

Laporan

Modul 5

Waktu Pelaksanaan : Kamis, 18 Maret 2019 Pukul 12.50 – 18.00

Mata Kuliah : Pemrograman Berbasis Framework

Tempat : Gedung Sipil Lantai 7 Ruang LPR 5

Pelaksana : Satria Putra Sabana

NIM / Absn : 1741720035 / 25

Kelas : TI – 3G

Praktikum

Basic Router

1. Setelah sukses melakukan instalasi react (npm install), kemudian menambahkan library react router dengan menambahkan npm install react-router-dom
2. Setelah selesai membuat file baru bernama appBasicExample.js dan membuat fungsi seperti berikut

```

export default function BasicExample(){
  return(
    <Router>
      <div>
        <ul>
          <li>
            <Link to="/">Home</Link>
          </li>
          <li>
            <Link to="/about">About</Link>
          </li>
          <li>
            <Link to="/dashboard">Dashboard</Link>
          </li>
        </ul>
        <hr/>

        <switch>
          <Route exact path = "/">
            <Home/>
          </Route>
          <Route exact path = "/about">
            <About/>
          </Route>
          <Route exact path = "/dashboard">
            <Dashboard/>
          </Route>
        </switch>
      </div>
    </Router>
  );
}

```

```

function Home(){
  return(
    <div>
      <h2>Home</h2>
    </div>
  );
}

function About(){
  return(
    <div>
      <h2>About</h2>
    </div>
  );
}

function Dashboard(){
  return(
    <div>
      <h2>Dashboard</h2>
    </div>
  );
}

```

3. Kemudian pada index.js import file appBasicExample.js

```
import BasicExample from './appBasicExample';
import ParamsExample from './appParamsExample';
import NestingExample from './appNestingExample';
import UseRedirects from './appUseRedirects';
import Tugas from './appTugas';
import * as serviceWorker from './serviceWorker';

// ReactDOM.render(<App />, document.getElementById('root'));
ReactDOM.render(<BasicExample />, document.getElementById('root'));

// If you want your app to work offline and load faster, you can change
// unregister() to register() below. Note this comes with some pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
serviceWorker.unregister();
```

4. Menjalankan server

- [Home](#)
- [About](#)
- [Dashboard](#)

About

Penjelasan : Pada fungsi basic example terdapat fungsi route yang berfungsi merutekan link dan memanggil fungsi sesuai dengan linknya

Url Parameter

Membuat link react router dengan menggunakan params.

1. Membuat file baru bernama appParamsExample.js kemudian membuat fungsi ParamsExample seperti berikut

```
import React from 'react';
import {
  BrowserRouter as Router,
  Switch,
  Route,
  Link,
  useParams,
  useRouteMatch
} from 'react-router-dom';
import './App.css';
```

```
export default function ParamsExample(){
  return(
    <Router>
      <div>
        <h2>Accounts</h2>
        <ul>
          <li>
            <Link to="/netflix">NetFlix</Link>
          </li>
          <li>
            <Link to="/gmail">Gmail</Link>
          </li>
          <li>
            <Link to="/yahoo">Yahoo</Link>
          </li>
          <li>
            <Link to="/amazon">Amazon</Link>
          </li>
        </ul>

        <switch>
          <Route path("/:id" children={<Child />}/>
        </switch>
      </div>
    </Router>
  );
}
```

```
function Child(){
  let {id} = useParams();

  return(
    <div>
      <h3>ID : {id}</h3>
    </div>
  );
}
```

2. Menjalankan server

Accounts

- [NetFlix](#)
- [Gmail](#)
- [Yahoo](#)
- [Amazon](#)

ID : yahoo

Penjelasan : Hampir sama dengan route sebelumnya pada route ini menggunakan params sebagai masukan berdasarkan id

Nesting Router

Percabangan link didalam react router

1. Membuat file baru bernama appNestingRouter.js kemudian membuat fungsi seperti berikut

```
import React from 'react';
import {
  BrowserRouter as Router,
  Switch,
  Route,
  Link,
  useParams,
  useRouteMatch
} from "react-router-dom";
import './App.css';

export default function NestingExample(){
  return(
    <Router>
      <div>
        <ul>
          <li>
            <Link to="/">Home</Link>
          </li>
          <li>
            <Link to="/topics">Topics</Link>
          </li>
        </ul>
        <hr/>

        <Switch>
          <Route exact path="/">
            <Home/>
          </Route>
          <Route path="/topics">
            <Topics/>
          </Route>
        </Switch>
      </div>
    </Router>
  );
}
```

```
function Home(){
  return(
    <div>
      <h2>Home</h2>
    </div>
  );
}
```

```
function Topics(){
  //'path' untuk membuat jalur <Route> yang terhadap rute induk,
  //'sedangkan url untuk membuat link
  let {path, url} = useRouteMatch();
  return(
    <div>
      <h2>Topics</h2>
      <ul>
        <li>
          <Link to={`${url}/Sate, Nasi Goreng`}>Kuliner</Link>
        </li>
        <li>
          <Link to={`${url}/Wisata Alama, Semeru`}>Traveling</Link>
        </li>
        <li>
          <Link to={`${url}/Savana, JW Marriot`}>Review Hotel</Link>
        </li>
      </ul>

      <switch>
        <Route exact path={path}>
          <h3>Please select a topic.</h3>
        </Route>
        <Route path={`${path}/:topicId`}>
          <Topic/>
        </Route>
      </switch>
    </div>
  );
}
```

```
function Topic(){
  let {topicId} = useParams();
  return(
    <div>
      <h3>{topicId}</h3>
    </div>
  );
}
```

