**PRAKTIKUM 11**

**Fundamental ReactJs**

**A. TUJUAN PEMBELAJARAN**

1. Mampu memahami fundamental React JS

2. Mampu memahami ReactDOM Render

3. Mampu memahami konsep Lanjutan React JS

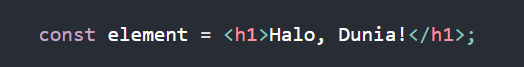
**B. DASAR TEORI**

**1. JSX**

JSX adalah kependekan dari javascript XML. Yaitu penggabungan HTML dengan javascript.

Dengan menggunakan JSX, kita dapat menuliskan HTML ke dalam javascript dengan menggunakan reactjs

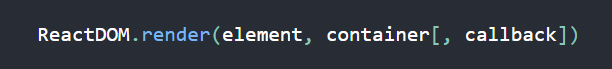
Contoh:



**2. ReactDOM**

Package react-dom menyediakan metode-metode spesifik DOM yang dapat digunakan di tingkat atas aplikasi Anda, juga sebagai jalan untuk keluar dari React model jika diperlukan. Pada umumnya komponen-komponen Anda tidak memerlukan modul ini.

Contoh:



Me-render sebuah elemen React ke dalam DOM di dalam container yang diberikan dan mengembalikan sebuah referensi kepada komponen (atau mengembalikan null untuk stateless components).

Jika elemen React tersebut sebelumnya di-render ke dalam container, ini akan memperbaruinya dan hanya mengubah DOM seperlunya untuk memperlihatkan elemen React terbaru.

Jika callback opsional diberikan, callback ini akan dijalankan setelah komponen di-render atau diperbarui.

**C. TUGAS PENDAHULUAN**

1. Apakah yang dimaksud dengan JSX dan ReactDOM ?

2. Buatlah contoh penggunaan JSX dan ReactDOM

**D. PERCOBAAN**

1. Instalasi ReactJS

Sumber <https://reactjs.org/docs/create-a-new-react-app.html>

a. Buat folder baru dengan nama reactnew di folder root



b. Buka terminal menggunakan CTRL + `

ketikan perintah

npx create-react-app my-app

tekan enter, tunggu hingga proses instalasi selesai.



c. masuk kedirektori my-app karena project react ada di folder my-app, lakukan dengan mengetikaan perintah

