



Adam Akbar

Frontend Engineer - Factor DAO

Fullstack Engineer - Fusion.Xyz

Typescript Product Engineer - Wander

Fullstack Engineer - Ruangguru

Education Background



2017-2021

Bachelor Degree

Informatics Engineer

Adam Akbar

4 Years experience as frontend Engineer

Introducing

React Router



Objektif sesi

- Memahami Router pada react
- Dapat melakukan http request dan handling file upload
- Dapat melakukan autentikasi dan otorisasi
- Dapat menggunakan google Oauth

React JS

- ☒ Routing Introduction
- ☐ HTTP Request
- ☐ File Processing
- ☐ Authentication
- ☐ Private Routes
- ☐ Implementation OAuth dengan Google

Apa itu React Router?

Library routing untuk React yang memungkinkan pengguna untuk menangani navigasi dan pembaruan tampilan di dalam aplikasi web React dengan mudah.



Beberapa hal yang bisa dipelajari dari react router:

- **Routing**
- **Navigation**
- **Nested Routing**

Instalasi react-router-dom



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

Contoh implementasi

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

const Home = () => {
  return <h1>Home</h1>;
};

const About = () => {
  return <h1>About</h1>;
};

const Contact = () => {
  return <h1>Contact</h1>;
};
```



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

function Home() {
  return (
    <div>
      <h1>Welcome to my app!</h1>
      <p>Click on the links above to navigate.</p>
    </div>
  );
}

function About() {
  return (
    <div>
      <h1>About me</h1>
      <p>I'm a web developer.</p>
    </div>
  );
}

function Contact() {
  return (
    <div>
      <h1>Contact me</h1>
      <p>You can reach me at example@example.com.</p>
    </div>
  );
}

function App() {
  return (
    <Router>
      <nav>
        <ul>
          <li><Link to="/">Home</Link></li>
          <li><Link to="/about">About</Link></li>
          <li><Link to="/contact">Contact</Link></li>
        </ul>
      </nav>

      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
        <Route path="/contact" element={<Contact />} />
      </Routes>
    </Router>
  );
}

export default App
```

React JS

- ☒ Routing Introduction
- ☒ HTTP Request
- ☐ File Processing
- ☐ Authentication
- ☐ Private Routes
- ☐ Implementation OAuth dengan Google

HTTP Request

React Router sendiri tidak secara langsung berhubungan dengan HTTP request. Namun, biasanya dalam aplikasi React yang kompleks, React Router akan digunakan bersama dengan HTTP request untuk mengambil atau mengirim data dari atau ke server.

Instalasi axios



```
npm i  
axios
```

Contoh implementasi



```
import React, { useState, useEffect } from 'react';  
import { useParams } from 'react-router-dom';  
import axios from 'axios';
```



```
const User = () => {
  const { id } = useParams();
  const [userData, setUserData] = useState(null);

  useEffect(() => {
    const getUserData = async () => {
      try {
        const response = await axios.get(`https://example.com/users/${id}`);
        setUserData(response.data);
      } catch (error) {
        console.error(error);
      }
    };

    getUserData();
  }, [id]);

  return (
    <div>
      {userData ? (
        <div>
          <h2>{userData.name}</h2>
          <p>{userData.email}</p>
        </div>
      ) : (
        <p>Loading user data...</p>
      )}
    </div>
  );
};

export default User;
```

React JS

- ☒ Routing Introduction
- ☒ HTTP Request
- ☒ File Processing
- ☐ Authentication
- ☐ Private Routes
- ☐ Implementation OAuth dengan Google

File Processing

React Router sendiri tidak secara langsung berkaitan dengan file processing, namun kita bisa menggunakan React Router untuk menavigasi ke halaman upload file

```
import React from 'react';
import { BrowserRouter as Router, Switch, Route } from 'react-router-dom';
import Home from './pages/Home';
import Upload from './pages/Upload';

function App() {
  return (
    <Router>
      <Switch>
        <Route exact path="/" component={Home} />
        <Route exact path="/upload" component={Upload} />
      </Switch>
    </Router>
  );
}

export default App;
```

menentukan dua Routing yang sesuai dengan halaman Home dan Upload

```
import React, { useCallback } from 'react';
import { useDropzone } from 'react-dropzone';

const Upload = () => {
  const onDrop = useCallback(acceptedFiles => {
    // proses file yang diunggah di sini
    console.log(acceptedFiles);
  }, []);

  const { getRootProps, getInputProps, isDragActive } = useDropzone({ onDrop });
  return (
    <div {...getRootProps()}>
      <input {...getInputProps()} />
      {isDragActive ? (
        <p>Drop the files here ...</p>
      ) : (
        <p>Drag 'n' drop some files here, or click to select files</p>
      )}
    </div>
  );
};

export default Upload;
```

menggunakan useDropzone dari react-dropzone untuk menentukan fungsi yang dijalankan saat file diunggah, dan menghubungkannya dengan elemen input

