

## **LAPORAN AKHIR PRAKTIKUM**

Mata Praktikum : Pemrograman Web

Kelas : 3IA24

Praktikum ke- : 4

Tanggal : 2 Januari 2024

Materi : React JS Lanjutan

NPM : 50421704

Nama : Juan Samuel Christopher

Ketua Asisten : Noviyanto Ramadhanu

Nama Asisten :

Paraf Asisten :

Jumlah Lembar : 13



**LABORATORIUM TEKNIK INFORMATIKA**

**UNIVERSITAS GUNADARMA**

**2024**

## LISTING PROGRAM

- app.js

```
fe3 > src > JS App.js > ...
1  import "../App.css";
2  import { BrowserRouter, Routes, Route } from "react-router-dom";
3  import HomePage from "../pages/HomePage";
4  import DataMahasiswaPage from "../pages/DataMahasiswaPage";
5  import TambahDataPage from "../pages/TambahDataPage";
6
7  function App() {
8    return (
9      <BrowserRouter>
10      <Routes>
11        <Route path="/" element={<HomePage />} />
12        <Route path="/data_mhs" element={<DataMahasiswaPage />} />
13        <Route path="/tambah_data" element={<TambahDataPage />} />
14      </Routes>
15    </BrowserRouter>
16  );
17 }
18
19 export default App;
```

- Editdata.jsx

```
src > pages > EditDataPage.jsx > ...
1  import React from "react";
2  import FormEditData from "../components/FormEditData";
3  import Layout from "../Layout";
4
5  const EditDataPage = () => {
6    return (
7      <Layout>
8        <div className="flex w-full justify-center items-center">
9          <h1 className="text-3xl font-semibold mt-5">Edit Data Mahasiswa</h1>
10        </div>
11        <div className="flex mt-10 justify-center">
12          <FormEditData />
13        </div>
14      </Layout>
15    );
16  };
17
18  export default EditDataPage;
19
```

- Formeditdata.jsx

```
src > components > FormEditData.jsx > ...
1  import React, { useEffect, useState } from "react";
2  import axios from "axios";
3  import { useParams } from "react-router-dom";
4  import Notification from "../Notification";
5
6  function FormEditData() {
7    const [npm, setNpm] = useState("");
8    const [nama, setNama] = useState("");
9    const [kelas, setKelas] = useState("");
10   const [msg, setMsg] = useState(""); // Menyimpan pesan kesalahan atau sukses
11   const [isError, setIsError] = useState(false);
12   const { id } = useParams();
13
14   useEffect(() => {
15     const getData = async () => {
16       try {
17         const response = await axios.get(
18           `http://localhost/pweb/be/read_by_id.php/${id}`
19         );
20
21         // Inisialisasi nilai input dengan nilai dari server
22         setNpm(response.data.data.npm);
23         setNama(response.data.data.nama);
24         setKelas(response.data.data.kelas);
25       } catch (error) {
26         console.error("Error: " + error);
27       }
28     };
29     getData();
30   }, [id]);
31
32   const updateMahasiswa = async (e) => {
33     e.preventDefault();
34     try {
35       await axios.put(`http://localhost/pweb/be/update.php/${id}`, {
36         npm: npm,
37         nama: nama,
38         kelas: kelas,
39       });
40       setMsg("Data Berhasil Edit");
41       setIsError(false);
42     } catch (error) {
43       if (error.response) {
44         setMsg("Data Berhasil Edit");
45         setIsError(true);
46       }
47     }
48   };
49
50   return (
```

```

51 <div className="flex w-full justify-center items-center">
52 <div className="z-999">
53 <Notification message={msg} isError={isError} />
54 </div>
55 <div className="p-4 lg:w-1/2">
56 <Form>
57   onSubmit={updateMahasiswa}
58   className="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4"
59 >
60   <div className="mb-4">
61     <label>
62       className="block text-gray-700 text-sm font-bold mb-2"
63       htmlFor="npm"
64     >
65       NPM
66     </label>
67     <input
68       className="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"
69       id="npm"
70       type="text"
71       placeholder="NPM"
72       value={npm || ""} // Menambahkan nilai awal
73       onChange={e => setNpm(e.target.value)}
74     />
75   </div>
76   <div className="mb-4">
77     <label>
78       className="block text-gray-700 text-sm font-bold mb-2"
79       htmlFor="nama"
80     >
81       Nama
82     </label>
83     <input
84       className="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"
85       id="nama"
86       type="text"
87       placeholder="Nama"
88       value={nama || ""} // Menambahkan nilai awal
89       onChange={e => setName(e.target.value)}
90     />
91   </div>
92   <div className="mb-4">
93     <label>
94       className="block text-gray-700 text-sm font-bold mb-2"
95       htmlFor="kelas"
96     >
97       Kelas
98     </label>
99     <input
100       className="shadow appearance-none border rounded w-full py-2 px-3 text-gray-700 leading-tight focus:outline-none focus:shadow-outline"
101       id="kelas"
102       type="text"
103       placeholder="Kelas"
104       value={kelas || ""} // Menambahkan nilai awal
105       onChange={e => setKelas(e.target.value)}