2. Menjalankan server

- [Home](#)
 - [Topics](#)
-

Topics

- [Kuliner](#)
- [Traveling](#)
- [Review Hotel](#)

Wisata Alama, Semeru

Penjelasan : Nesting Route adalah route yang didalamnya terdapat route atau child

Redirects (Auth)

Redirects dapat digunakan untuk masuk ke halaman yang membutuhkan authentication. Setelah masuk dan melakukan klik ke menu lain termasuk masuk ke halaman yang memerlukan authentication, posisi user tidak perlu login ulang (karena posisi sudah logged) sampai user melakukan sign out.

1. Membuat file js baru bernama appUseRedirects.js kemudian membuat fungsi berikut

```
import React from 'react';
import {
  BrowserRouter as Router,
  Switch,
  Route,
  Link,
  Redirect,
  useHistory,
  useLocation
} from "react-router-dom";
import './App.css';
```

```
export default function AuthExample(){
  return(
    <Router>
      <div>
        <AuthButton/>
        <ul>
          <li>
            <Link to="/public">Public Page</Link>
          </li>
          <li>
            <Link to="/private">Private Page</Link>
          </li>
        </ul>

        <Switch>
          <Route path="/public">
            <PublicPage/>
          </Route>
          <Route path="/login">
            <LoginPage/>
          </Route>
          <PrivateRoute path="/private">
            <ProtectedPage/>
          </PrivateRoute>
        </Switch>
      </div>
    </Router>
  );
}
```



```
const fakeAuth = {
  isAuthenticated: false,
  authenticate(cb){
    fakeAuth.isAuthenticated = true;
    setTimeout(cb, 100);
  },
  signout(cb){
    fakeAuth.isAuthenticated = false;
    setTimeout(cb, 100);
  }
};

function AuthButton(){
  let history = useHistory();

  return fakeAuth.isAuthenticated ? (
    <p>
      Wellcome!{" "}
      <button
        onClick={() => {
          fakeAuth.signout(() => history.push("/"));
        }}>
        Sign out
      </button>
    </p>
  ) : (
    <p>You are no logged in.</p>
  );
}
```

```

function PrivateRoute({ children, ...res }) {
  return(
    <Route
      {...res}
      render={({ location }) =>
        fakeAuth.isAuthenticated ? (
          children
        ) : (
          <Redirect
            to={{
              pathname: "/login",
              state: { from: location }
            }}
          />
        )
      />
    );
}

function PublicPage(){
  return <h3>Public</h3>;
}

function ProtectedPage(){
  return <h3>Private</h3>
}

```

```

function LoginPage(){
  let history = useHistory();
  let location = useLocation();

  let {from} = location.state || {from: {pathname: "/"}};
  let login = () => {
    fakeAuth.authenticate(() => {
      history.replace(from);
    });
  };

  return(
    <div>
      <p>You must log in to view the page at {from.pathname}</p>
      <button onClick={login}>Log in</button>
    </div>
  )
}

```

2. Menjalankan server

ketika klik private page maka diharuskan untuk login terlebih dahulu, ketika sudah masuk dan klik private page kembali kita tidak perlu login kembali karena sudah berada dalam posisi logged.

You are no logged in.

- [Public Page](#)
- [Private Page](#)

You must log in to view the page at /private

[Log in](#)

Wellcome! [Sign out](#)

- [Public Page](#)
- [Private Page](#)

Private

Tugas

Buatlah sebuah situs marketplace(online shop) sederhana, dengan menerapkan konsep SPA (Single Page Application) React Router dimana beberapa menu nya jika di klik terdapat fitur “Nesting” dan “Redirect (Auth)”.