cd my-app

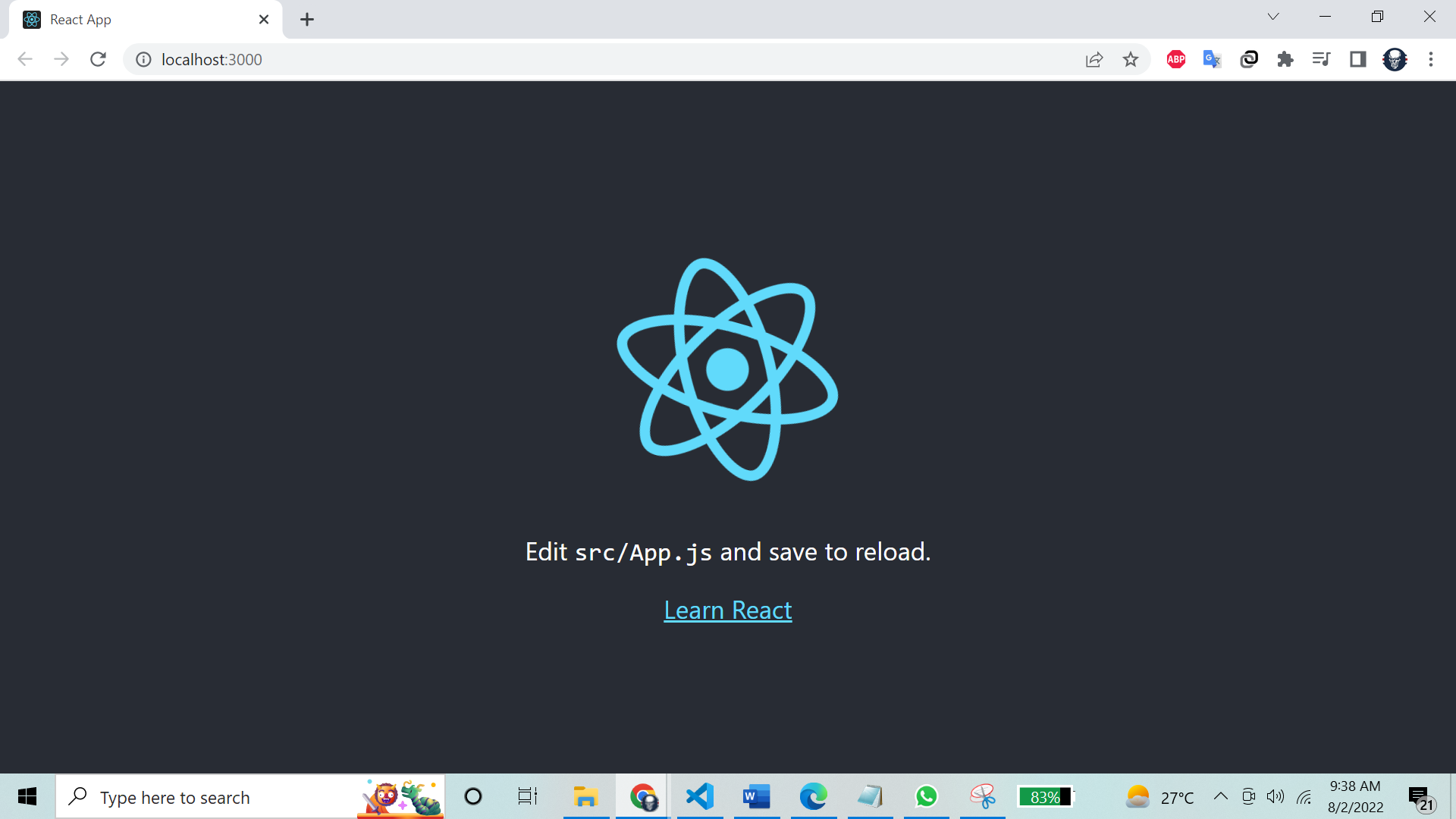


d. jalankan react dengan mengetikan perintah

npm start



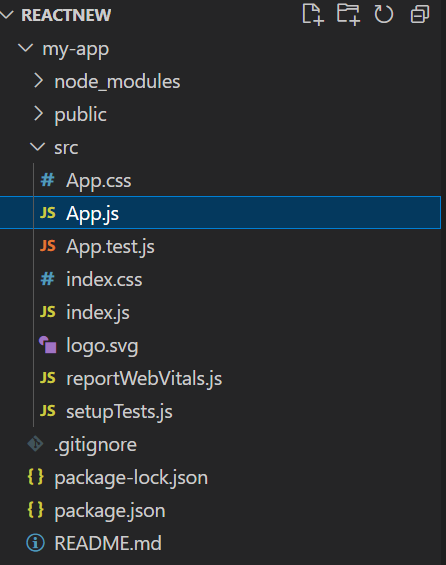
Jika berhasil maka otomatis akan menjalankan server local berjalan dibrowser seperti berikut.



