

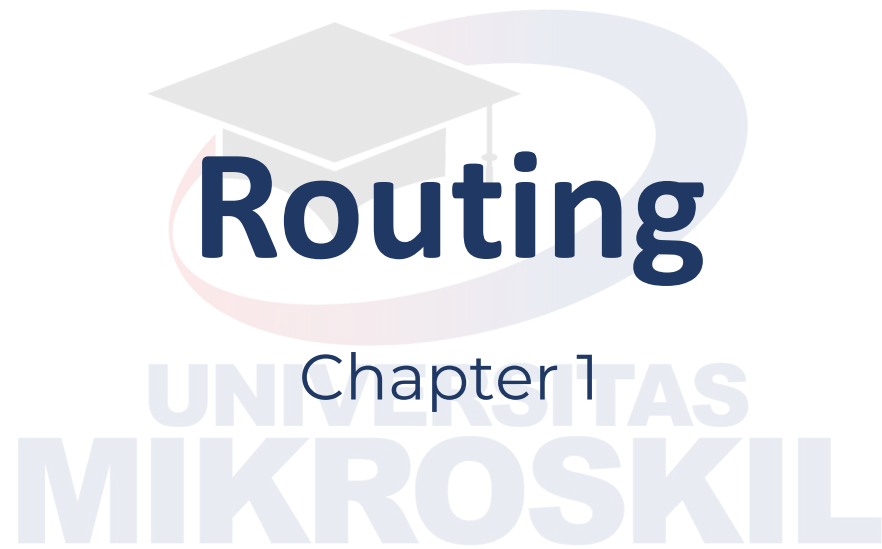
TI2104 – Pengembangan Aplikasi Web

Routing dengan React Router

- Pengenalan React Router
- Implementasi routing dalam aplikasi React Js
- Pengelolaan navigasi dan parameter URL