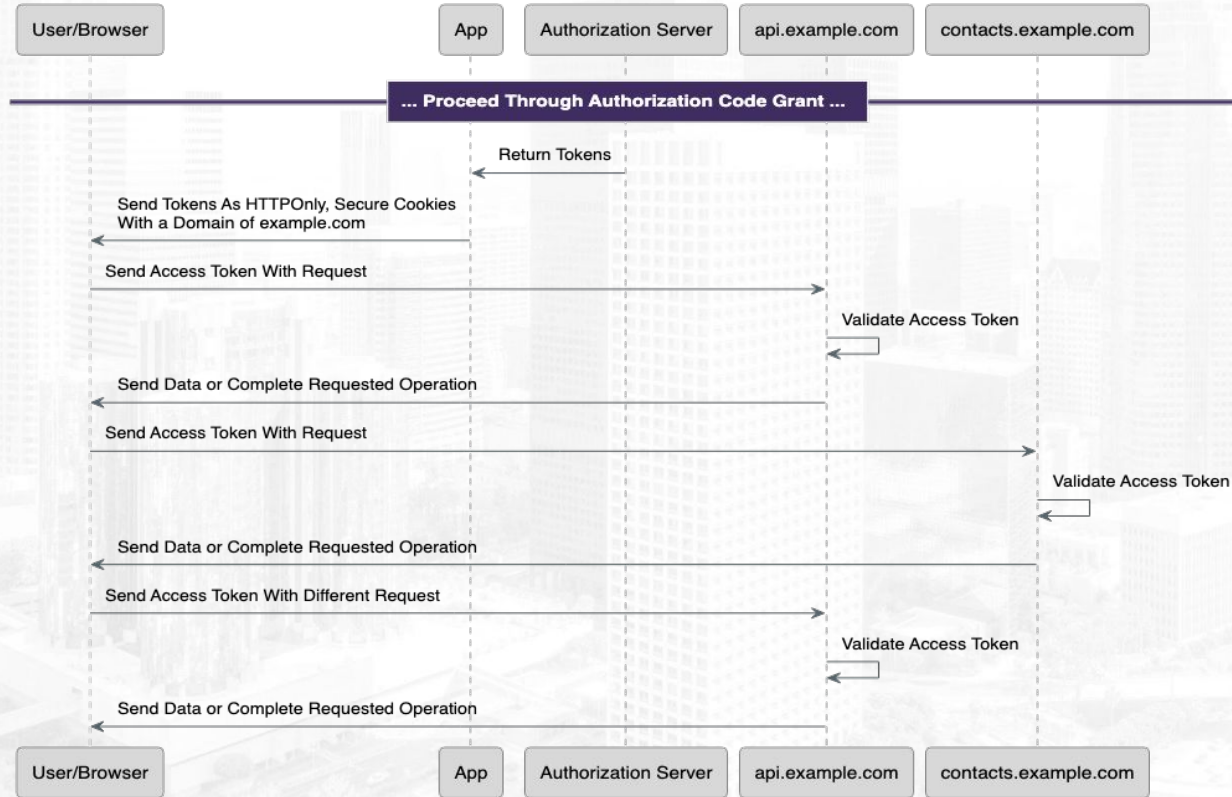
React JS

- ☒ Routing Introduction
- ☒ HTTP Request
- ☒ File Processing
- ☒ Authentication
- ☐ Private Routes
- ☐ Implementation OAuth dengan Google

Authentication

React Router dapat digunakan untuk mengatur navigasi pada aplikasi web yang memiliki fitur autentikasi, seperti halaman login dan halaman dashboard, dll.

Authentication adalah proses untuk memastikan identitas pengguna yang ingin mengakses suatu sistem atau aplikasi. Dalam konteks aplikasi web, authentication biasanya dilakukan dengan cara meminta pengguna untuk memasukkan username dan password yang sudah terdaftar.



```
import React, { useState } from 'react';
import { BrowserRouter as Router, Switch, Route, Redirect } from 'react-router-dom';
import Login from './pages/Login';
import Dashboard from './pages/Dashboard';

function App() {
  const [isAuthenticated, setIsAuthenticated] = useState(false);

  const handleLogin = () => {
    setIsAuthenticated(true);
  };

  const handleLogout = () => {
    setIsAuthenticated(false);
  };

  return (
    <Router>
      <Switch>
        <Route exact path="/login">
          {isAuthenticated ? (
            <Redirect to="/dashboard" />
          ) : (
            <Login handleLogin={handleLogin} />
          )}
        </Route>
        <Route path="/dashboard">
          {isAuthenticated ? (
            <Dashboard handleLogout={handleLogout} />
          ) : (
            <Redirect to="/login" />
          )}
        </Route>
      </Switch>
    </Router>
  );
}

export default App;
```