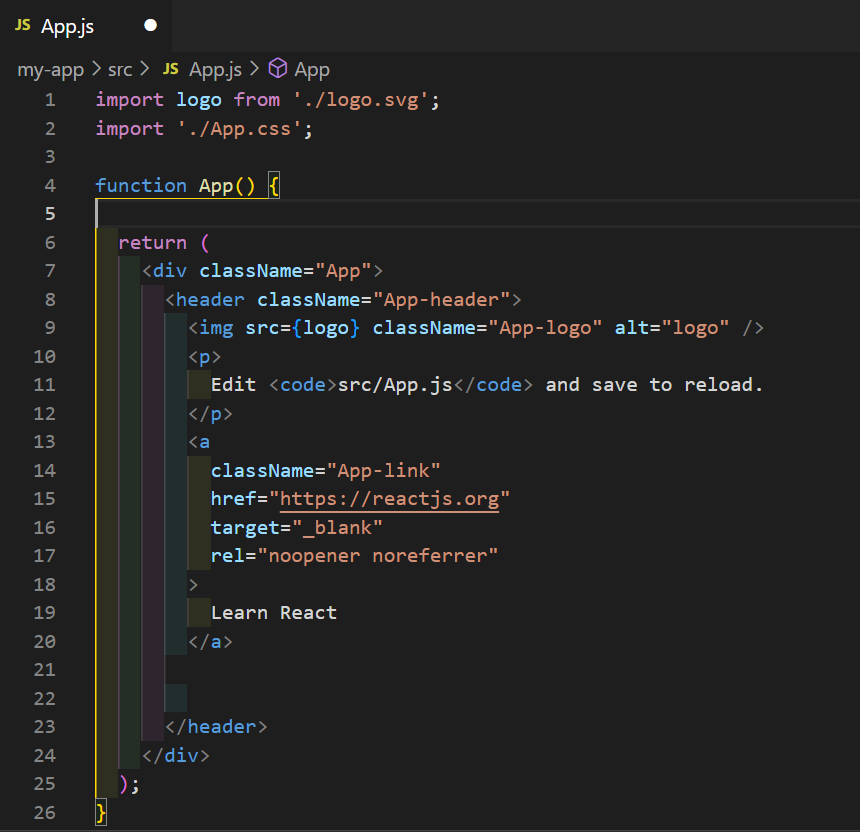
2. JSX

a. Buka file app.js

App.js adalah tempat dimana kita melakukan proses coding dalam membangun singglepageApplication / SPA



App.js



Dengan menggunakan JSX kita bisa menyisipkan html di dalam Javascript.

Sebagai contoh kita akan membuat variable dengan data string.

const userLogged = "khoironi";

Contoh diatas disebut dengan variable dengan data string biasa.

Pada JSX kita bisa meletakan script HTML didalam variable seperti berikut.

const userLoggedJSX = <strong>khoironi</strong>;

b. Buatlah variabel dengan data string biasa, dan buat variabel dengan penulisan JSX dengan menyisipkan script HTML didalamnya, letakkan di antara function APP pada app.js seperti dibawah.

import logo from './logo.svg';

import './App.css';

function App() {

    const userLogged = "khoironi";

    const userLoggedJSX = <strong>khoironi for future</strong>;

  return (

    <div className="App">

      <header className="App-header">

        <img src={logo} className="App-logo" alt="logo" />

        <p>

          Edit <code>src/App.js</code> and save to reload.

        </p>

        <a

          className="App-link"

          href="https://reactjs.org"

          target="\_blank"

          rel="noopener noreferrer"

        >

          Learn React

        </a>

      </header>

    </div>

  );

}

export default App;

c. Hapuslah pada bagian header menjadi seperti berikut.

import logo from './logo.svg';

import './App.css';

function App() {

    const userLogged = "khoironi";

    const userLoggedJSX = <strong>khoironi for future</strong>;

  return (

    <div className="App">

    </div>

  );

}

export default App;

Lalu kita akan mencoba melakukan komparasi untuk membandingkan dua variabel yang sudah kita buat, cara penggunaannya dengan menggunakan curly bracket seperti berikut

{userLogged}

{userLoggedJSX}

Contoh lengkapnya seperti ini, simpan dan lihat hasilnya.