```

```

106     />
107   </div>
108 </div>
109 <div className="flex items-center justify-between">
110 <button
111   className="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded focus:outline-none focus:shadow-outline"
112   type="submit"
113 >
114   Edit Data
115 </button>
116 </div>
117 </form>
118 </div>
119 </div>
120 }
121
122 export default FormEditData;
123

```



## LOGIKA PROGRAM

Listing program merupakan susunan dari beberapa struktur data / computer codes, Selanjutnya, susunan tersebut dapat disusun menjadi semacam perintah programming yang digunakan dalam menyusun sebuah perintah software, berikut adalah Langkah dan logika dari pengerjaan ACT4

- **app.js**

```
fe3 > src > JS App.js > ...
1  import './App.css';
2  import { BrowserRouter, Routes, Route } from 'react-router-dom';
3  import HomePage from './pages/HomePage';
4  import DataMahasiswaPage from './pages/DataMahasiswaPage';
5  import TambahDataPage from './pages/TambahDataPage';
6
7  function App() {
8    return (
9      <BrowserRouter>
10     <Routes>
11       <Route path="/" element={<HomePage />} />
12       <Route path="/data_mhs" element={<DataMahasiswaPage />} />
13       <Route path="/tambah_data" element={<TambahDataPage />} />
14     </Routes>
15   </BrowserRouter>
16 );
17 }
18
19 export default App;
```

Blok program di atas menciptakan sebuah aplikasi web menggunakan React dan React Router untuk mengelola navigasi halaman. Dengan mengimpor modul-modul yang diperlukan, termasuk `./App.css` untuk gaya khusus, program tersebut mendefinisikan fungsi `App` sebagai komponen utama. Di dalamnya, `<BrowserRouter>` digunakan untuk manajemen navigasi berbasis browser, sementara `<Routes>` dan `<Route>` digunakan untuk menentukan rute-rute aplikasi dan komponen yang akan dirender untuk setiap rute.

Aplikasi ini memiliki tiga rute: `/` untuk halaman utama, `/data_mhs` untuk halaman data mahasiswa, dan `/tambah_data` untuk halaman penambahan data. Dengan demikian, saat pengguna menjelajahi aplikasi, React Router akan memastikan bahwa komponen yang sesuai dengan path yang diakses akan ditampilkan, memberikan tata letak yang terstruktur. Dengan

pendekatan ini, program berhasil membangun pengalaman pengguna yang terorganisir dan mudah dipahami melalui penggunaan React dan React Router.

- **Editdata.jsx**

```
src > pages > EditDataPage.jsx > ...
1  import React from "react";
2  import FormEditData from "../components/FormEditData";
3  import Layout from "../Layout";
4
5  const EditDataPage = () => {
6    return (
7      <Layout>
8        <div className="flex w-full justify-center items-center">
9          <h1 className="text-3xl font-semibold mt-5">Edit Data Mahasiswa</h1>
10        </div>
11        <div className="flex mt-10 justify-center">
12          <FormEditData />
13        </div>
14      </Layout>
15    );
16  };
17
18  export default EditDataPage;
19
```

Blok program diatas adalah untuk merender halaman untuk mengedit data mahasiswa. Program ini menggunakan komponen-komponen React untuk mengatur tata letak. Komponen EditDataPage terdiri dari dua bagian utama yang ditempatkan dalam layout umum (Layout component).

Bagian pertama berisi wadah flex dengan konten yang diatur ke tengah, menampilkan judul "Edit Data Player" dengan gaya tertentu untuk teksnya.

