

Pemrograman Web Modern

Insan Taufik, S.Kom., M.Kom.



React Router (Navigasi)

Sub-Materi

Pengenalan Routing di SPA.
Implementasi React Router
DOM (BrowserRouter, Routes,
Route, Link). Membuat
navigasi antar halaman.
Menggunakan path
parameters dan query
parameters.

Tujuan Pembelajaran (Capaian)

Mampu membuat struktur
navigasi multipage dalam
konteks SPA (Single Page
Application).

Apa itu Routing dalam SPA?

Single Page Application (SPA):

- Aplikasi web yang berjalan dalam satu halaman HTML
- Navigasi terjadi tanpa reload halaman
- Hanya konten yang berubah

Kebutuhan Routing:

- Mengelola tampilan berbeda berdasarkan URL
- Memungkinkan bookmark dan sharing URL
- Browser history (back/forward button)

Analog:

Seperti gedung dengan banyak ruangan - routing adalah peta yang menunjukkan cara berpindah ruangan.

React Router DOM

Package untuk Routing:

```
npm install react-router-dom
```

Komponen Inti:

- BrowserRouter: Container untuk routing
- Routes: Wrapper untuk route definitions
- Route: Mendefinisikan path dan component
- Link: Navigasi tanpa reload
- Navigate: Programmatic navigation

Setup Dasar Router

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

function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
        <Route path="/contact" element={<Contact />} />
      </Routes>
    </BrowserRouter>
  );
}
```

Penjelasan:

- BrowserRouter: Membungkus seluruh aplikasi
- Routes: Grup semua route definitions
- Route: Map path ke component
- element: Component yang akan dirender

Navigasi dengan Link

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

function Navigation() {
  return (
    <nav>
      <Link to="/">Home</Link>
      <Link to="/about">About</Link>
      <Link to="/contact">Contact</Link>
    </nav>
  );
}
```

Perbedaan Link vs Anchor Tag:

- : Reload halaman
- <Link to="/about">: Tidak reload, hanya update component

Path Parameters

```
// Route definition
<Route path="/user/:userId" element={<UserProfile />} />

// Component mengakses parameter
import { useParams } from 'react-router-dom';

function UserProfile() {
  const { userId } = useParams();
  return <h1>Profile User {userId}</h1>;
}
```

Kegunaan:

- Dynamic routes berdasarkan ID
- Contoh: /products/123,
/users/456

Query Parameters

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

function SearchPage() {
  const [searchParams, setSearchParams] = useSearchParams();
  const query = searchParams.get('q');

  const handleSearch = (term) => {
    setSearchParams({ q: term });
  };

  return (
    <div>
      <h1>Search: {query}</h1>
      { /* Search form */ }
    </div>
  );
}
```

Kegunaan:

- Filter dan pencarian
- Sorting options
- Pagination

Nested Routes

```
<Route path="/products" element={<ProductsLayout />}>
  <Route index element={<ProductList />} />
  <Route path=":productId" element={<ProductDetail />} />
  <Route path="category/:categoryId" element={<CategoryProducts />} />
</Route>

// ProductsLayout component
import { Outlet } from 'react-router-dom';

function ProductsLayout() {
  return (
    <div>
      <h1>Products</h1>
      <Outlet /> {/* Tempat nested routes dirender */}
    </div>
  );
}
```

Programmatic Navigation

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

function LoginForm() {
  const navigate = useNavigate();

  const handleLogin = async () => {
    // Login logic...
    navigate('/dashboard'); // Redirect setelah login
  };

  const handleGoBack = () => {
    navigate(-1); // Kembali ke halaman sebelumnya
  };

  return (
    // Form JSX
  );
}
```

Protected Routes & Error Handling

```
// Protected Route Component
function ProtectedRoute({ children }) {
  const isAuthenticated = useSelector(state => state.auth.isAuthenticated);

  if (!isAuthenticated) {
    return <Navigate to="/login" />;
  }

  return children;
}

// Usage
<Route
  path="/dashboard"
  element={
    <ProtectedRoute>
      <Dashboard />
    </ProtectedRoute>
  }
/>

// 404 Not Found
<Route path="*" element={<NotFound />} />
```

Yang telah dipelajari:

- ✓ Konsep routing dalam SPA
- ✓ Setup React Router DOM
- ✓ Navigasi dengan Link dan useNavigate
- ✓ Path parameters dan query parameters
- ✓ Nested routes dan layout
- ✓ Protected routes dan error handling