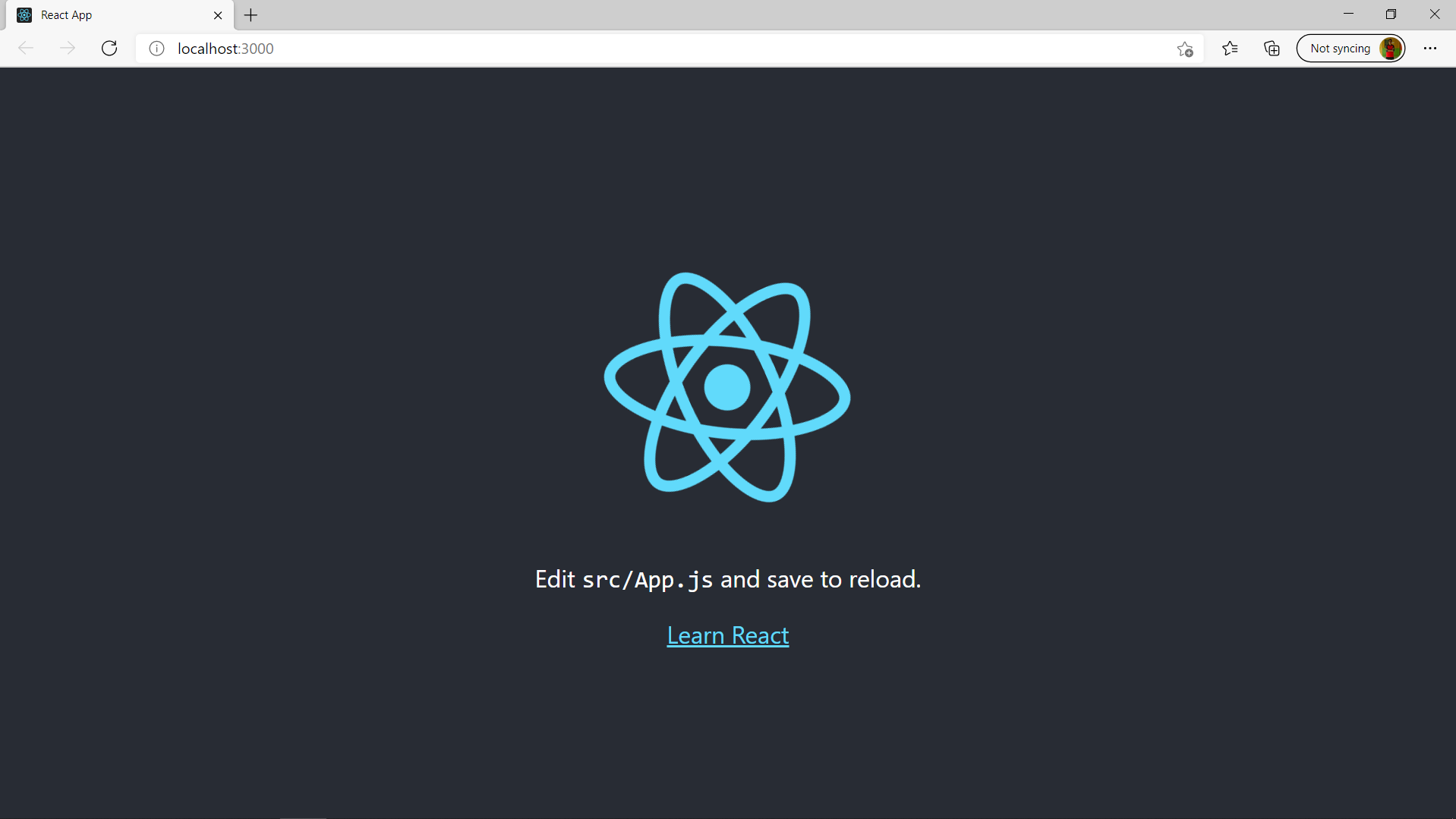
Project sebelummnya create hello world

Bagaimana cara membuat komponen dengan stateless component :