Bagian kedua adalah wadah flex lainnya yang menempatkan komponen FormEditData di tengah. Komponen formulir ini kemungkinan berisi bidang dan elemen input untuk mengedit data mahasiswa

- **Formeditdata.jsx**

```
src > components > FormEditData.jsx > ...
1  import React, { useEffect, useState } from "react";
2  import axios from "axios";
3  import { useParams } from "react-router-dom";
4  import Notification from "../Notification";
5
6  function FormEditData() {
7      const [npm, setNpm] = useState("");
8      const [nama, setNama] = useState("");
9      const [kelas, setKelas] = useState("");
10     const [msg, setMsg] = useState(""); // Menyimpan pesan kesalahan atau sukses
11     const [isError, setIsError] = useState(false);
12     const { id } = useParams();
13
14     useEffect(() => {
15         const getData = async () => {
16             try {
17                 const response = await axios.get(
18                     `http://localhost/pweb/be/read_by_id.php/${id}`
19                 );
20
21                 // Inisialisasi nilai input dengan nilai dari server
22                 setNpm(response.data.data.npm);
23                 setNama(response.data.data.nama);
24                 setKelas(response.data.data.kelas);
25             } catch (error) {
26                 console.error("Error: " + error);
27             }
28         };
29         getData();
30     }, [id]);
31
32     const updateMahasiswa = async (e) => {
33         e.preventDefault();
34         try {
35             await axios.put(`http://localhost/pweb/be/update.php/${id}`, {
36                 npm: npm,
37                 nama: nama,
38                 kelas: kelas,
39             });
40             setMsg("Data Berhasil Edit");
41             setIsError(false);
42         } catch (error) {
43             if (error.response) {
44                 setMsg("Data Berhasil Edit");
45                 setIsError(true);
46             }
47         }
48     };
49
50     return (
```

Blok program di atas adalah komponen React yang disebut `FormEditData`. Komponen ini bertanggung jawab untuk merender formulir yang memungkinkan pengguna mengedit data seorang mahasiswa. Inisialisasi nilai formulir didasarkan pada data mahasiswa yang diperoleh dari server melalui permintaan HTTP GET saat komponen ini dimuat.



Dalam komponen ini, terdapat beberapa state yang digunakan untuk menyimpan data mahasiswa (NPM, nama, dan kelas), pesan yang akan ditampilkan kepada pengguna (seperti pesan kesalahan atau pesan sukses setelah pengeditan data), dan status error. Penggunaan `useEffect` digunakan untuk mendapatkan data mahasiswa awal dari server saat komponen ini dimuat, dan `useState` digunakan untuk mengelola perubahan pada formulir.

Selanjutnya, terdapat fungsi `updateMahasiswa` yang dijalankan saat formulir disubmit. Fungsi ini mengirimkan permintaan HTTP PUT ke server untuk memperbarui data mahasiswa. Jika permintaan berhasil, pesan sukses ditetapkan, sedangkan jika terjadi kesalahan, pesan kesalahan ditetapkan dan status error diatur sesuai.

Komponen ini juga menggunakan komponen `Notification` untuk menampilkan pesan kepada pengguna, dan menggunakan desain tata letak dan gaya yang ditentukan menggunakan Tailwind CSS. Keseluruhan, `FormEditData` menyediakan antarmuka yang intuitif untuk mengedit data mahasiswa dengan responsif dan tata letak yang estetik.

- **DataMahasiswaPage.jsx**

```
fe3 > src > pages > DataMahasiswaPage.jsx > DataMahasiswaPage
1  import React, { useEffect, useState } from "react";
2  import Layout from "../Layout";
3  import axios from "axios";
4  import Notification from "../components/Notification";
5
6  const DataMahasiswaPage = () => {
7    const [mahasiswaData, setMahasiswaData] = useState([]);
8    const [msg, setMsg] = useState(""); // Menyimpan pesan kesalahan atau sukses
9    const [isError, setIsError] = useState(false);
10
11    useEffect(() => {
12      getData();
13    }, []);
14
15    const getData = async () => {
16      try {
17        const response = await axios.get("http://localhost/pweb/be/read.php");
18        if (response.data.data.length >= 0) {
19          setMahasiswaData(response.data.data);
20          console.log(response.data.data);
21        }
22      } catch (error) {
23        console.error("Error: " + error);
24      }
25    };
26
27    const deleteMhs = async (mhsId) => {
28      try {
29        await axios.delete(`http://localhost/pweb/be/delete.php/${mhsId}`);
30        setMsg("Data Delete Success");
31        setIsError(false);
32      } catch (error) {
33        setMsg("Data Gagal Edit");
34        setIsError(false);
35      }
36    };
37
38    // Memperbarui updateMemoryList
```