import logo from './logo.svg';

import './App.css';

function App() {

    const userLogged = "khoironi";

    const userLoggedJSX = <strong>khoironi for future</strong>;

  return (

    <div className="App">

      {userLogged}

      {userLoggedJSX}

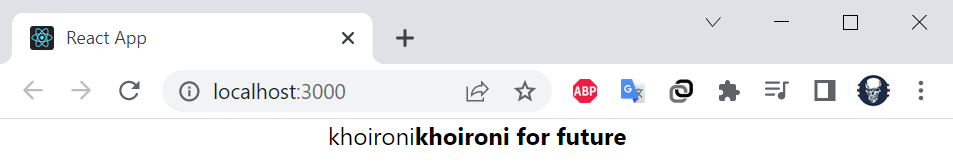
    </div>

  );

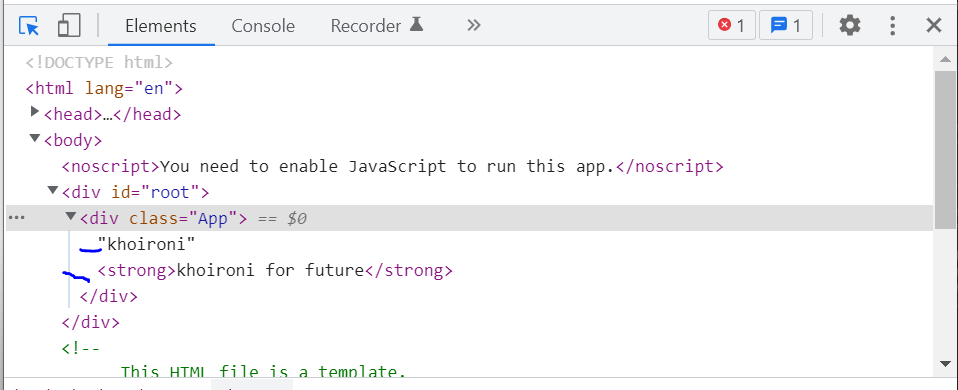
}

export default App;

maka hasilnya akan seperti ini



Coba kita lakukan inspect element dan arahkan ke tuilisan yang muncul, maka akan terlihat perbedaannya antara dua variabel yang kita buat, variabel yang pertama merupakan string biasa yang tidak dibungkus dengan tag html, variabel yang kedua menggunakan penulisan JSX dimana kita bisa menyisipkan tag html dialam javascript di dalam variabel.



3. ReactDOM Render

Pada react ini akan merender javascript (a) dalam satu dokumen atau root (b)

|  |  |
| --- | --- |
|  |  |
| a | b |

Jika kita lihat pada file index.htnl (c) disitu terdapat id root tempat semua hasil render di tampilkan, dan pada dokumen index.js (d) merender App yang ada pada app.js (a) dan ditampilkan ke root.

|  |  |
| --- | --- |
|  |  |
| c | d |

Kemudian kita lakukan uji coba dengan menambahkan suatu komponen header 1 pada class app, komponen tersebut akan di render dan ditampilkan pada id root yang ada pada index.html

import logo from './logo.svg';

import './App.css';