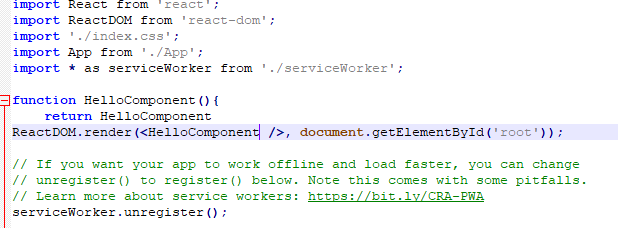
1. Membuka halaman react



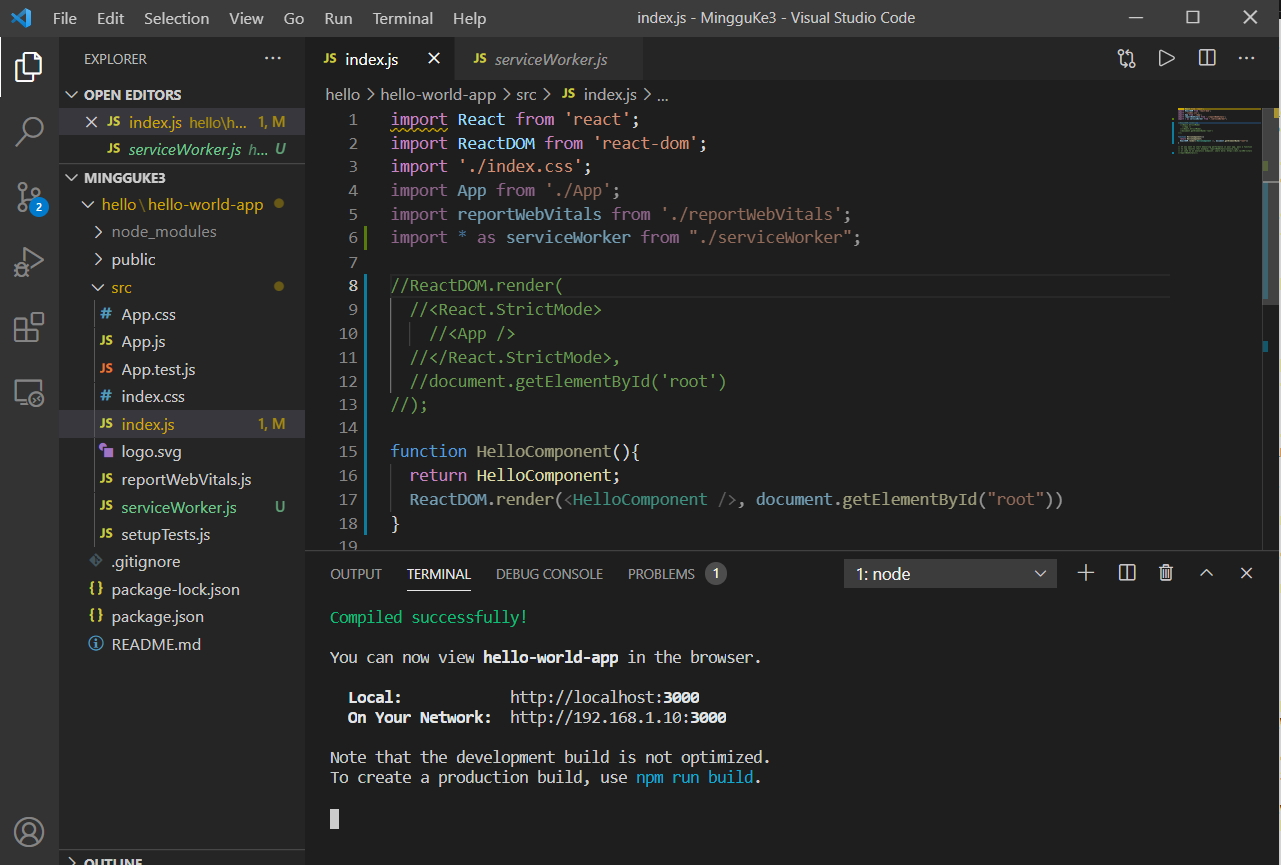
**Hasil :**



1. Buka index.js
2. Membuat function baru dengan hello component (stateless component)

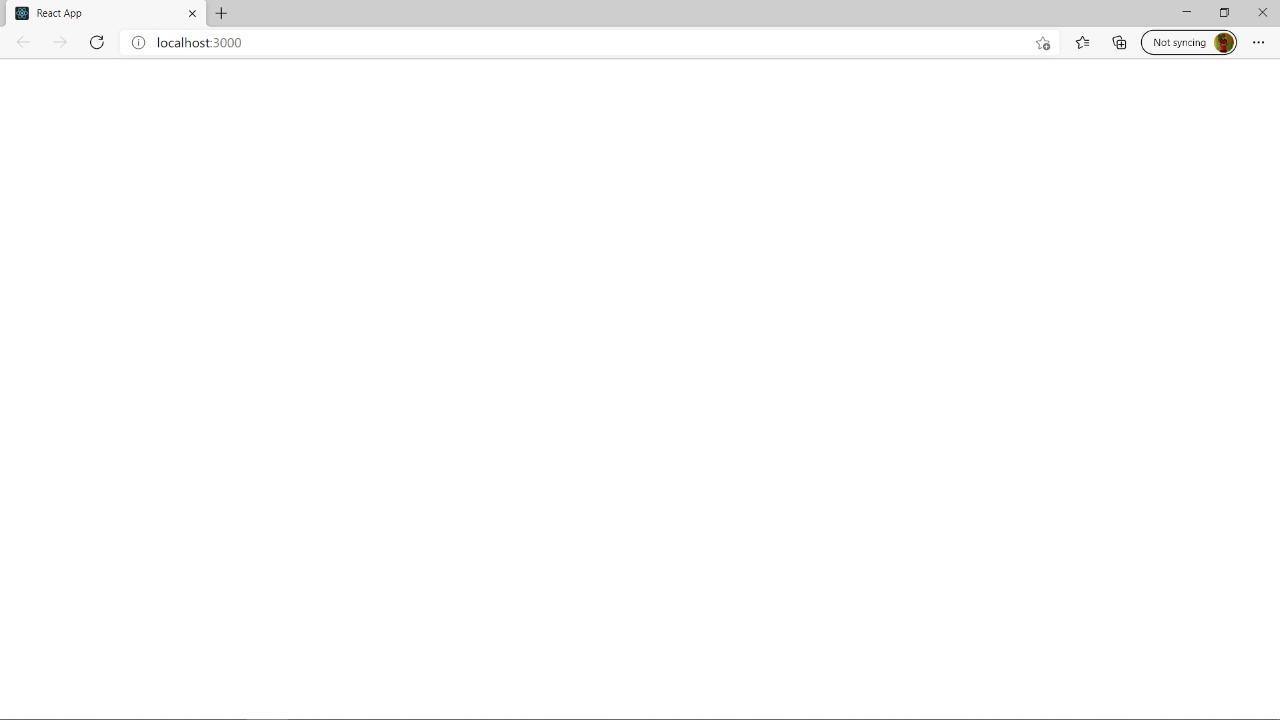


**Hasil :**



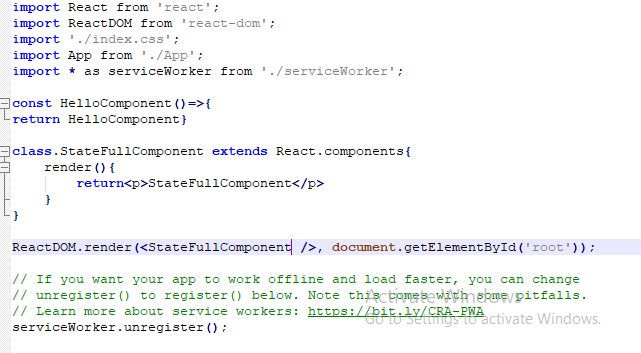
1. Run di browser dan catat hasilnya

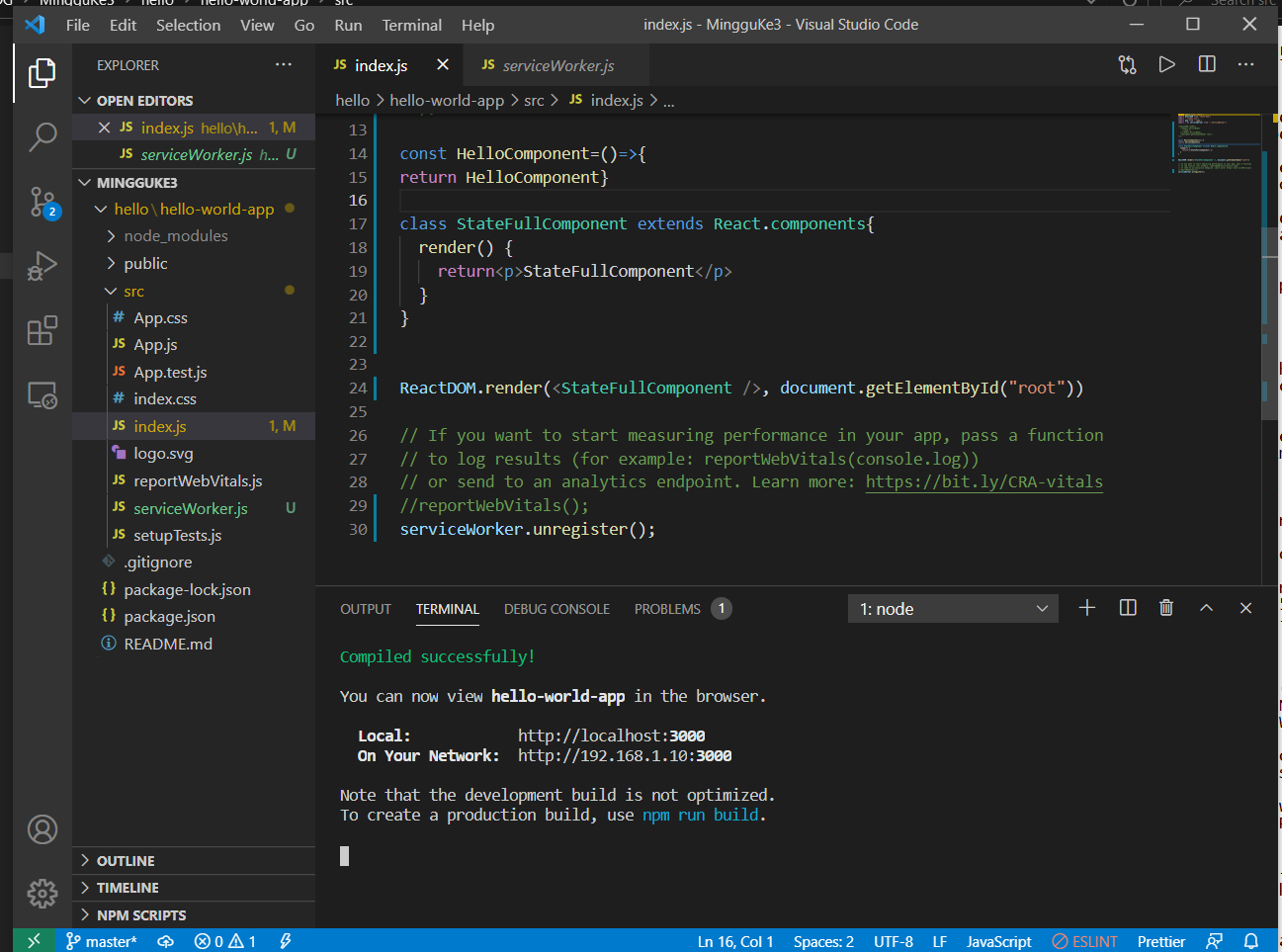
**Hasil :**

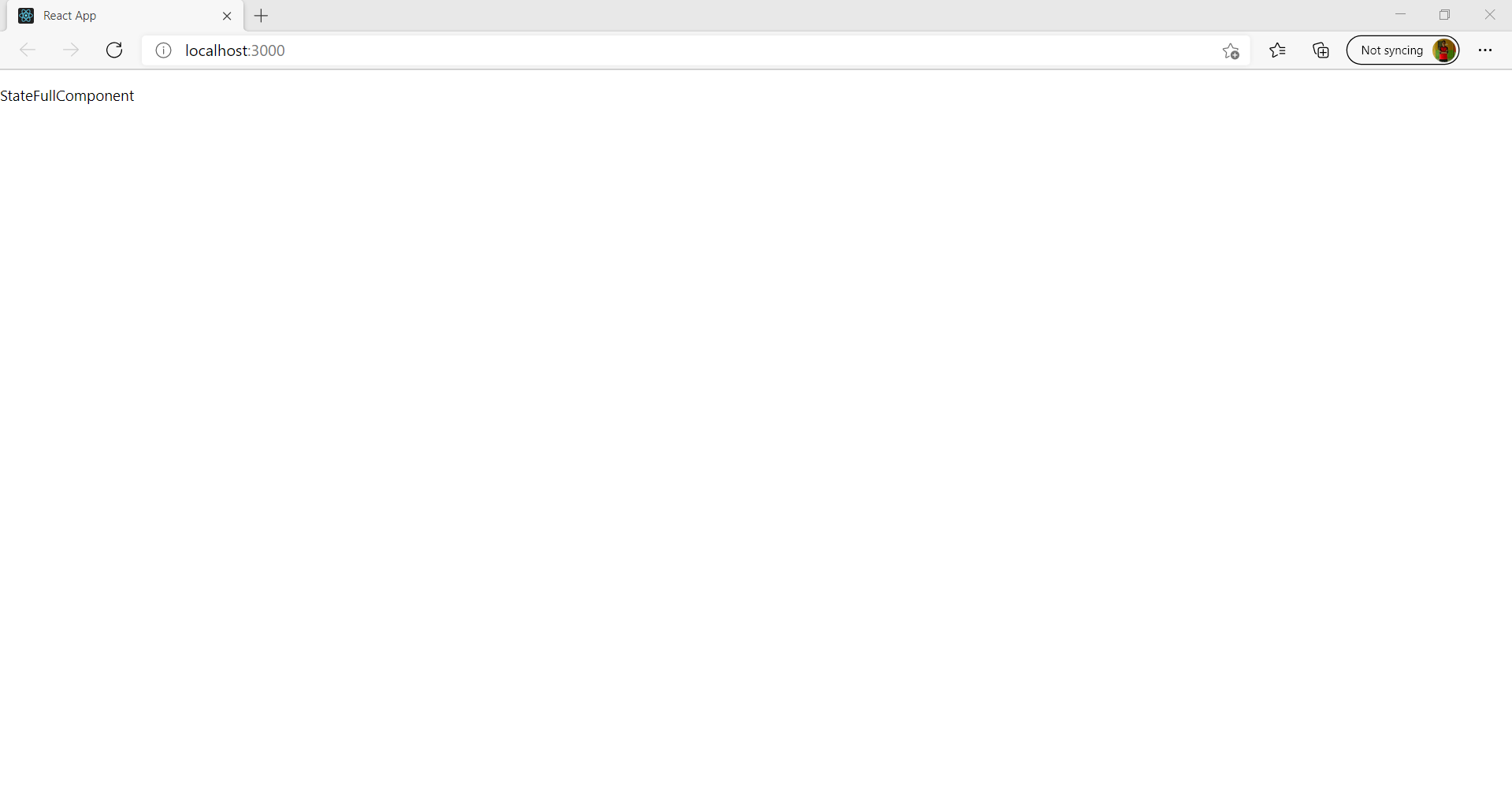


Hasilnya masih kosong

1. Buat class component / statefull component dengan memanggil react .component
2. Memanggil StateFullComponent didalam react







**Membuat Props**

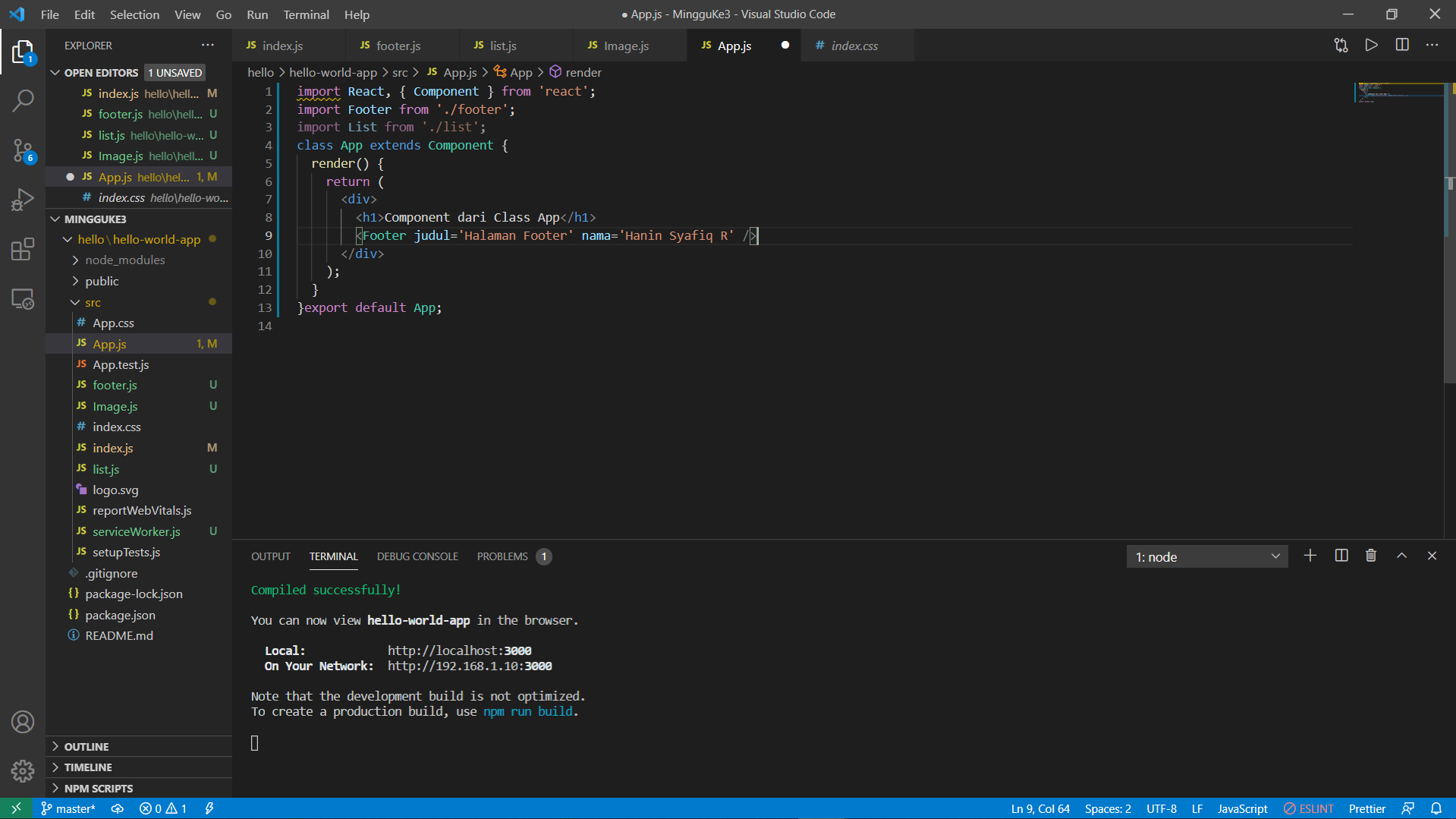
Props atau properties digunakan untuk menampung nilai yang nantinya dikirimkan ke suatu Component.

1. Buatlah Program berikut ini

import React, { Component } from 'react';  
import Header from './Header';  
import Footer from './Footer';  
import List from './List';class App extends Component {  
 render() {  
 return (  
 <div>  
 <Header/>  
 <h1>Component dari Class App</h1>  
 <List/>  
 <Footer judul='Halaman Footer' nama='Aufa' />  
 </div>  
 );  
 }  
}export default App;

1. Simpan ke Component Footer atau File Footer.js.

**Hasil :**



**Menampilkan props**

1. Buatlah Program berikut ini

import React from 'react';// Component menggunakan Functionconst Footer = (props) => {  
 return (  
 <div>  
 <h3>Halaman Footer</h3>  
 <h3>Component ini dibuat menggunakan Function bukan Class</h3>  
 <p>Nilai ini ditampilkan dari props: { props.judul } </p>  
 <p>Nama Saya: { props.nama } </p>  
 </div>  
 );  
}export default Footer;