1. Membuat file js baru bernama appTugas.js kemudian membuat fungsi berikut

```
import React from 'react';
import 'bootstrap/dist/css/bootstrap.min.css';
import {
  BrowserRouter as Router,
  Switch,
  Route,
  Link,
  Redirect,
  useHistory,
  useLocation,
  useParams,
  useRouteMatch
} from "react-router-dom";
import {
  Button,
  Navbar,
  Card
} from 'react-bootstrap';
import './App.css';
```

```

export default Function AuthExample() {
  return (
    <Router>
      <div>
        <nav className="navbar navbar-expand-lg navbar-dark bg-dark">
          <a className="navbar-brand" href="#">MarketPlaceku</a>
          <button className="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav"
            <span className="navbar-toggler-icon"></span>
          </button>
          <div className="collapse navbar-collapse" id="navbarNav">
            <ul className="navbar-nav">
              <li className="nav-item active">
                <Link className="nav-link" to="/public">Public Page <span className="sr-only">(current)</span></Link>
              </li>
              <li className="nav-item">
                <Link className="nav-link" to="/products">Products</Link>
              </li>
              <li className="nav-item">
                <Link className="nav-link" to="/private">Log in</Link>
              </li>
              <li className="nav-item">
                <Link className="nav-link" to="/about">About</Link>
              </li>
            </ul>
          </div>
        </nav>
        <AuthButton />
      </div>
    </Router>
  );
}

```

```

      <Switch>
        {/* <Route path="/public">
          <PublicPage />
        </Route> */}
        <Route path="/login">
          <LoginPage />
        </Route>
        <Route path="/products">
          <Products />
        </Route>
        <Route path="/about">
          <About />
        </Route>
        <PrivateRoute path="/private">
          <Admin />
        </PrivateRoute>
      </Switch>
    </div>
  </Router>
);
}

```

```

function About(){
  return(
    <div>
      <h3>Created by : Satria Putra Sabana</h3>
      <h5>&copy;2020</h5>
    </div>
  );
}

const fakeAuth = {
  isAuthenticated: false,
  authenticate(cb) {
    fakeAuth.isAuthenticated = true;
    setTimeout(cb, 100);
  },
  signout(cb) {
    fakeAuth.isAuthenticated = false;
    setTimeout(cb, 100);
  }
};

function AuthButton() {
  let history = useHistory();

  return fakeAuth.isAuthenticated ? (
    <p>
      Selamat Datang dihalaman admin!{" "}
      <button
        onClick={() => {
          fakeAuth.signout(() => history.push("/"));
        }}>
        Sign out
      </button>
    </p>
  ) : (
    <p>You are no logged in.</p>
  );
}

```

```

function PrivateRoute({ children, ...res }) {
  return (
    <Route
      {...res}
      render={({ location }) =>
        fakeAuth.isAuthenticated ? (
          children
        ) : (
          <Redirect
            to={{
              pathname: "/login",
              state: { from: location }
            }}
          />
        )
      />
    />
  );
}

function PublicPage() {
  return <h3>Public</h3>;
}

function Admin() {
  return <h3>Wellcome Admin</h3>;
}

```

```

function Products() {
  let { path, url } = useRouteMatch();
  return (
    <div className="container">
      <div className="card">
        <div className="card-body">
          <h5 className="card-title">Products</h5>
          <p className="card-text">
            <ul>
              <li>
                <Link to={`${url}/BajuAnak`} >Baju Anak</Link>
              </li>
              <li>
                <Link to={`${url}/CelanaAnak`} >Celana Anak</Link>
              </li>
            </ul>
            <hr />
            <Switch>
              <Route exact path={path}>
                <h3></h3>
              </Route>
              <Route path={`${path}/:productId`} >
                <Product />
              </Route>
            </Switch>
          </p>
        </div>
      </div>
    </div>
  );
}

```

```

function Product() {
  let { productId } = useParams();
  if (productId === 'BajuAnak') {
    return (
      <div>
        <h3>Baju Anak</h3>
        <div class="card lebar">
          
          <div class="card-body">
            <h5 class="card-title">Baju Loreng Anak</h5>
            <p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>
            <a href="#" class="btn btn-primary">Beli</a>
          </div>
        </div>
        <div class="card lebar">
          
          <div class="card-body">
            <h5 class="card-title">Baju Lengan Panjang</h5>
            <p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>
            <a href="#" class="btn btn-primary">Beli</a>
          </div>
        </div>
      </div>
    );
  } else if (productId === 'CelanaAnak') {
    return (
      <div>
        <h3>Celana Anak</h3>
        <div class="card lebar">
          
          <div class="card-body">
            <h5 class="card-title">Celana Anak Pendek</h5>
            <p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>
            <a href="#" class="btn btn-primary">Beli</a>
          </div>
        </div>
        <div class="card lebar">
          
          <div class="card-body">