Terakhir pada blok program di atas adalah bagian dari halaman "Data Mahasiswa" dalam sebuah aplikasi web berbasis React. Pertama-tama, komponen `DataMahasiswaPage` menggunakan `useState` dan `useEffect` dari React untuk mengelola state lokal dan efek samping saat komponen ini dirender.

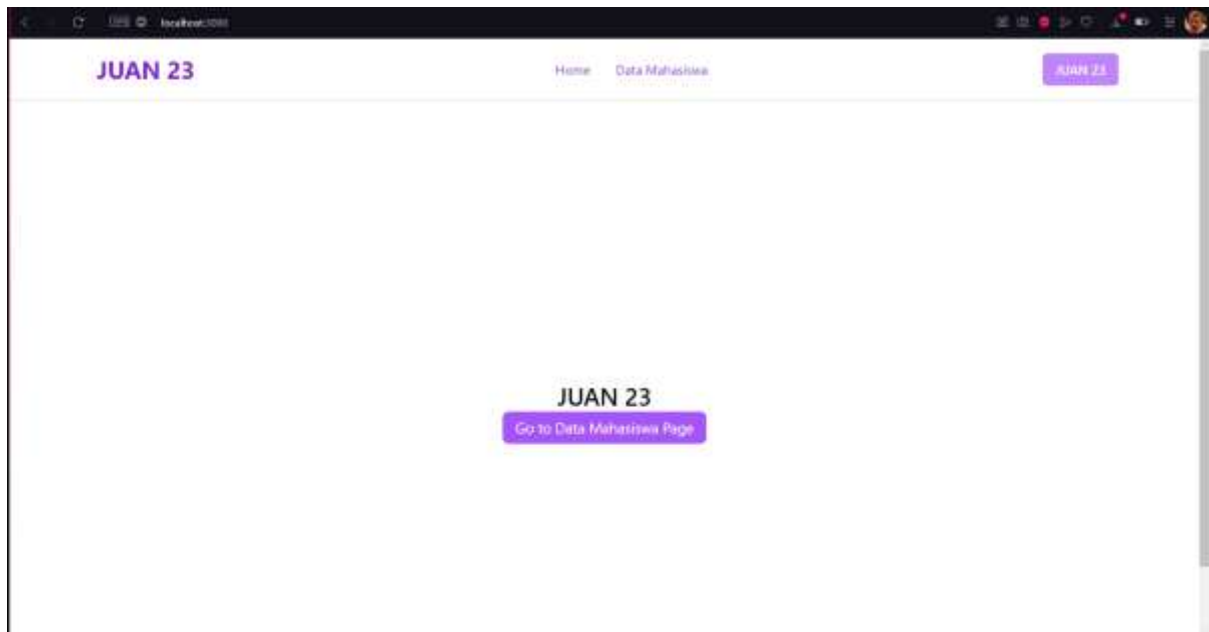
Dalam `useEffect`, fungsi `getData` dipanggil untuk mengambil data mahasiswa dari server menggunakan `Axios`. Hasil respons dari server digunakan untuk mengupdate state `mahasiswaData`, yang berisi array data mahasiswa. Jika data berhasil diambil, array tersebut akan digunakan untuk merender tabel yang menampilkan informasi mahasiswa, termasuk NPM, nama, kelas, dan opsi untuk menghapus data.

Selanjutnya, terdapat fungsi `deleteMhs` yang menggunakan `Axios` untuk mengirim permintaan DELETE ke server berdasarkan ID mahasiswa. Setelah operasi berhasil atau gagal, pesan hasilnya akan ditampilkan melalui state `msg` dan `isError`, yang selanjutnya dirender menggunakan komponen `Notification`.