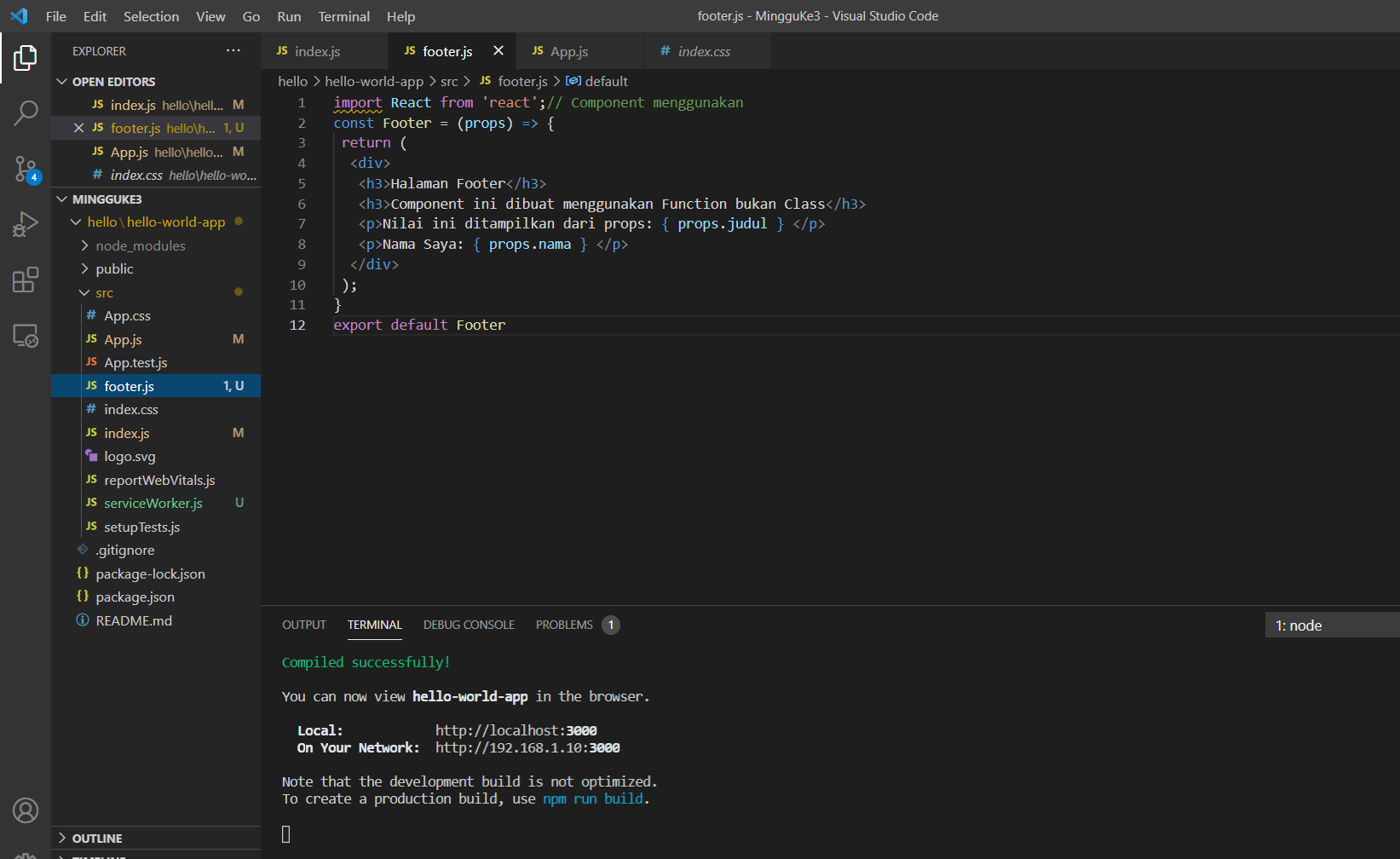
Pada kode di atas, kirim mengirimkan sebuah parameter props ke fungsi (anonymous function), lalu parameter tersebut akan kita gunakan untuk mengakses variabel yang dikirimkan dari file App.js.

# **Props pada Class Component**

Contoh Kasus:

Kita ingin menampilkan banyak gambar di dalam component list, di mana component list akan menggunakan component image.

**Hasil :**

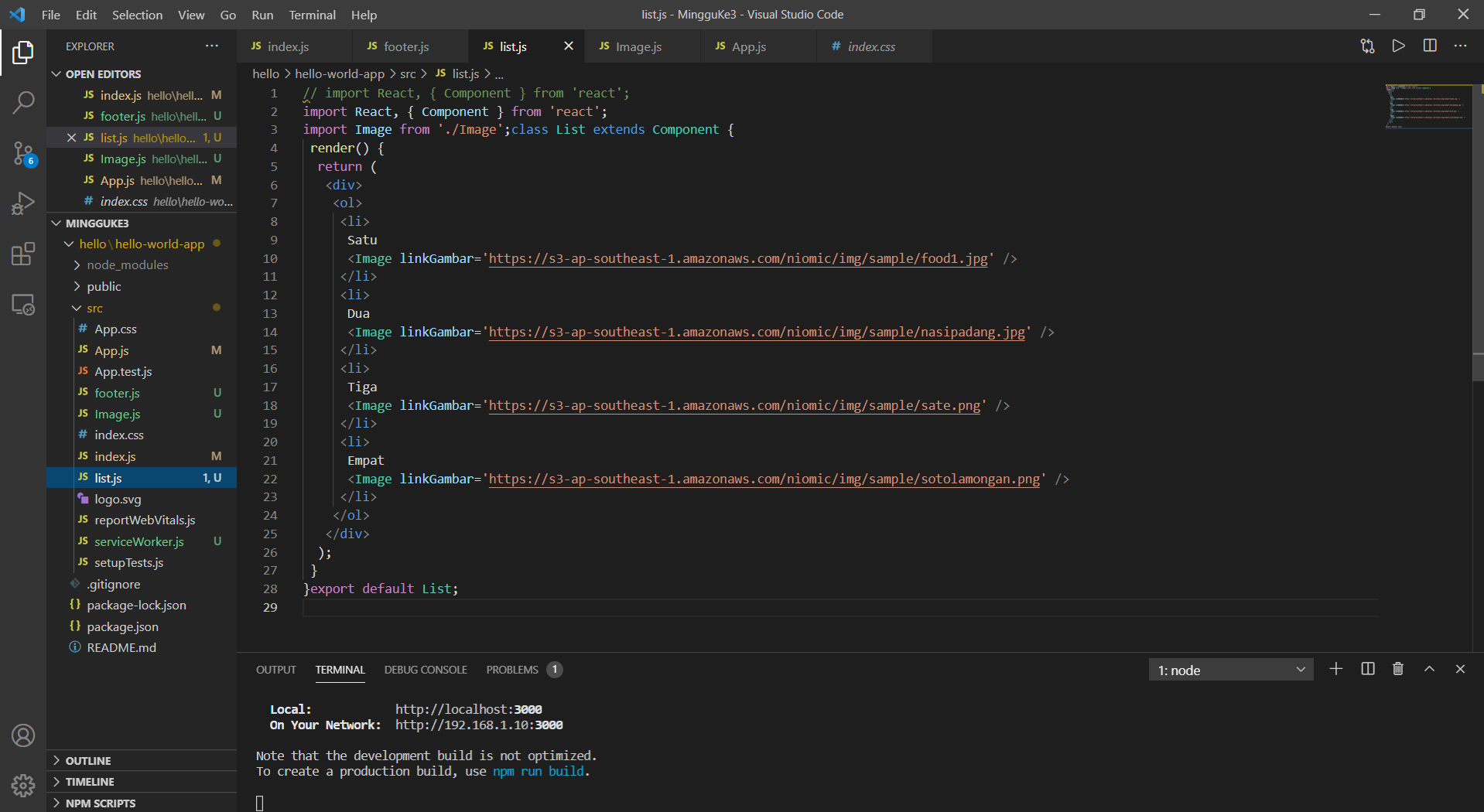


**Component List**

1. **Buatlah program berikut ini**

// import React, { Component } from 'react';  
import React, { Component } from 'react';  
import Image from './Image';class List extends Component {  
 render() {  
 return (  
 <div>  
 <ol>  
 <li>  
 Satu  
 <Image linkGambar='<https://s3-ap-southeast-1.amazonaws.com/niomic/img/sample/food1.jpg'> />  
 </li>  
 <li>  
 Dua  
 <Image linkGambar='<https://s3-ap-southeast-1.amazonaws.com/niomic/img/sample/nasipadang.jpg'> />  
 </li>  
 <li>  
 Tiga  
 <Image linkGambar='<https://s3-ap-southeast-1.amazonaws.com/niomic/img/sample/sate.png'> />  
 </li>  
 <li>  
 Empat  
 <Image linkGambar='<https://s3-ap-southeast-1.amazonaws.com/niomic/img/sample/sotolamongan.png'> />  
 </li>  
 </ol>  
 </div>  
 );  
 }  
}export default List;

Hasil :