React JS

- ☒ Routing Introduction
- ☒ HTTP Request
- ☒ File Processing
- ☒ Authentication
- ☒ Private Routes
- ☐ Implementation OAuth dengan Google

Private Routes

React Router juga dapat digunakan untuk membuat halaman private yang hanya dapat diakses oleh pengguna yang sudah login atau ter-otentikasi.



```
import React from 'react';
import { Route, Redirect } from 'react-router-dom';

const PrivateRoute = ({ children, isAuthenticated, ...rest }) =>
{ return (
  <Route {...rest}>
    {isAuthenticated ? (
      children
    ) : (
      <Redirect to={{ pathname: '/login' }} />
    )}
  </Route>
);
};

export default PrivateRoute;
```

```
import React, { useState } from 'react';
import { BrowserRouter as Router, Switch, Route, Redirect } from 'react-router-dom';
import Login from './pages/Login';
import Dashboard from './pages/Dashboard';
import PrivateRoute from './components/PrivateRoute';

function App() {
  const [isAuthenticated, setIsAuthenticated] = useState(false);

  const handleLogin = () => {
    setIsAuthenticated(true);
  };

  const handleLogout = () => {
    setIsAuthenticated(false);
  };

  return (
    <Router>
      <Switch>
        <Route exact path="/login">
          {isAuthenticated ? (
            <Redirect to="/dashboard" />
          ) : (
            <Login handleLogin={handleLogin} />
          )}
        </Route>
        <PrivateRoute path="/dashboard" isAuthenticated={isAuthenticated}>
          <Dashboard handleLogout={handleLogout} />
        </PrivateRoute>
      </Switch>
    </Router>
  );
}

export default App;
```

React JS

- ✓ Routing Introduction
- ✓ HTTP Request
- ✓ File Processing
- ✓ Authentication
- ✓ Private Routes
- ✓ Implementation OAuth dengan Google

Implementation OAuth dengan Google

React Router juga bisa di integrasikan dengan Google OAuth untuk hal otentikasi

Instalasi react google login



```
npm i @react-oauth/google
```

Contoh implementasi



```
import React, { useState, useEffect } from 'react';  
import { BrowserRouter as Router, Switch, Route, Link, useHistory } from 'react-router-dom';  
import { GoogleLogin, GoogleLogout } from '@react-oauth/google';  
  
const CLIENT_ID = '<your-google-client-id>;
```



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

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

```
function App() {
  const [isLoggedIn, setIsLoggedIn] = useState(false);
  const history = useHistory();

  useEffect(() => {
    // Check if user is already logged in
    const token = localStorage.getItem('token');
    if (token) {
      setIsLoggedIn(true);
    }
  }, []);

  const onSuccess = (response) => {
    // Store token in local storage
    localStorage.setItem('token',
response.isLoggedIn ? token : null);

    // Navigate to dashboard
    history.push('/dashboard');
  };

  const onFailure = (response) => {
    console.error(response);
  };

  const onLogoutSuccess = () => {
    // Remove token from local storage
    localStorage.removeItem('token');
    setIsLoggedIn(false);

    // Navigate to login page
    history.push('/');
  };

  return (
    <Router>
      ...
    </Router>
  );
}

export default App;
```



