UNIVERSITAS MIKROSKIL

Computer Laboratory Handbook

Pengembangan Aplikasi Web

TI2109

By:

Participant Handbook

Course Version: 2023

COURSE OVERVIEW

COURSE OVERVIEW

Mata kuliah ini, mahasiswa akan mempelajari tentang bagaimana membangun dan mengembangkan aplikasi web mulai dari front end, back-end server, aplikasi dan basis data sampai pembuatan RestAPI dengan memanfaatkan React Js, Node Js dan Express Js.

COURSE GOALS

Capaian Pembelajaran Lulusan yang dibebankan:

- Mahasiswa mampu memahami dan menerapkan konsep-konsep dasar dalam pengembangan web menggunakan React Js.
- Mahasiswa Mampu mengimplementasikan teknik routing dan styling dalam pengembangan web menggunakan React Js.
- Mahasiswa mampu membangun dan mengelola server menggunakan Express Js.
- Mahasiswa mampu mengintegrasikan API dalam pengembangan web menggunakan React Js.

COURSE OBJECTIVES

Capaian Pembelajaran Mata Kuliah:

- Mahasiswa mampu memahami dan menerapkan konsep-konsep ke dalam HTML.
- Mahasiswa mampu memilih element dan semantik yang sesuai pada HTML.
- Mahasiswa mampu menerapkan konsep tipografi dan multimedia pada halaman HTML.
- Mahasiswa mampu memilih CSS format yang sesuai untuk diterapkan ke dalam HTML.
- Mahasiswa mampu merencanakan penggunaan GIT untuk kolaborasi dalam proses pengembangan aplikasi.
- Mahasiswa mampu menerapkan konsep dari layout dan grid pada CSS.
- Mahasiswa mampu membangun sebuah halaman web yang responsive secara design dengan menggunakan media queries.
- Mahasiswa mampu menerapkan penggunaan dari list untuk pembuatan daftar pada halaman web.
- Mahasiswa mampu membangun tabel dan form untuk proses pendataan pada halaman web.
- Mahasiswa mampu menyusun web dengan menerapkan konsep javascript.

UNIT 5
ROUTING DENGAN REACT ROUTER

UNIT OVERVIEW

Pada minggu ini, kita akan menjelajahi salah satu aspek penting dalam pengembangan aplikasi web dengan React, yaitu "Routing dengan React Router." Dalam dunia aplikasi web modern, kemampuan untuk menavigasi antara berbagai halaman atau tampilan adalah kunci dalam menciptakan pengalaman pengguna yang mulus. Dengan React Router, kita akan belajar bagaimana mengelola navigasi dalam aplikasi React kita dengan mudah dan efisien. Kita akan membahas konsep dasar, mengenali perbedaan antara React Router Dom dan React Router Native, serta langkah-langkah instalasi dan konfigurasi. Selain itu, kita akan mempelajari cara membuat dan mengatur router untuk mengarahkan pengguna ke halaman-halaman yang sesuai. Dengan pemahaman tentang routing, Anda akan dapat membuat aplikasi web yang terstruktur dengan baik dan mudah dinavigasi, memberikan pengalaman pengguna yang lebih baik.

UNIT OBJECTIVES

Pada minggu ini, mahasiswa akan fokus pada praktik langsung dalam menerapkan konsep "react routing" pada aplikasi React. Berikut adalah capaian yang diharapkan setelah menyelesaikan topik ini:

1. Pengenalan Routing: Mahasiswa memahami konsep dasar routing dalam pengembangan aplikasi web & Mengapa routing penting dalam aplikasi React.
2. React Router:Mahasiswa mengenal terhadap React Router sebagai solusi routing di React & Memaham perbedaan antara React Router Dom dan React Router Native.
3. Instalasi dan Konfigurasi React Router Dom: Mahasiswa memahami langkah-langkah instalasi React Router Dom & Konfigurasi dasar untuk mulai menggunakan React Router.
4. Membuat dan Mengatur Router: Mahasiswa memahami cara membuat dan mengatur router di dalam aplikasi React& Penggunaan komponen <BrowserRouter> untuk membungkus aplikasi.
5. Routing Dasar: Mahasiswa dapat menggunakan komponen <Route> untuk mengaitkan URL dengan komponen & Penggunaan komponen <Link> untuk navigasi, serta Navigasi dengan fungsi navigate().