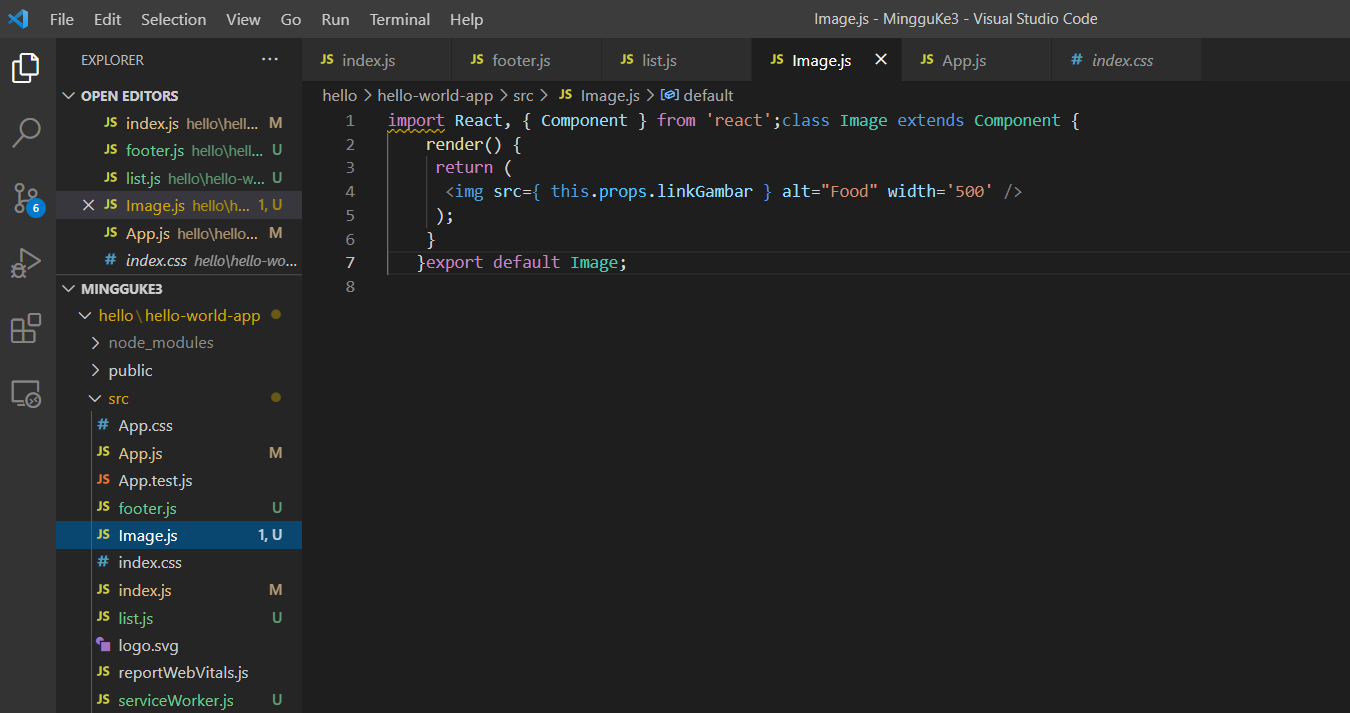
1. Pada kode di atas, kita membuat empat buah list di mana masing masing list memanggil Component image. Kemudian Component Image di panggil dengan program berikut ini

**Component Image**

1. Buatlah Program berikut ini

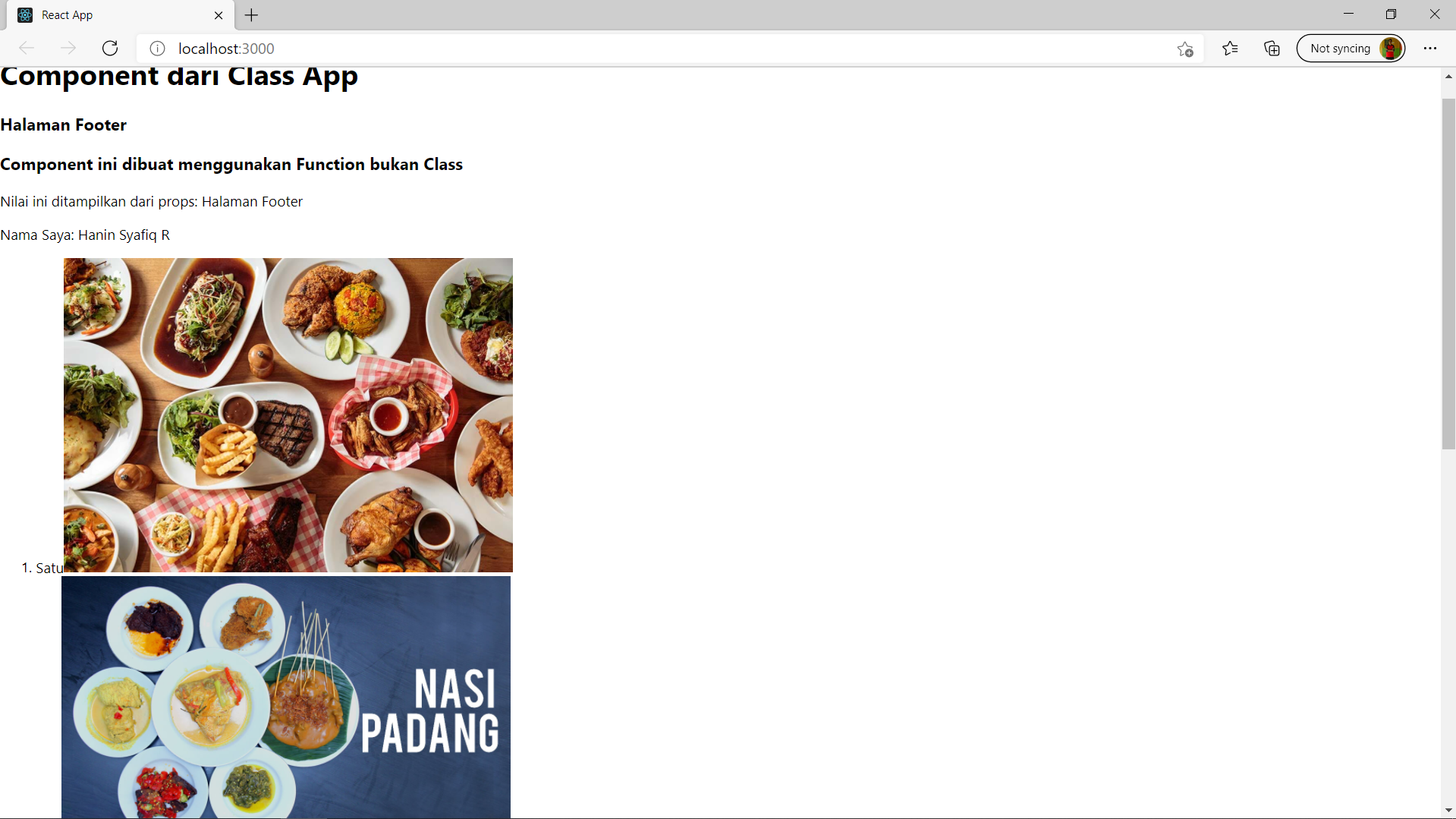
import React, { Component } from 'react';class Image extends Component {  
 render() {  
 return (  
 <img src={ this.props.linkGambar } alt="Food" width='500' />  
 );  
 }  
}export default Image;

**Hasil :**



1. Cara mengaksesnya adalah: this.props.linkGambar.
2. Dengan begitu, setiap gambar yang tampil akan memanggil nilai src yang berbeda-beda, berdasarkan yang sudah kita tentukan di Component List di atas.

**Hasil :**



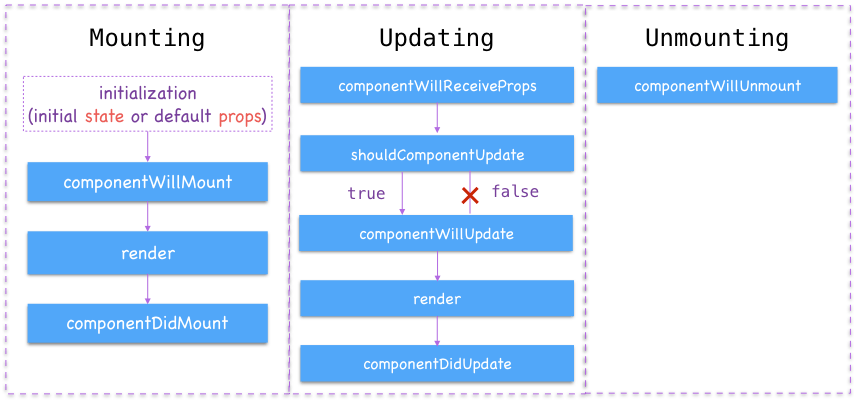
**Lifecycle Component**

Component — component di react setidaknya akan melewati tiga fase, yaitu :

1. **Mounting**
2. **Updating**
3. **Unmounting**

Pada setiap fase, components akan mengeksekusi fungsi atau method yang berbeda-beda yang di kenal sebagai lifecycle methods.

Lebih jelasnya perhatikan gambar di bawah ini !