function App() {

  return (

    <div className="App">

     <h1>belajar bareng</h1>

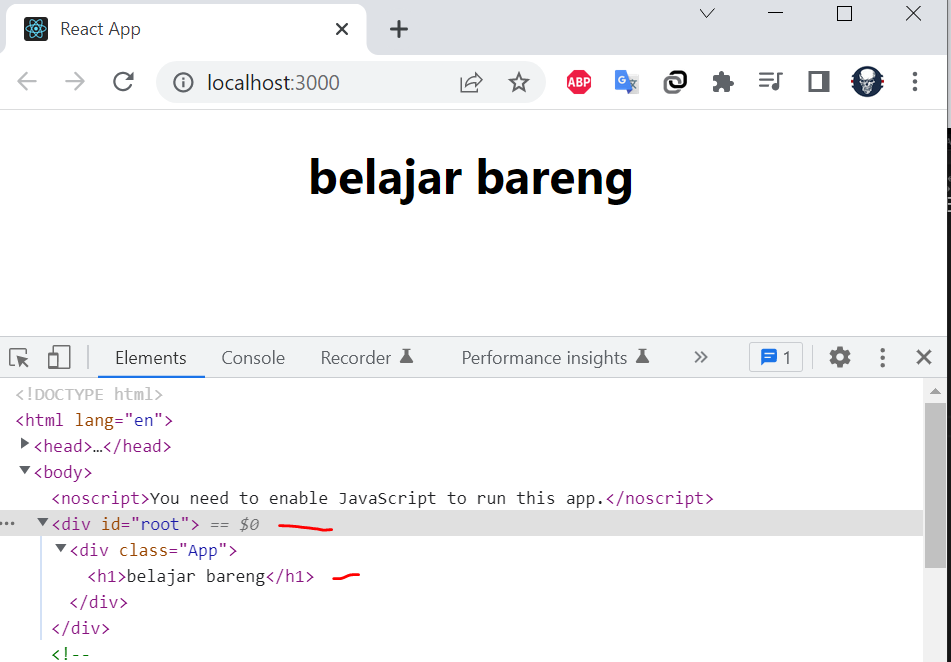
    </div>

  );

}

export default App;

berikut adalah hasilnya.



Lakukan uji coba dengan menambahkan header 1 dan paragraph, simpan dan lihat hasilnya.

import logo from './logo.svg';

import './App.css';

function App() {

  return (

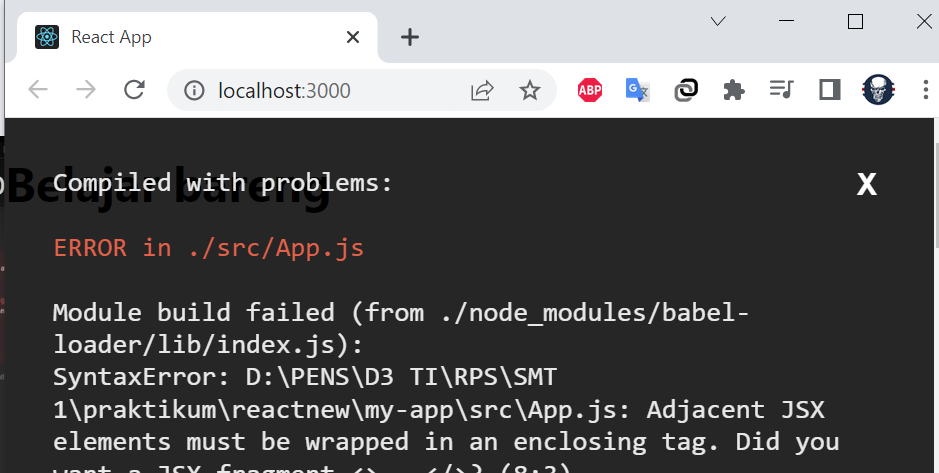
   <h1>Belajar bareng</h1>

   <p>meningkatkan keahlian design</p>

  );

}

export default App;



hasilnya akan eror sperti diatas, karena pada JSX jika kita memiliki lebih dari satu komponen kita perlu membungkusnya dengan parent <div>

import logo from './logo.svg';

import './App.css';

function App() {

  return (

   <iv>

    <h1>Belajar bareng</h1>

    <p>meningkatkan keahlian design</p>

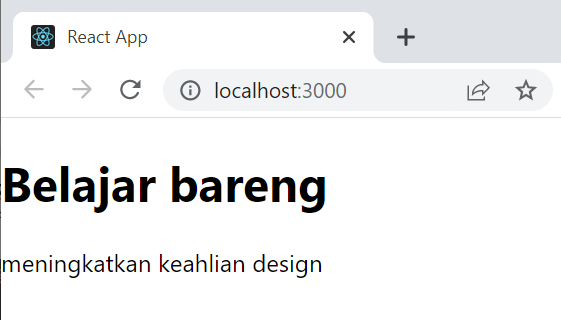
   </iv>

  );