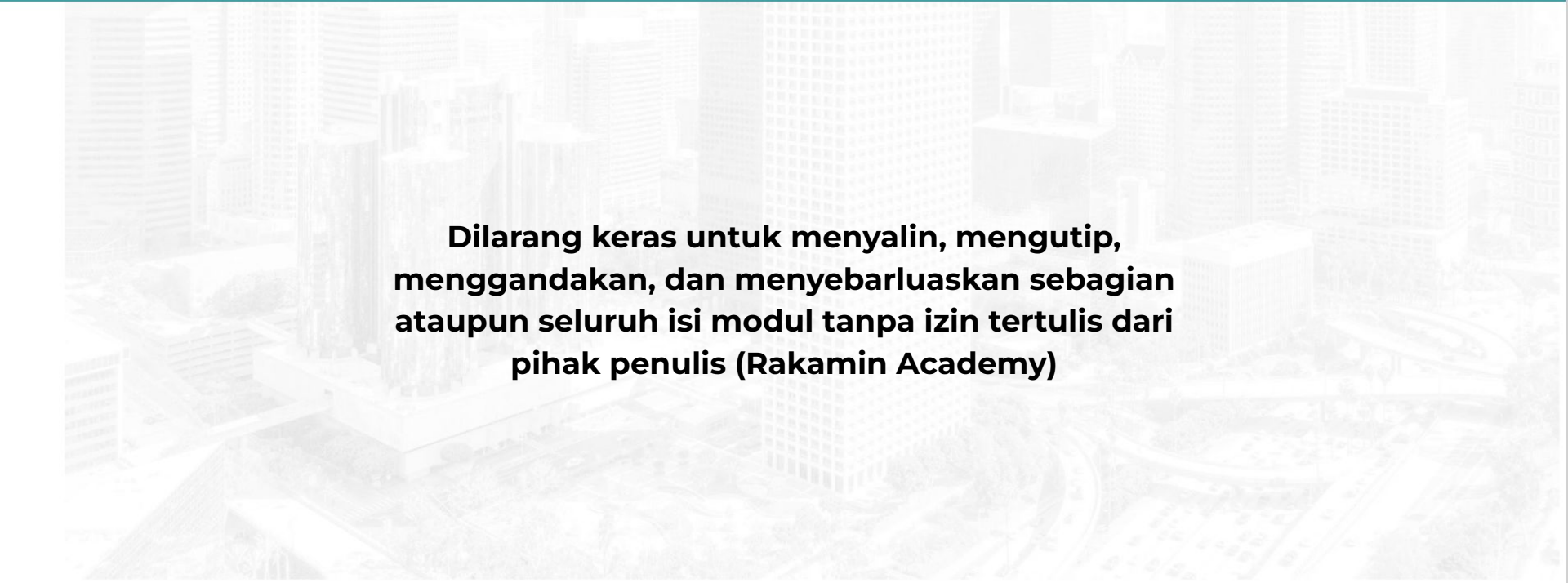
```
function App() {  
  ...  
  
  return (  
    <Router>  
      <div>  
        <nav>  
          <ul>  
            <li>  
              <Link to="/">Home</Link>  
            </li>  
            <li>  
              <Link to="/dashboard">Dashboard</Link>  
            </li>  
          </ul>  
          {isLoggedIn ? (  
            <GoogleLogout  
              clientId={CLIENT_ID}  
              buttonText="Logout"  
              onLogoutSuccess={onLogoutSuccess}  
            </>  
          ) : (  
            <GoogleLogin  
              clientId={CLIENT_ID}  
              buttonText="Login"  
              onSuccess={onSuccess}  
              onFailure={onFailure}  
              cookiePolicy={'single_host_origin'}  
              responseType="code,token"  
              isSignedIn={true}  
            </>  
          )}  
        </nav>  
        <Switch>  
          <Route path="/dashboard">  
            {isLoggedIn ? <Dashboard /> : <Redirect to="/" />  
          </Route>  
          <Route path="/">  
            <Home />  
          </Route>  
        </Switch>  
      </div>  
    </Router>  
  );  
}  
  
export default App;
```

Reference material

1. <https://fusionauth.io/learn/expert-advice/oauth/oauth-token-storage>
2. <https://blog.webdevsimplified.com/2022-07/react-router/>

Terima kasih!

Copyright Rakamin Academy

A faded, grayscale background image of a city skyline with various skyscrapers and buildings.

**Dilarang keras untuk menyalin, mengutip,
menggandakan, dan menyebarkan sebagian
ataupun seluruh isi modul tanpa izin tertulis dari
pihak penulis (Rakamin Academy)**

References

1. <https://fusionauth.io/learn/expert-advice/oauth/oauth-token-storage>
2. <https://blog.webdevsimplified.com/2022-07/react-router/>

Rancangan Subtopik

Di bawah ini merupakan list subtopik yang sudah dirancang. Namun masih sangat bisa dirubah dan disesuaikan dengan kebutuhan

<<Subtopic>>

Expected output

<<Competency>>



Homework:

Buat sebuah tampilan antarmuka sederhana yang dapat digunakan untuk

1. Memasukkan Nama
2. Memasukkan Usia
3. Tampilkan data user
 - a. Tampilkan Nama
 - b. Tampilkan Usia
 - c. Tampilkan jenis Usia
 - i. 0-5 = Balita
 - ii. 5-10 = Anak-Anak
 - iii. 10-20 = Remaja
 - iv. 20-50 = Dewasa
 - v. 50 - unlimited = Senja

Study Kasus

Kita akan membuat sebuah game tic-tac-toe sederhana.

Selesaikan boilerplate dari game tic-tac-toe sebagai berikut :

1. Buatlah state / variable yang nantinya akan digunakan untuk menyimpan data.
2. Akan ada 2 derived state :
 - a. nextValue (untuk menentukan value selanjutnya adalah "X" atau "O")
 - b. Winner (pemenang)
3. Menyelesaikan fungsi selectSquare
4. Menyelesaikan fungsi reset
5. Saat user refresh, pastikan data tidak berubah dan user bisa kembali bermain hingga tamat.

Hands on

<<Hands on>>