<https://rangle.github.io/react-training/react-lifecycles/>

Buatlah Program dibawah ini dan deteksilah bagian dari lifecycle Component

import React from 'react';

import ReactDOM from 'react-dom';

class Test extends React.Component {

    constructor(props)

    {

        super(props);

        this.state = { hello : "World!" };

    }

    componentWillMount()

    {

        console.log("componentWillMount()");

    }

    componentDidMount()

    {

        console.log("componentDidMount()");

    }

    changeState()

    {

        this.setState({ hello : "Geek!" });

    }

    render()

    {

        return (

            <div>

            <h1>GeeksForGeeks.org, Hello{ this.state.hello }</h1>

            <h2>

             <a onClick={this.changeState.bind(this)}>Press Here!</a>

            </h2>

            </div>);

    }

    shouldComponentUpdate(nextProps, nextState)

    {

        console.log("shouldComponentUpdate()");

        return true;

    }

    componentWillUpdate()

    {

        console.log("componentWillUpdate()");

    }

    componentDidUpdate()

    {

        console.log("componentDidUpdate()");

    }

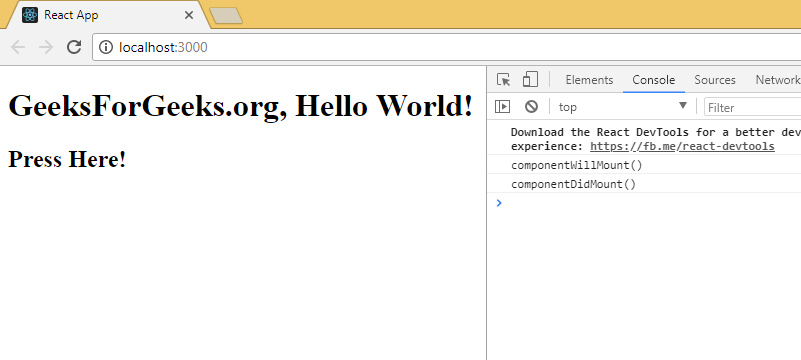
}

ReactDOM.render(

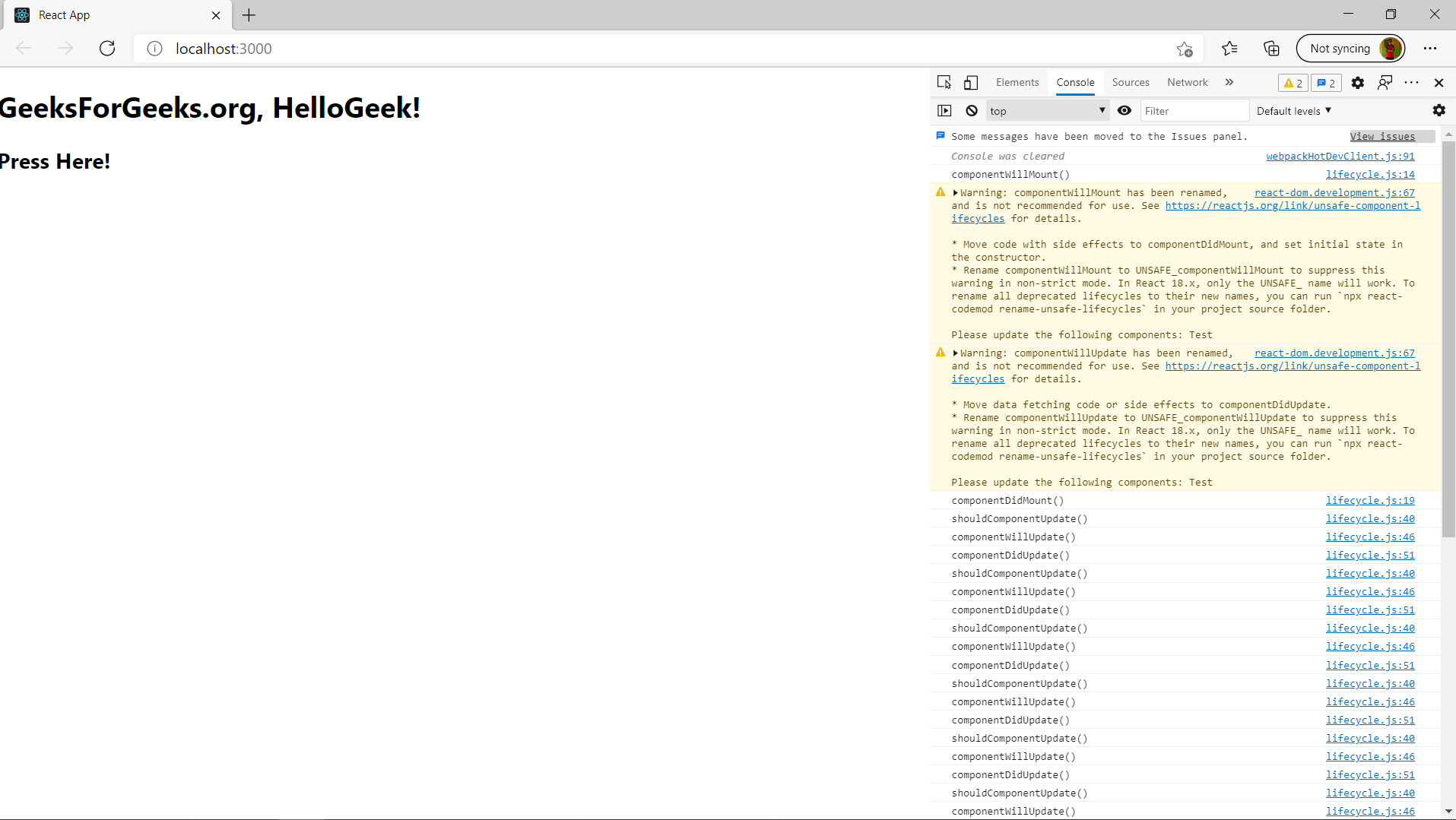
    <Test />,

    document.getElementById('root'));