}

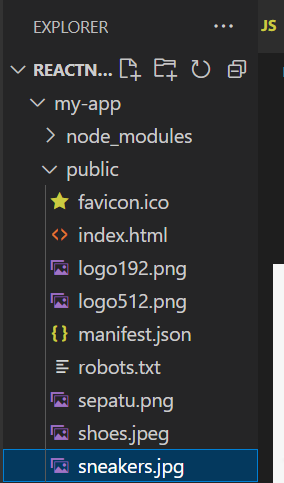
export default App;

dan hasilnya akan seperti ini



4. Design WEB Part 1

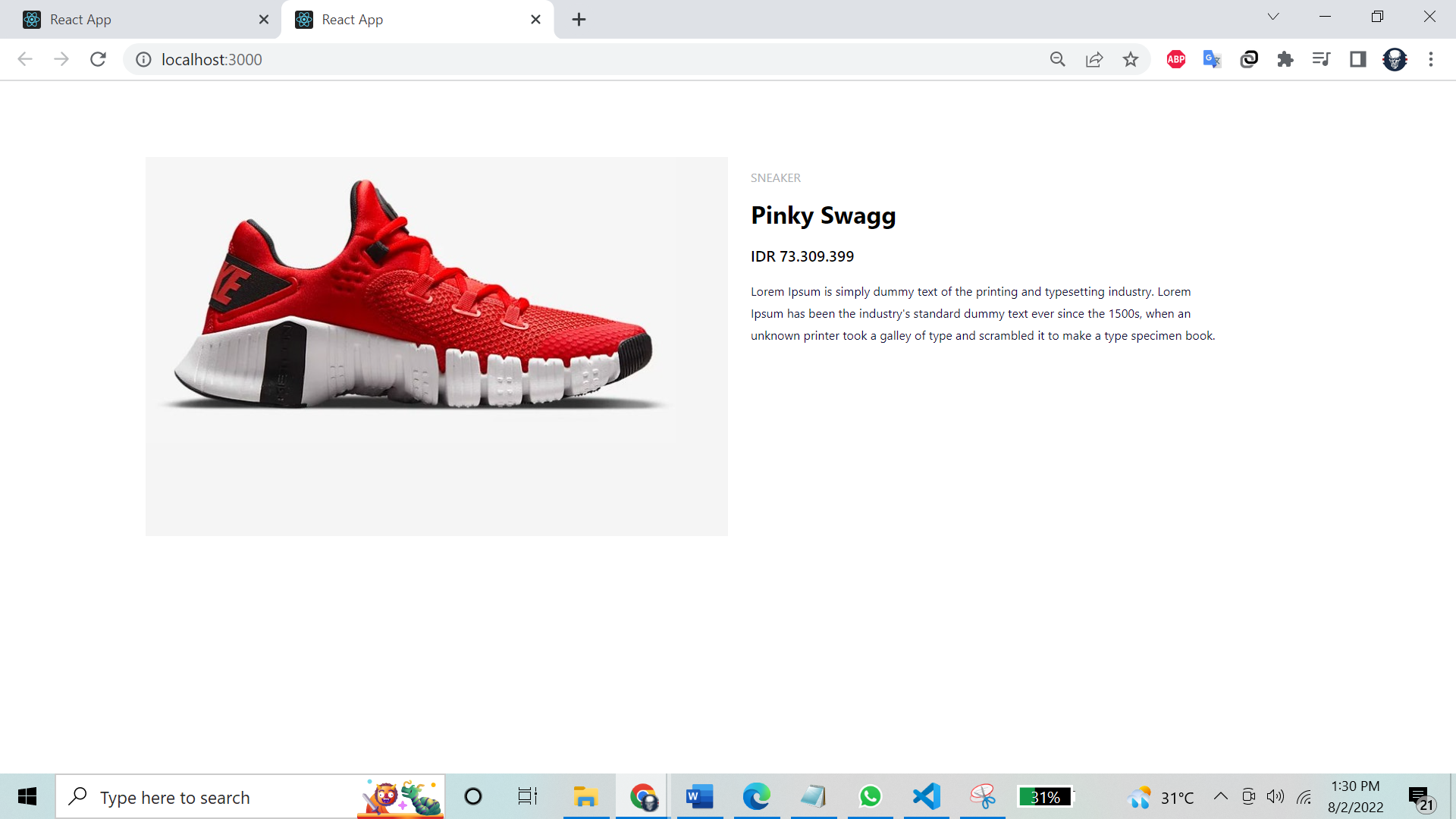
Siapkan gambar dan letakkan pada folder public seperi berikut.