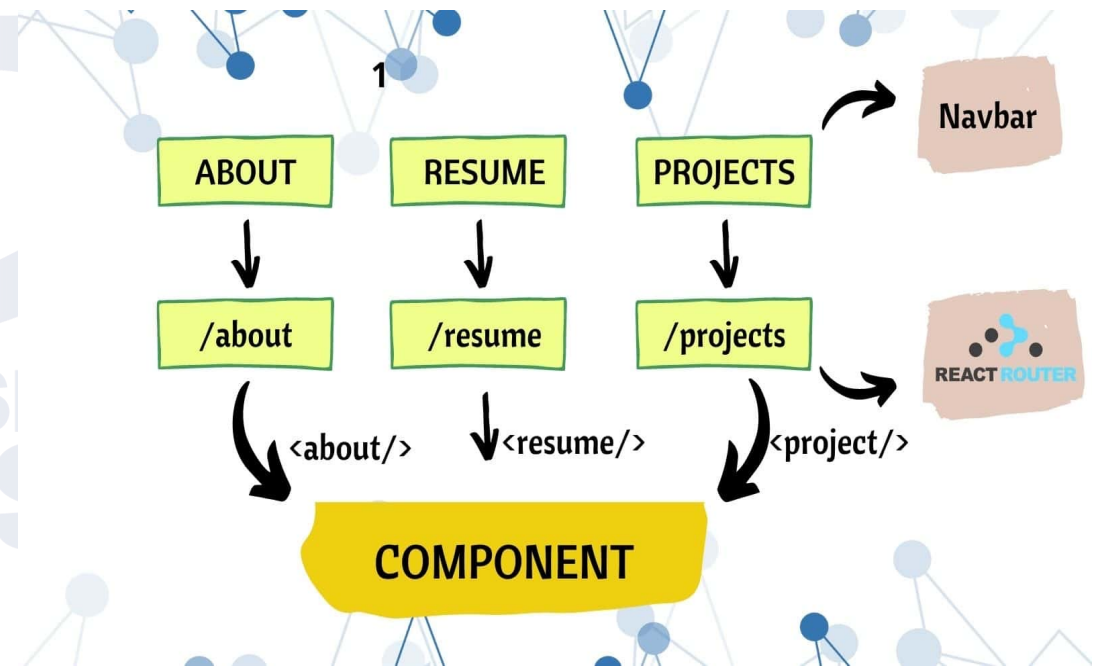
Strata - 1

Teknologi Informasi

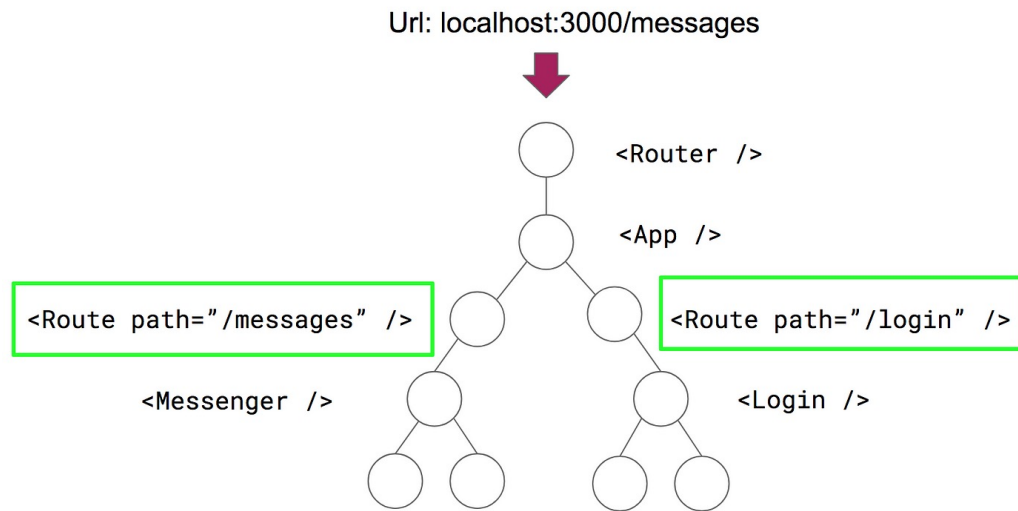


Routing dalam Aplikasi Web

- **Navigasi yang Terstruktur:** Routing membantu dalam mengatur tampilan dan konten yang spesifik untuk setiap bagian aplikasi.
- **URL yang Menyediakan Informasi:** Dalam aplikasi web, URL adalah cara untuk memberikan informasi kepada server dan pengguna. ***Routing yang baik membantu dalam menciptakan URL yang bersih, deskriptif, dan mudah dimengerti.***
- **Pemisahan Logika:** Dengan routing, logika aplikasi terpisah menjadi berbagai halaman atau komponen. Ini memudahkan pemeliharaan dan pengembangan aplikasi.
- **SEO (Search Engine Optimization):** Mesin pencari seperti Google lebih suka aplikasi yang memiliki struktur URL yang baik dan terstruktur.
- **Deep Navigation:** Routing memungkinkan navigasi yang lebih dalam ke dalam aplikasi, seperti navigasi ke detail item atau halaman lain. Ini sangat penting untuk aplikasi yang lebih kompleks.
- **Manajemen Status Aplikasi:** Routing dapat digunakan untuk mengelola status aplikasi, seperti mengatur kapan aplikasi berada di mode edit atau tampilan biasa.



Pengenalan React Router



React Router adalah pustaka khusus yang dibangun untuk mengelola routing dalam aplikasi React. Beberapa poin penting tentang React Router:

- **Declarative:** React Router mengadopsi pendekatan deklaratif yang sesuai dengan konsep dasar React. Anda mendefinisikan rute dan tindakan navigasi secara deklaratif dengan komponen React.
- **Komponen-based:** React Router merangkul prinsip komponen React. Ini berarti Anda akan bekerja dengan komponen-komponen seperti **<Route>**, **<Link>**, dan **<Switch>** **<Routes>** untuk mengelola routing.
- **Nested Routing:** React Router mendukung routing bertingkat (nested routing) sehingga Anda dapat membuat tampilan yang kompleks dengan berbagai tingkatan halaman.
- **Parameter dan Query:** Anda dapat dengan mudah menangani parameter dan query dalam URL dengan React Router.
- **Dinamis:** React Router dapat digunakan untuk membuat rute dinamis berdasarkan data dan parameter, memungkinkan Anda untuk membuat rute dengan kode.

React Router Dom

React Router adalah pustaka pihak ketiga yang **menyediakan fungsi dan komponen untuk mengatur navigasi dalam aplikasi React**. Jadi, Anda memerlukan **react-router-dom** atau **react-router-native**, tergantung pada apakah Anda membangun aplikasi web atau aplikasi mobile.

npm install react-router-dom

```
package.json > {} dependencies > react-dom
1 {
2   "name": "react-complete-guide",
3   "version": "0.1.0",
4   "private": true,
5   "dependencies": {
6     "@testing-library/jest-dom": "^5.11.6",
7     "@testing-library/react": "^11.2.2",
8     "@testing-library/user-event": "^12.5.0",
9     "react": "^18.0.0",
10    "react-dom": "^18.0.0",
11    "react-router-dom": "^6.16.0",
12    "react-scripts": "^5.0.1",
13    "web-vitals": "^0.2.4"
14  },
```

```
import React from 'react';
import {
  BrowserRouter,
  Routes,
  Route,
} from 'react-router-dom';
import Page1 from './pages/page1.js';
import Page2 from './pages/page2.js';
import Page3 from './pages/page3.js';

function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route index element={<Page1 />}/>
        <Route path="page2" element={<Page2 />}/>
        <Route path="page3/:id" element={<Page3 />}/>
      </Routes>
    </BrowserRouter>
  );
}
```

Membuat dan Mengatur Router

Perlu membuat router menggunakan salah satu jenis router yang sesuai, seperti **<BrowserRouter>** atau **<HashRouter>**.