UNIT CONTENTS

PRE LAB

Pre lab, berisi pertanyaan mendasar terkait teori materi yang sedang diajarkan.

QUESTION

1. Apa yang dimaksud dengan routing dalam konteks pengembangan aplikasi web dan mengapa itu penting dalam pengembangan aplikasi React?
2. Apa perbedaan antara React Router Dom dan React Router Native?
3. Bagaimana cara menginstal dan mengkonfigurasi React Router Dom dalam sebuah proyek React?
4. Jelaskan perbedaan antara komponen <Route> dan komponen <Link>. Bagaimana Anda menggunakannya untuk navigasi di aplikasi React?
5. Bagaimana cara mengirimkan parameter melalui URL menggunakan React Router? Jelaskan konsep useParams() dalam hal ini.
6. Apa itu nested routing, dan bagaimana Anda menggunakannya dalam pengembangan aplikasi React?
7. Bagaimana Anda menangani kasus ketika URL yang dimasukkan pengguna tidak cocok dengan rute yang ada? Apa yang biasanya ditampilkan sebagai respons?
8. Mengapa penting untuk menyediakan dependensi pada useEffect? Apa yang akan terjadi jika Anda tidak melakukannya?
9. Bagaimana Anda mengelola efek samping (side effects) dan pembersihan (cleanup) dalam useEffect?
10. Apakah ada perbedaan antara penggunaan useEffect dan metode siklus hidup componentDidMount dan componentDidUpdate dalam kelas komponen?
11. Berikan contoh penggunaan praktis dari React Router dalam pengembangan aplikasi yang lebih kompleks. Bagaimana Anda mengatur navigasi dan mengganti konten dengan React Router dalam kasus tersebut?

CONTENT LESSON

CASE STUDY / PROJECT

PT XYZ adalah sebuah perusahaan yang ingin meningkatkan pengalaman pengguna di situs web mereka. Salah satu langkah yang ingin mereka ambil adalah membuat halaman yang dinamis untuk pengguna yang memiliki peran sebagai admin. Mereka ingin mengarahkan admin ke halaman dashboard admin saat mereka masuk.

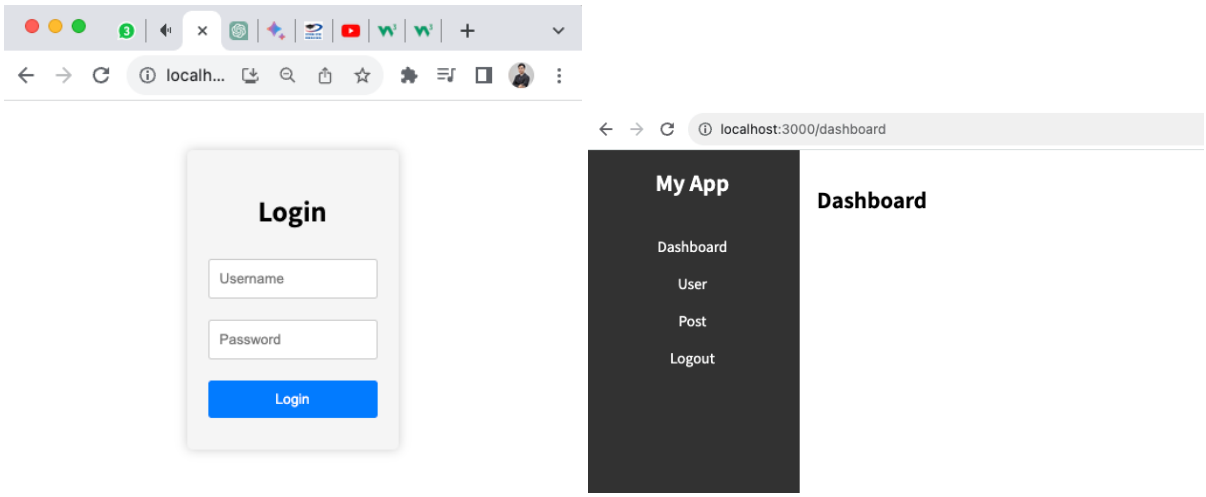
Tantangan

Tantangan utama adalah membuat halaman yang dinamis, yang berarti kontennya dapat berubah tergantung pada peran pengguna. Untuk ini, PT XYZ memutuskan untuk menggunakan React Router. Mereka ingin memastikan bahwa saat seorang admin masuk, mereka diarahkan ke halaman dashboard admin yang sesuai.