1. Sehingga hasilnya berikut ini



**Hasil :**

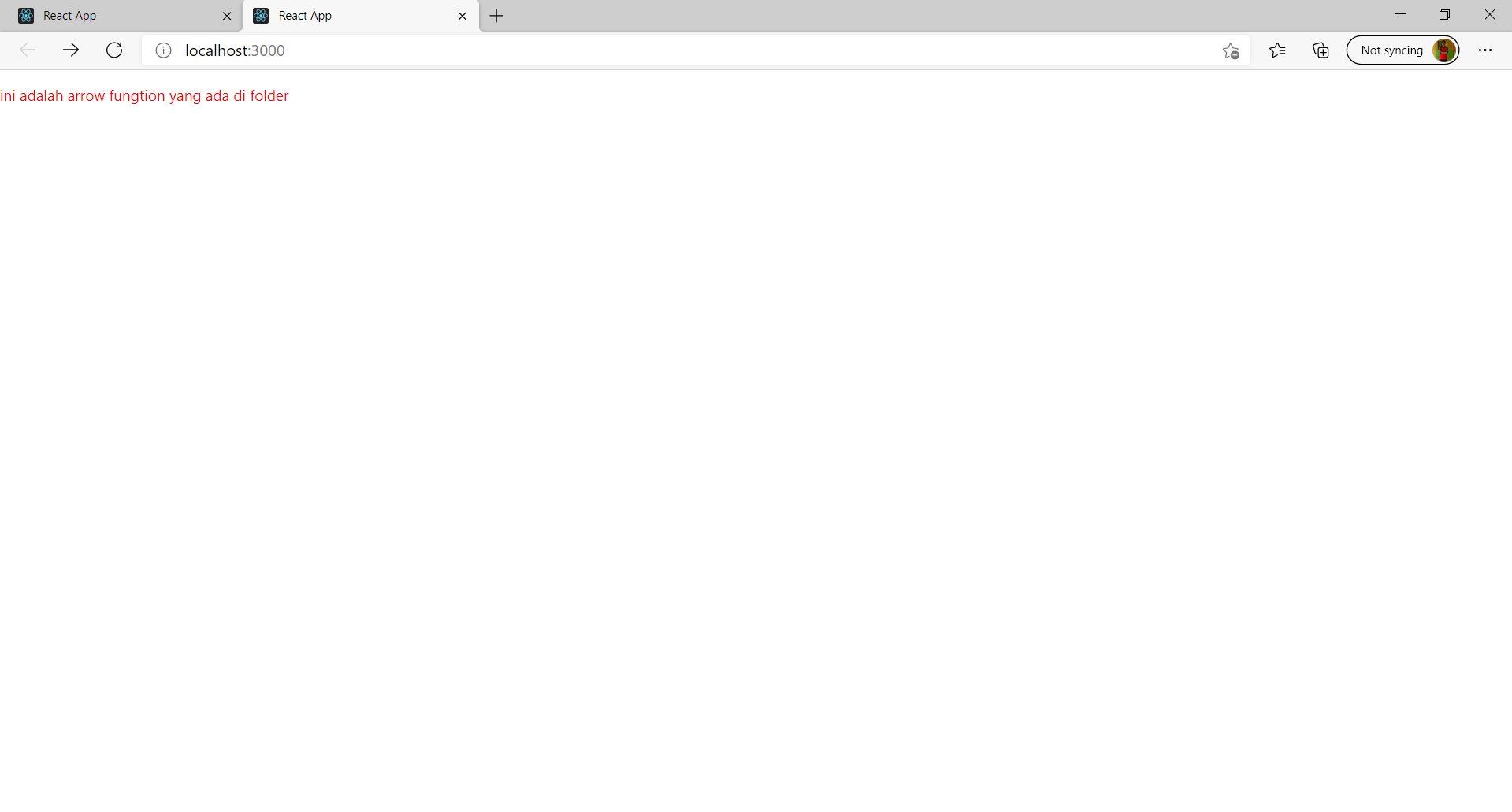


* Selain kita dapat membuat file stateless component pada folder component, kita juga dapat menambahkan file css seperti pada gambar berikut ( sebaiknya nama file css disamakan dengan nama file js nya dan tambahkan code seperti berikut)

* Setelah itu import HelloComponent.css pada HelloComponent.js seperti pada berikut

Hasilnya

**Hasil :**

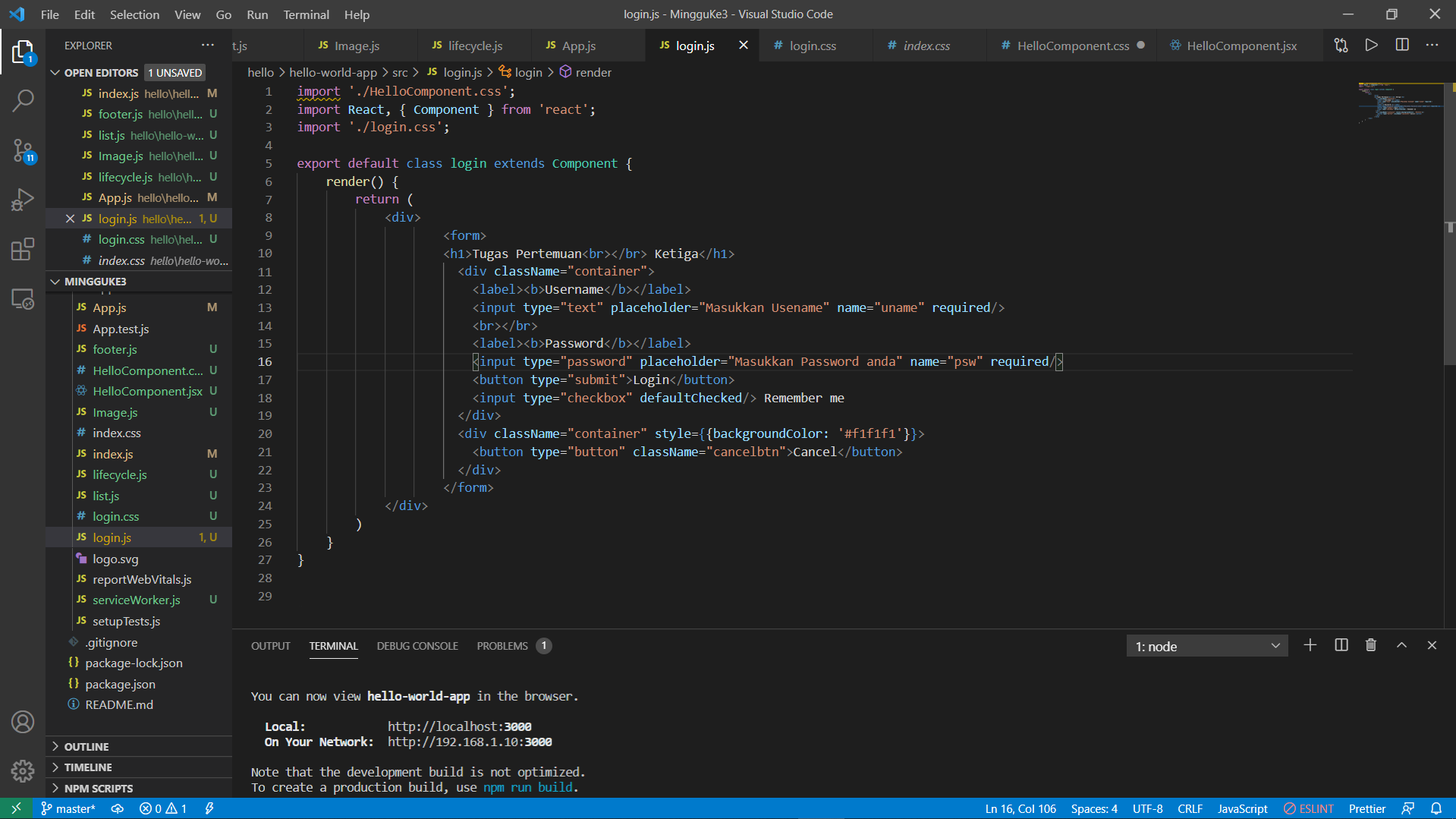


Buatlah menu login seperti pada gambar menggunakan CSS

**Jawab :**

1. **Membuat Login.js**

**Hasil :**



**Code :**

import './HelloComponent.css';

import React, { Component } from 'react';

import './login.css';

export default class login extends Component {

    render() {

        return (

            <div>

                    <form>

                    <h1>Tugas Pertemuan<br></br> Ketiga</h1>

                      <div className="container">

                        <label><b>Username</b></label>

                        <input type="text" placeholder="Masukkan Usename" name="uname" required/>

                        <br></br>

                        <label><b>Password</b></label>

                        <input type="password" placeholder="Masukkan Password anda" name="psw" required/>

                        <button type="submit">Login</button>

                        <input type="checkbox" defaultChecked/> Remember me

                      </div>

                      <div className="container" style={{backgroundColor: '#f1f1f1'}}>

                        <button type="button" className="cancelbtn">Cancel</button>

                      </div>

                    </form>

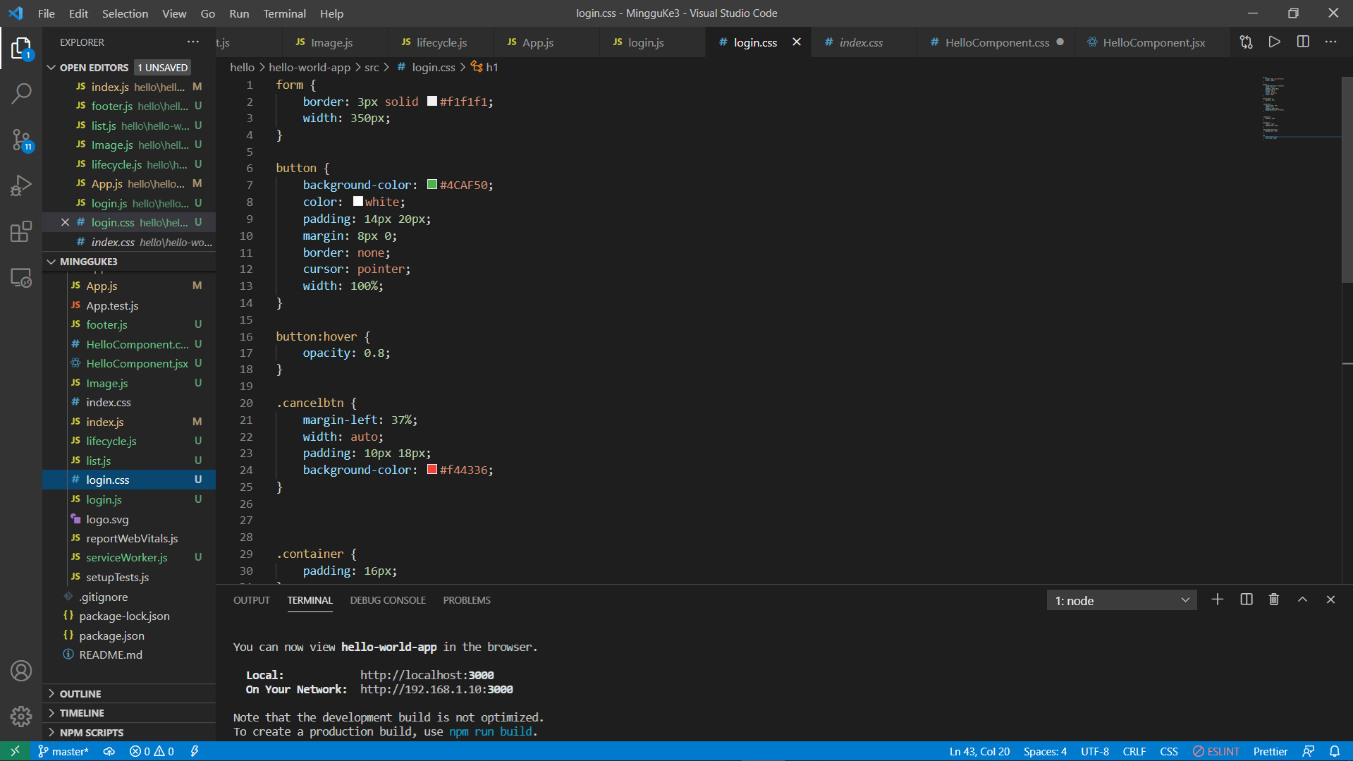
            </div>

        )