```

```

      <h3>{productId}</h3>
    </div>
  );
}

function LoginPage() {
  let history = useHistory();
  let location = useLocation();

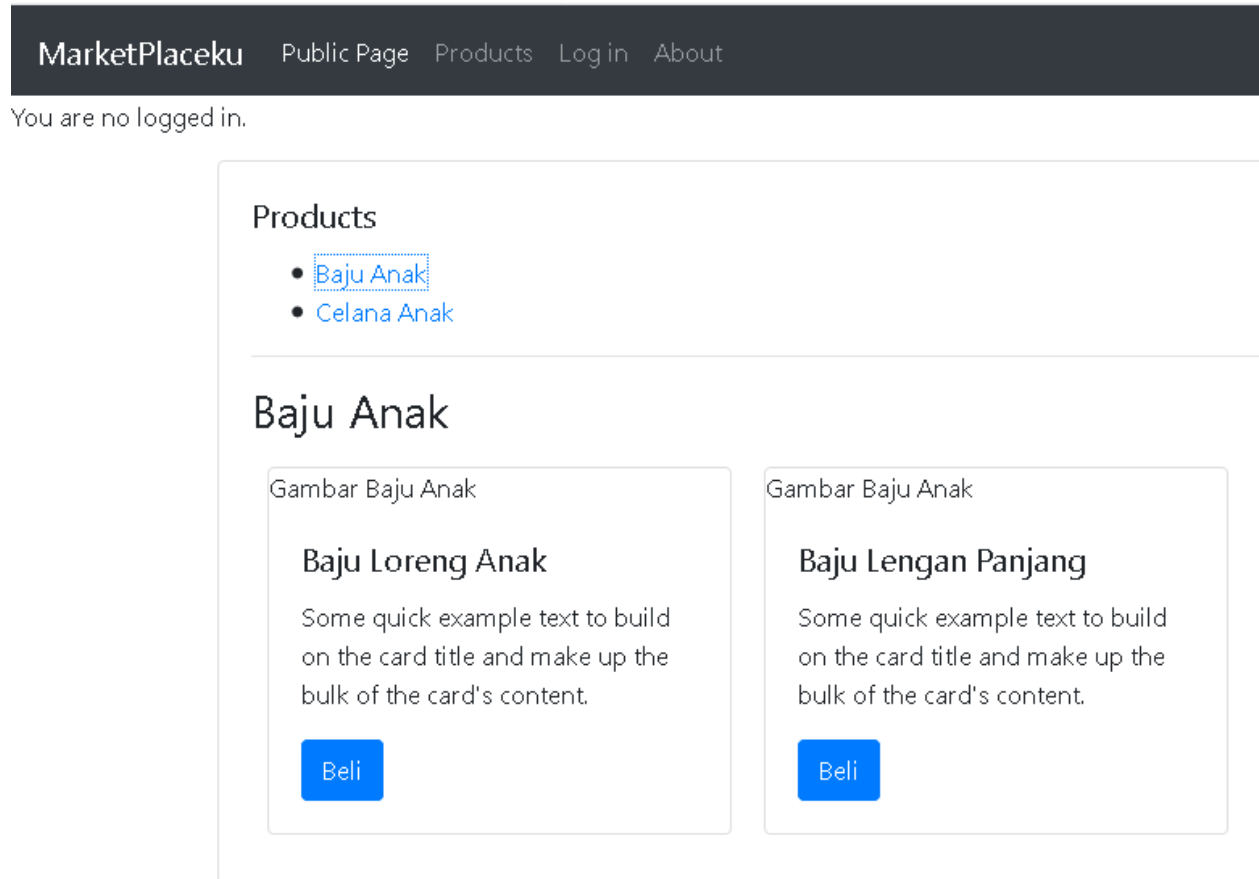
  let { from } = location.state || { from: { pathname: "/" } };
  let login = () => {
    fakeAuth.authenticate(() => {
      history.replace(from);
    });
  };

  return (
    <div>
      <p>You must log in to view the page</p>
      <button class="btn btn-primary" onClick={login}>Log in</button>
    </div>
  )
}

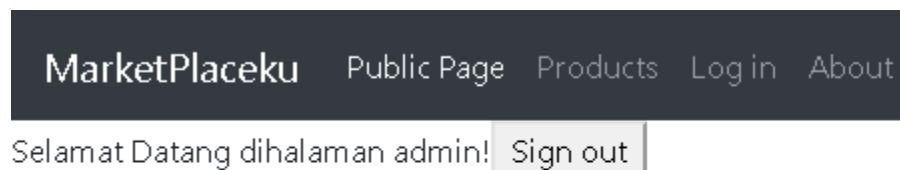
```


2. Menjalankan server

Ketika Baju anak diklik maka akan muncul dibawah menu dari baju anak begitu juga dengan celana anak



Tampilan setelah login



Wellcome Admin