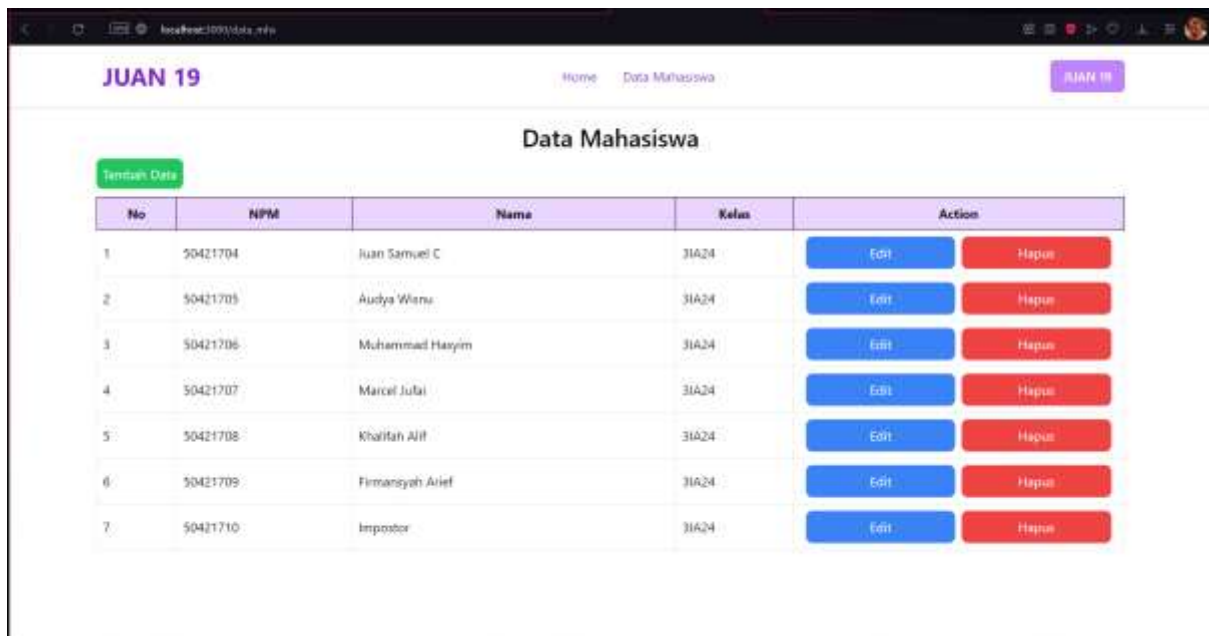
Tata letak halaman ini diatur menggunakan komponen `Layout`, dan terdapat tombol "Tambah Data" yang mengarahkan pengguna ke halaman penambahan data baru. Tabel ditampilkan dengan menggunakan elemen `<table>`, `<thead>`, dan `<tbody>`, serta menggunakan kondisi untuk menangani apakah terdapat data mahasiswa yang dapat ditampilkan atau tidak. Keseluruhannya, blok program ini memberikan pengalaman pengguna yang terstruktur dan informatif terkait data mahasiswa.

## OUTPUT PROGRAM

- Homepage



- Data Mahasiswa Page



- Edit Data

JUAN 19

HomeData Mahasiswa

JUAN 19

Data Berhasil Edit

Edit Data Mahasiswa

NPM

50421704

Nama

Juan Samuel Christopher

Kelas

BA24

Edit Data

- Hapus Data

JUAN 19

HomeData Mahasiswa

JUAN 19

Data Delete Success

Tambah Data

Data Mahasiswa

No	NPM	Nama	Kelas	Action	
1	50421704	Juan Samuel Christopher	BA24	Edit	Hapus
2	50421705	Audya Witanu	BA24	Edit	Hapus
3	50421706	Muhammad Haqim	BA24	Edit	Hapus
4	50421707	Marcel Jufai	BA24	Edit	Hapus
5	50421708	Khalifah Alf	BA24	Edit	Hapus
6	50421709	Firmansyah Anief	BA24	Edit	Hapus
7	50421710	Impostor	BA24	Edit	Hapus

- Data akhir mahasiswa

JUAN 19

Home Data Mahasiswa

JUAN 19

Tambah Data

No	NPM	Nama	Kelas	Action	
1	50421704	Juan Samuel Christopher	3A24	Edit	Hapus
2	50421705	Audyo Wino	3A24	Edit	Hapus
3	50421706	Muhammad Haryim	3A24	Edit	Hapus
4	50421707	Mercel Jufai	3A24	Edit	Hapus
5	50421708	Khulifah Alf	3A24	Edit	Hapus
6	50421709	Fitriansyah Arief	3A24	Edit	Hapus