Pada App.js dibagian return tambahkan script html seperti berikut (a). dan pada App.css tambahkan script css seperti berikut.

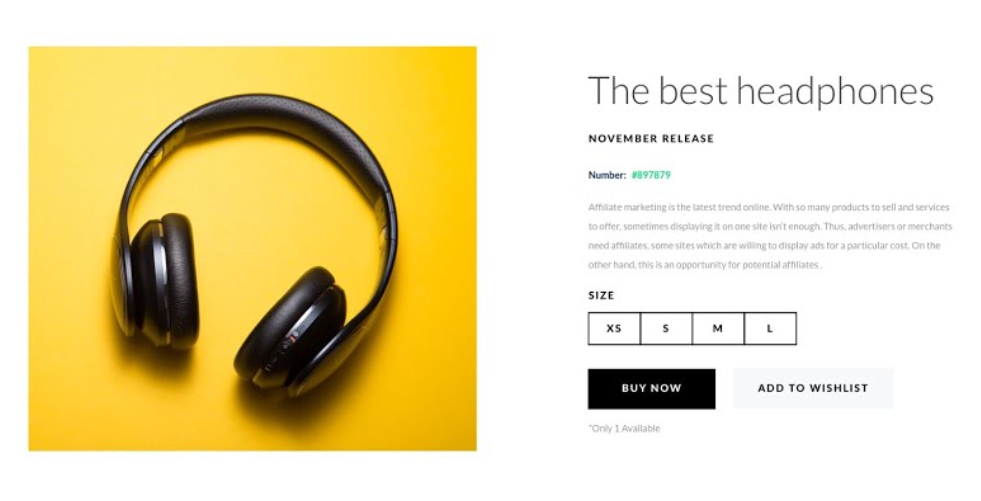
|  |  |
| --- | --- |
| import logo from './logo.svg';  import './App.css';  function App() {    return (     <div className="Parentbox">      <div className="Foto">        <img src="sneakers.jpg"/>      </div>      <div className="Deskripsi">        <p className="Cate">SNEAKER</p>        <h1 className="Title">Pinky Swagg</h1>        <p className="Price">IDR 73.309.399</p>        <p className="Info">        Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.        </p>      </div>     </div>    );  }  export default App; | .Parentbox{    width: 80%;    overflow: hidden;    margin: auto;    margin-top: 100px;  }  .Foto{    float: left;    width: 50%;    height: 500px;    background: whitesmoke;  }  .Foto img{    width: 700px;  }  .Deskripsi{    width: 40%;    overflow: hidden;    float: left;    margin-left: 30px;  }  .Deskripsi .Info {    line-height: 1.8;    color: rgb(18, 11, 49);  }  .Deskripsi .Price{    font-size: 20px;    font-weight: 500;  }  .Deskripsi .Title {    margin: 0;  }  .Deskripsi .Cate {    color: #acaeb2;  } |
| a | b |