Kemudian, definisikan rute-rute yang sesuai dengan URL dan komponen yang akan dirender saat URL cocok dengan rute tersebut.

Juga dapat menggunakan **<Link>** untuk membuat tautan antar rute.

****Pastikan bahwa komponen ini ada dalam hirarki komponen yang dibungkus dalam <BrowserRouter>.**

```
1 import React from "react";
2 import { BrowserRouter, Routes, Route, Link } from 'react-router-dom';
3 import About from "../pages/About";
4 import Home from "../pages/Home";
5 import Login from "../pages/Login";
6
7 const M05AppTest = () => {
8   return (
9     /* dengan menggunakan react-router-dom, kita dapat mengatur navigasi antar halaman */
10    <BrowserRouter>
11      <div>
12        <nav>
13          <ul>
14            <li>
15              /* untuk menggunakan react-router-dom, kita harus menggunakan link */
16              <Link to="/">Home</Link>
17            </li>
18            <li>
19              <Link to="/about">About</Link>
20            </li>
21            <li>
22              <Link to="/login">Login</Link>
23            </li>
24          </ul>
25        </nav>
26
27        /* dengan menggunakan routes, kita dapat menampilkan beberapa halaman */
28        <Routes>
29          <Route path="/" element={<Home />} />
30          <Route path="/about" element={<About />} />
31          <Route path="/login" element={<Login />} />
32        </Routes>
33      </div>
34    </BrowserRouter>
35  );
36 };
37
38 export default M05AppTest;
39
```


Routing Dasar

No	Routing	Fungsi	Contoh
1.	<BrowserRouter>	React Router menggunakan komponen <BrowserRouter> untuk mengelola navigasi di aplikasi. Ini biasanya ditempatkan di dalam file index.js atau App.js	<pre>import { BrowserRouter } from 'react-router-dom'; ReactDOM.render(<BrowserRouter> <App /> </BrowserRouter>, document.getElementById('root'));</pre>
2	<Router>	Digunakan untuk menangani routing di aplikasi React	<pre>/* dengan menggunakan routes, kita dapat menampilkan beberapa halaman */ <Routes> <Route path="/" element={<Home />} /> <Route path="/about" element={<About />} /> <Route path="/login" element={<Login />} /> </Routes></pre>
3	<Route>	Digunakan untuk mendefinisikan sebuah rute dalam aplikasi React	
4	<Link>	Digunakan untuk membuat tautan ke rute lain dalam aplikasi React	<pre> <Link to="/"><button>Home</button></Link> </pre>
5	<Navigate>	Digunakan untuk bernavigasi ke rute lain secara manual dalam aplikasi React	<pre>const navigate = useNavigate(); const handleClick = () => { navigate("/about"); } return (<nav> <Link to="/"><button>Home</button></Link> <button onClick={handleClick}>Profile</button> </pre>

Routing dengan Parameter

Routing dengan parameter memungkinkan **untuk mengirim data** melalui URL dan mengaksesnya di halaman yang dituju.

- Meneruskan Komponen Anak ke Route

```
<Route path="/product/:id" component={ProductDetail} />
```

untuk mengidentifikasi produk.

- Mengakses Parameter dengan useParams()

```
import { useParams } from 'react-router-dom';

function ProductDetail() {
  const { id } = useParams();

  // Gunakan id dalam logika Anda
}
```


Routing Bertingkat (Nested Routing)

Routing dengan parameter memungkinkan **untuk mengirim data** melalui URL dan mengaksesnya di halaman yang dituju.

Dalam contoh di atas, didefinisikan rute, seperti **element={<DashboardLayout />}**. Ini akan menunjukkan bahwa saat rute /dashboard diakses, **DashboardLayout** akan ditampilkan.

- Meneruskan Komponen Anak ke Route

```
<Router>
  <Routes>
    <Route path="/dashboard" element={<DashboardLayout />}>
      <Route index element={<DashboardHome />} />
      <Route path="profile" element={<Profile />} />
      <Route path="settings" element={<Settings />} />
    </Route>
  </Routes>
</Router>
```

(?) Cari Tahu

- Bagaimana Redirect dan NotFound dalam navigasi halaman?
- Apa itu Guarding Navigasi? Dan kapan penerapannya?
- Bagaimana Pengaturan Lanjutan dan Lazy Loading (Advance)