    }

}

1. **Membuat login.css**

**code :**

form {

    border: 3px solid #f1f1f1;

    width: 350px;

}

button {

    background-color: #4CAF50;

    color: white;

    padding: 14px 20px;

    margin: 8px 0;

    border: none;

    cursor: pointer;

    width: 100%;

}

button:hover {

    opacity: 0.8;

}

.cancelbtn {

    margin-left: 37%;

    width: auto;

    padding: 10px 18px;

    background-color: #f44336;

}

.container {

    padding: 16px;

}

span.psw {

    float: right;

    padding-top: 16px;

}

input[type=checkbox]{

    margin-left: 30%;

}

h1{

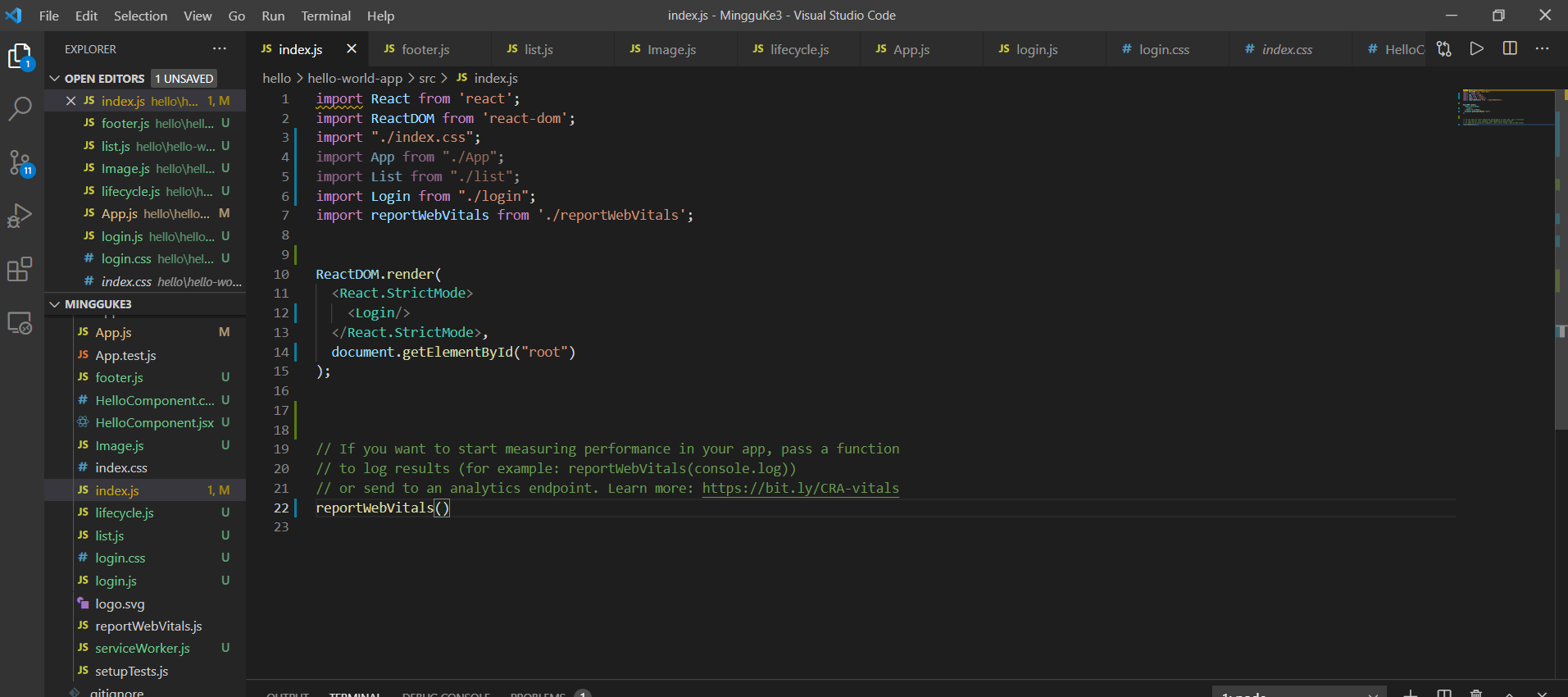
    margin-left: 27%;

    font-size: 20px;

}

1. **Mengubah code di index.js**

**Hasil :**



**Code :**

import React from 'react';

import ReactDOM from 'react-dom';

import "./index.css";

import App from "./App";

import List from "./list";

import Login from "./login";

import reportWebVitals from './reportWebVitals';

ReactDOM.render(

  <React.StrictMode>

    <Login/>

  </React.StrictMode>,

  document.getElementById("root")

);

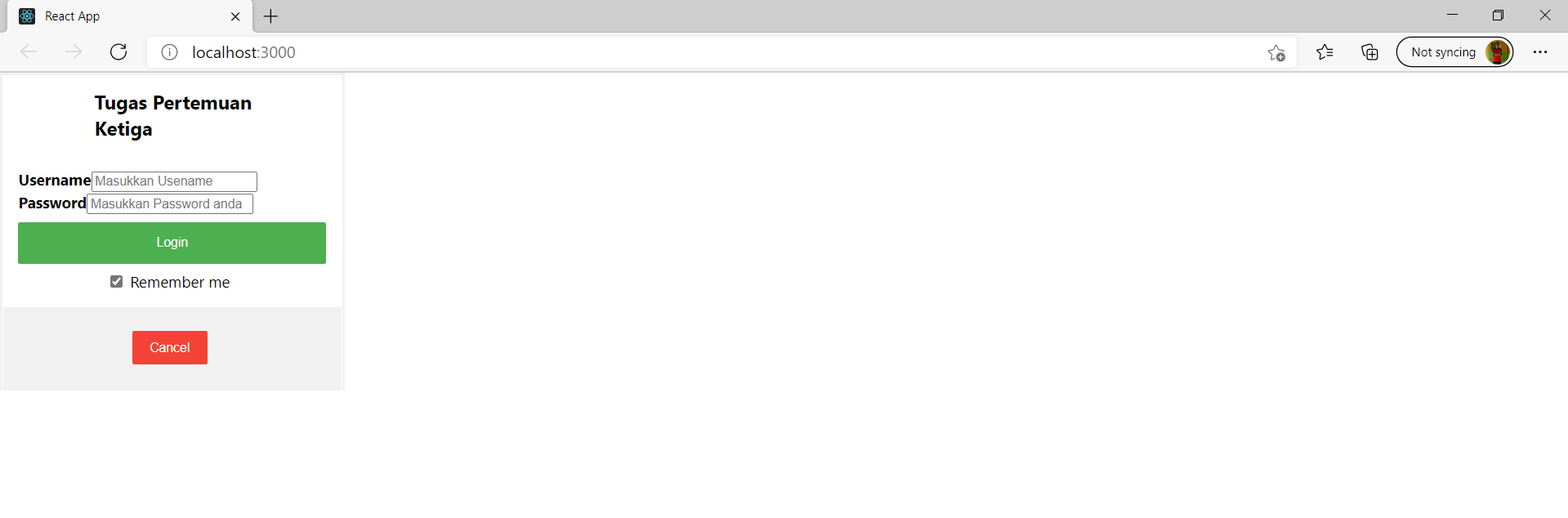
// If you want to start measuring performance in your app, pass a function

// to log results (for example: reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals

reportWebVitals()

1. **Hasil Run :**



**Link Github :**