Solusi

Untuk mengatasi tantangan ini, PT XYZ memutuskan untuk menggunakan React Router, sebuah pustaka populer dalam pengembangan aplikasi React yang memungkinkan navigasi yang dinamis antara berbagai halaman.

Tampilan



SOLUTION

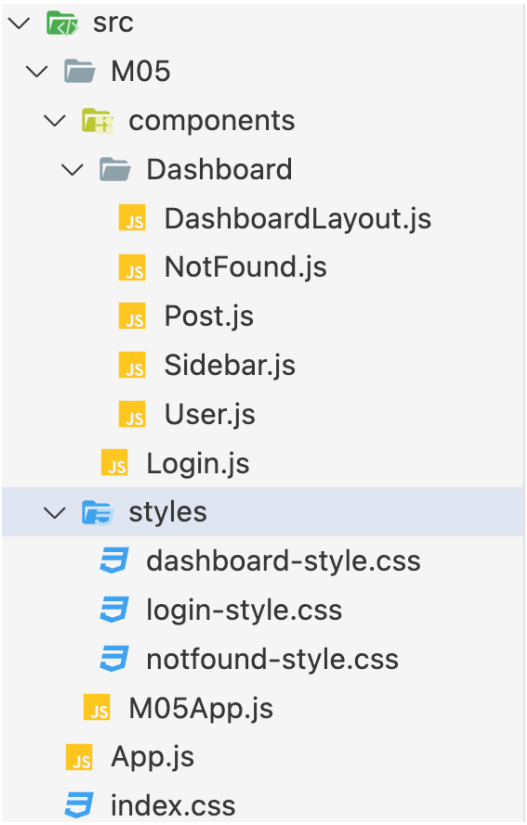
Berikut kita akan menerapkan react router secara bertahap.

1. Instalasi React Router Dom: PT XYZ menginstal paket "**react-router-dom**" dalam proyek React mereka. Ini memungkinkan mereka untuk menggunakan komponen dan fungsi yang diperlukan untuk menangani navigasi.
2. Konfigurasi Route: Mereka menambahkan komponen <BrowserRouter> di tingkat tertinggi aplikasi mereka, yang memberi tahu React Router bahwa aplikasi akan menggunakan routing. Kemudian, mereka mengkonfigurasi rute yang sesuai dengan peran pengguna. Dalam hal ini, mereka menentukan bahwa rute "/dashboard" akan mengarahkan admin ke halaman dashboard admin.
3. Menerapkan Link: PT XYZ menggunakan komponen <Link> untuk membuat tautan yang sesuai. Mereka menambahkan tautan "Dashboard" pada halaman utama situs web dan mengarahkannya ke "/dashboard".
4. Penggunaan Route: Di halaman dashboard admin, mereka menggunakan komponen <Route> untuk mengatur konten yang sesuai dengan rute. Ini memungkinkan mereka untuk menampilkan konten yang sesuai ketika admin mengakses "/dashboard".

INSTRUCTION

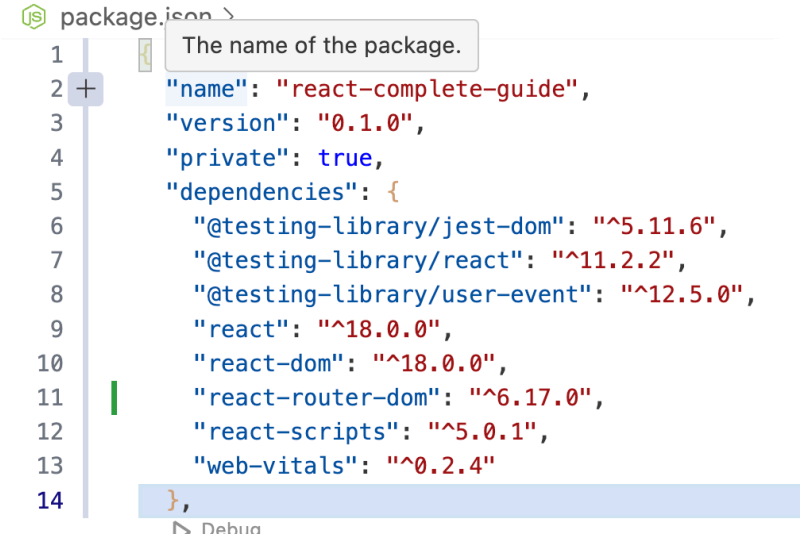
Silahkan ikuti langkah-langkah berikut untuk menyelesaikan permasalahan yang kalian buat:

1. Gunakan folder project react sebelumnya.
2. Buka project react menggunakan vscode dan jalankan Aplikasi React dengan **npm start**.
3. Perhatikan penjelasan dosen untuk struktur web app react.
4. Setelah semua siap, implementasikan struktur folder, dan file berikut :
   Buat struktur folder dan file berikut pada project anda :



5. Install react-router-dom melalui terminal (root project) menggunakan perintah **npm i react-router-dom**.

   Pada package.json perhatikan apakah telah terinstall



6. Konfigurasi BrowserRoutes pada index.js

```
import ReactDOM from "react-dom/client";

import "./index.css";
import App from "./App";
import { BrowserRouter } from "react-router-dom";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <BrowserRouter>
    <App />
  </BrowserRouter>
);

```

7. Buat halaman Login dan terapkan routes.
   a. Copy css (**login-style.css**) berikut untuk halaman login:

```
.login-container {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.login-form {
  background: #f5f5f5;
  padding: 20px;
  border-radius: 5px;
  box-shadow: 0px 0px 10px 0px rgba(0, 0, 0, 0.2);
}

.login-form h2 {
  text-align: center;
}

.login-form form {
  display: flex;
  flex-direction: column;
}
```

```
.login-form input {
  margin: 10px 0;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 3px;
}

.login-form button {
  margin: 10px 0;
  padding: 10px;
  background: #007bff;
  color: #fff;
  border: none;
  border-radius: 3px;
  cursor: pointer;
}

.login-form button:hover {
  background: #0056b3;
}
```

b. Lengkapi halaman login berikut:

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

import "../styles/login-style.css";
import { useNavigate } from "react-router-dom";

const Login = () => {
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");

  const navigate = useNavigate();

  const handleLogin = (e) => {
    e.preventDefault();
    // Contoh validasi login
    if (username === "admin" && password === "admin") {
      alert("Login berhasil");
      // Redirect ke halaman dashboard atau halaman berikutnya
      navigate("/dashboard");
    } else {
      alert("Login gagal");
    }
  };

  return (
    <div className="login-container">
```

```
      <div className="login-form">
        <h2>Login</h2>
        <form onSubmit={handleLogin}>
          <input
            type="text"
            placeholder="Username"
            value={username}
            onChange={(e) => setUsername(e.target.value)}
          />
          <input
            type="password"
            placeholder="Password"
            value={password}
            onChange={(e) => setPassword(e.target.value)}
          />
          <button type="submit">Login</button>
        </form>
      </div>
    </div>
  );
};

export default Login;
```

8. Atur route dan akses halaman login dengan terapkan M05App.js

```
import React from "react";

import Login from "./components/Login";
import { Routes, Route } from "react-router-dom";

const M05App = () => {
  return (
    <Routes>
      <Route path="/" element={<Login />} />
    </Routes>
  );
};

export default M05App;
```

9. Panggil M05App pada App.js

```
import M05App from "./M05/M05App";


function App() {
  return <M05App />;
}
export default App;
```

Lalu akses halaman utama aplikasi react, maka akan tampil halaman login.

10. Berikutnya, buat layout untuk dashboard dengan menerapkan pemisahan komponen sidebar, post dan user. Kita mulai dengan **sidebar.**

```
// Sidebar.js

import React from "react";
import { Link } from "react-router-dom";

const Sidebar = () => {
  return (
    <div className="sidebar">
      <div className="logo">My App</div>
      <nav className="nav-menu">
        <ul>
          <li>
            <Link to="/dashboard">Dashboard</Link>
          </li>
          <li>
            <Link to="/dashboard/user">User</Link>
          </li>
          <li>
            <Link to="/dashboard/post">Post</Link>
          </li>
          <li>
            <Link to="/logout">Logout</Link>
          </li>
        </ul>
      </nav>
    </div>
  );
};
export default Sidebar;
```

11. Berikutnya halaman Post.js

```
import React from 'react'


const Post = () => {
  return (
    <div>
      <h2>Post Page</h2>
      {/* Tambahkan konten halaman Post di sini */}
    </div>
  )
}

export default Post
```

12. Berikutnya halaman User.js

```
import React from "react";


const User = () => {
  return (
    <div>
      <h2>User Page</h2>
      <ul>
        <li>User 1</li>
        <li>User 2</li>
        <li>User 3</li>
        <li>User 4</li>
        <li>User 5</li>
      </ul>
    </div>
  );
};

export default User;
```

13. Berikut adalah style untuk halaman dashboard:

```
/* Dashboard.css */
.dashboard-container {
  display: flex;
  height: 100vh;
}

.sidebar {
  width: 250px;
  background: #333;
  color: #fff;
}

.sidebar .logo {
  text-align: center;
  padding: 20px 0;
  font-size: 1.5rem;
  font-weight: bold;
}

.nav-menu ul {
  list-style: none;
  padding: 0;
}
```

```
.nav-menu ul li {
  padding: 10px;
  text-align: center;
}

.nav-menu ul li a {
  color: #fff;
  text-decoration: none;
  display: block;
}

.main-content {
  flex: 1;
  padding: 20px;
}
```

14. Jika sudah, mari kita susun halaman dashboard sebagai berikut:

```
import React from "react";

import { Outlet } from "react-router-dom";
import "../../styles/dashboard-style.css";
import Sidebar from "./Sidebar";

const DashboardLayout = () => {
  return (
    <div className="dashboard-container">
      <Sidebar />
      <div className="main-content">
        <h2>Dashboard</h2>
        <Outlet />
      </div>
    </div>
  );
};

export default DashboardLayout;
```

Disini kita menggunakan Outlet dalam react. Outlet dalam konteks React Router digunakan untuk menentukan tempat di mana komponen rute anak akan dirender. Ini membantu dalam merancang tata letak yang dinamis di mana satu komponen berperan sebagai "kontainer" atau "wadah" untuk komponen rute anak yang berbeda.

15. Sekarang kita susun route untuk keseluruhan halaman website. Akses kembali M05App.js lalu lengkap sebagai berikut:

```jsx
import React from "react";

import Login from "./components/Login";
import DashboardLayout from "./components/Dashboard/DashboardLayout";
import { Routes, Route } from "react-router-dom";
import User from "./components/Dashboard/User";
import Post from "./components/Dashboard/Post";
import NotFound from "./components/Dashboard/NotFound";

const M05App = () => {
  return (
    <Routes>
      <Route path="/" element={<Login />} />
      <Route path="/dashboard/" element={<DashboardLayout />}>
          <Route path="user" element={<User />} />
          <Route path="post" element={<Post />} />
        </Route>
      <Route path="*" element={<NotFound />} />
    </Routes>
  );
};

export default M05App;
```

Kamu perlu handle halaman tidak tersedia rutenya dengan menggunakan path='*' maka kita perlu buat halman NotFound.js.

```jsx
import React from "react";

import '../../styles/notfound-style.css'

const NotFound = () => {
  return (
    <div className="not-found-container">
      <h1>404 - Page Not Found</h1>
      <p>The page you are looking for does not exist.</p>
    </div>
  );
};

export default NotFound;
```

Berikut adalah stylenya:

```
.not-found-container {

  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  height: 100vh;
  background-color: #f5f5f5;
}

h1 {
  font-size: 3rem;
  margin-bottom: 1rem;
  color: #333;
}

p {
  font-size: 1.2rem;
  color: #555;
}
```

16. Lihat hasil source code di atas dengan memastikan koding di atas telah **disimpan** dan aplikasi react telah dijalankan dengan **npm start**

EXERCISE

EXERCISE OBJECTIVES

Pada latihan ini, mahasiswa diharapkan mampu untuk:

- Menerapkan routing sederhana pada homa page website.

TASK 1: SETELAH BERHASIL MEMBUAT HALAMAN DASHBOARD, GUNAKAN BASE CODE DI ATAS LALU BUATLAH SEBUAH HOMEPAGE DENGAN ALUR (STYLE BEBAS, FOKUS PADA PENERAPAN ROUTES):

1. Halaman pertama yaitu homepage (berisi navigasi dan body)
2. Navigasi terdiri dari:
    a. Home (dengan body: H1 => 'welcome')
    b. About (dengan body: paragraf tentang diri anda),
    c. dan Login.
3. Jika klik halaman login maka akan berali ke halaman Login.
4. Pada halaman login buat kondisi untuk mengecek, jika username adalah 'admin' maka arahkan ke halaman dashboard admin, selain itu jika login dengan nama anda maka alihkan ke halaman homepage (dengan body: h1 => welcome 'nama-mu').

UNIVERSITAS
MIKROSKIL