Dan hasilnya akan seperti berikut ini



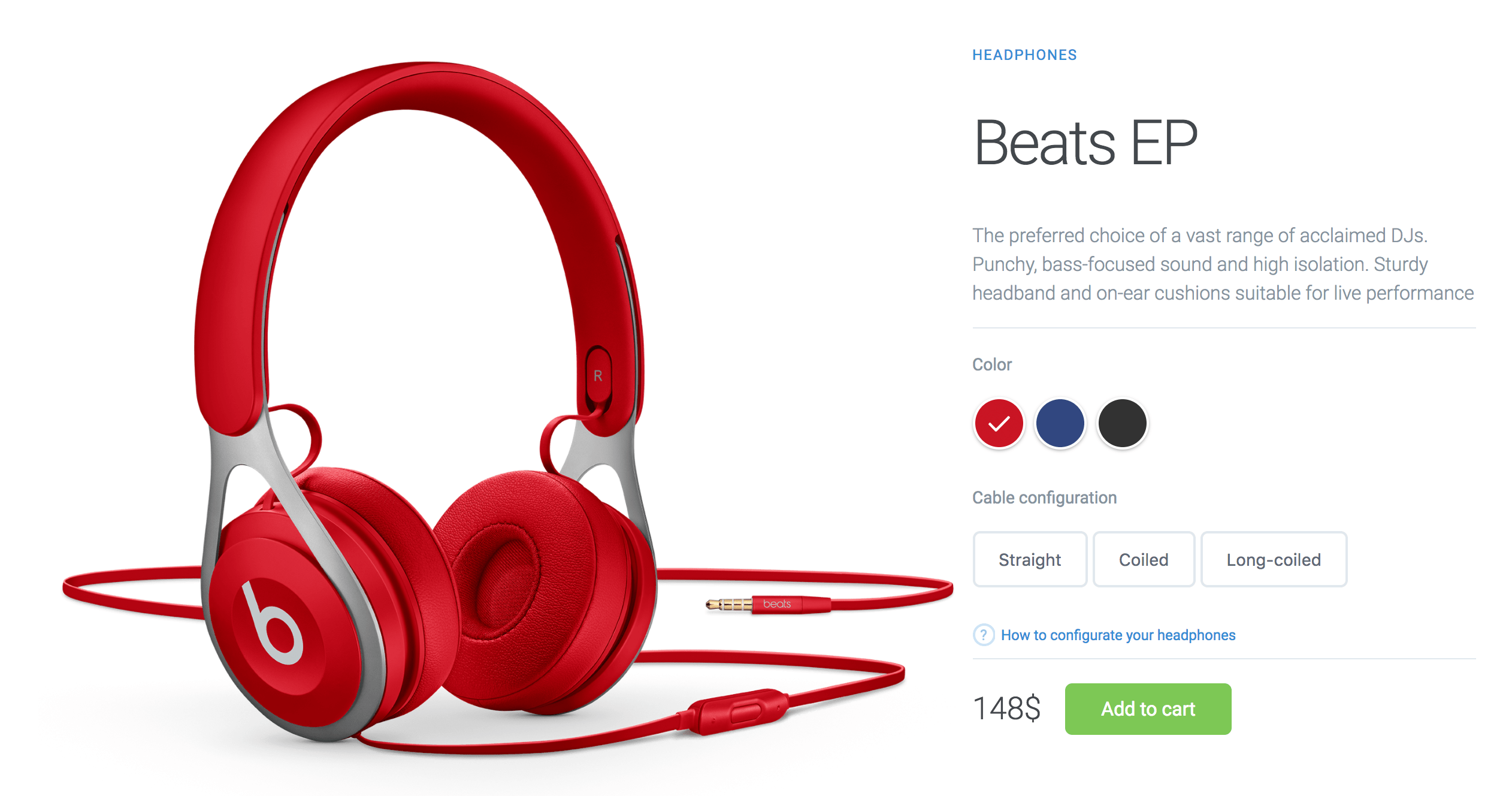
**E. TUGAS**

Buatlah halaman Detail Product Seperti berikut ini menggunakan React.



**F. Latihan**

Buatlah halaman Detail Product Seperti berikut ini menggunakan React.



**G. LAPORAN RESMI**

Kumpulkan hasil latihan dan tugas di atas. Tambahkan analisa dalam laporan resmi.