

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Pemrograman Web

Kelas : 3IA24

Praktikum ke- : 3

Tanggal : 5 Desember 2023

Materi : Read, Creater, Axios, Getter, & Setter

NPM : 50421704

Nama : Juan Samuel Christopher

Ketua Asisten : Noviyanto Ramadhanu

Nama Asisten :

Paraf Asisten :

Jumlah Lembar : 13



LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2023

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;
```

- **FormTambahData.jsx**

- **TambahDataPage.jsx**

```
fe3 > src > pages > ✎ TambahDataPage.jsx > ...
1  import React from "react";
2  import FormTambahData from "../components/FormTambahData";
3  import Layout from "../Layout";
4
5  const TambahDataPage = ({message, Error}) => {
6    return (
7      <Layout>
8
9        <div className="flex w-full justify-center items-center">
10         <h1 className="text-3xl font-semibold mt-5">Tambah Data Mahasiswa</h1>
11       </div>
12       <div className="flex mt-10 justify-center">
13         <FormTambahData />
14       </div>
15     </Layout>
16   );
17 };
18
19 export default TambahDataPage;
```

- **Notification.jsx**

```
fe3 > src > components > ✎ Notification.jsx > ...
1  import React, { useState, useEffect } from "react";
2
3  function Notification({ message, isError }) {
4    const [isVisible, setIsVisible] = useState(false);
5
6    useEffect(() => {
7      if (message) {
8        setIsVisible(true);
9        setTimeout(() => {
10          setIsVisible(false);
11          // Memperbarui halaman setelah pesan muncul selama 2 detik
12          window.location.reload();
13        }, 2000);
14      }
15    }, [message]);
16
17    const notificationClass = isError ? "bg-red-500" : "bg-green-500";
18    return (
19      <div>
20        {isVisible && (
21          <div
22            className={`text-white z-50 fixed top-0 right-0 mt-20 mr-24 w-48 p-3 rounded ${notificationClass}`}
23          >
24            {message}
25          </div>
26        )}
27      </div>
28    );
29  };
30
31
32  export default Notification;
```


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 ACT3

- **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.

- **FormTambahData.jsx**

```
fe3 > src > components > FormTambahData.jsx > FormTambahData > saveMahasiswa
1  import React, { useState } from "react";
2  import axios from "axios";
3  import Notification from "../Notification";
4
5  function FormTambahData() {
6    const [npm, setNpm] = useState("");
7    const [nama, setNama] = useState("");
8    const [kelas, setKelas] = useState("");
9    const [msg, setMsg] = useState(""); // Menyimpan pesan kesalahan atau sukses
10   const [isError, setIsError] = useState(false);
11
12   const saveMahasiswa = async (e) => {
13     e.preventDefault();
14
15     const formData = new FormData();
16     formData.append("npm", npm);
17     formData.append("nama", nama);
18     formData.append("kelas", kelas);
19
20     try {
21       await axios.post("http://localhost/pweb/be/create.php", formData);
22       setMsg("Data Berhasil Ditambah");
23       setIsError(false);
24     } catch (error) {
25       if (error.response) {
26         setMsg("Data Gagal Ditambah");
27         setIsError(true);
28       }
29     }
30   };
31 }
```

Pada blok program diatas ini menggunakan modul React dan Hooks React, seperti useState, yang memungkinkan penggunaan komponen dan state lokal dalam React. Axios diimpor untuk melakukan permintaan HTTP, khususnya untuk mengirim data formulir ke server. Selain itu, komponen Notification diimpor untuk kemungkinan penggunaan dalam menampilkan pesan terkait hasil operasi formulir kepada pengguna.

Dalam komponen fungsi FormTambahData, dibuat beberapa state lokal menggunakan useState, antara lain 'npm', 'nama', 'kelas', 'msg', dan 'isError'. State ini akan menyimpan nilai dari input formulir serta pesan yang akan ditampilkan kepada pengguna terkait hasil operasi formulir. Fungsi saveMahasiswa sebagai event handler diinisialisasi untuk dipanggil saat formulir dikirim. Dalam fungsi ini, objek FormData digunakan untuk mengumpulkan data formulir, kemudian dilakukan permintaan POST ke server menggunakan axios. Jika permintaan berhasil, pesan sukses diset, sedangkan jika terjadi kesalahan, pesan error diset dan status error diubah sesuai kondisi respons dari server.

Secara keseluruhan, blok program ini menggambarkan logika penggunaan state, event handling, dan pengiriman permintaan ke server dalam sebuah formulir React.

- **TambahDataPage.jsx**

```
fe3 > src > pages > ✎ TambahDataPage.jsx > ...
1  import React from "react";
2  import FormTambahData from "../components/FormTambahData";
3  import Layout from "../Layout";
4
5  const TambahDataPage = ({message, Error}) => {
6    return (
7      <Layout>
8        <div className="flex w-full justify-center items-center">
9          <h1 className="text-3xl font-semibold mt-5">Tambah Data Mahasiswa</h1>
10         </div>
11         <div className="flex mt-10 justify-center">
12           <FormTambahData />
13         </div>
14       </Layout>
15     );
16   };
17 };
18
19 export default TambahDataPage;
```

Selanjutnya blok program diatas ini menggunakan modul React untuk membuat dan menggunakan komponen React. Komponen FormTambahData diimpor dari direktori "../components" untuk digunakan dalam halaman ini. Demikian pula, komponen Layout diimpor dari direktori "../Layout" untuk mengatur tata letak halaman. Fungsi komponen TambahDataPage didefinisikan untuk menerima properti message dan Error, yang kemungkinan akan digunakan untuk menampilkan pesan terkait hasil operasi formulir kepada pengguna.

Fungsi komponen TambahDataPage mengembalikan elemen-elemen React yang akan dirender. Komponen Layout digunakan untuk menyusun tata letak halaman, di mana elemen-elemen dalam halaman ini akan diposisikan sesuai dengan aturan yang ditentukan dalam komponen Layout. Elemen div dengan kelas CSS "flex w-full justify-center items-center" digunakan untuk menciptakan tata letak responsif, di mana judul "Tambah Data Mahasiswa" ditampilkan di tengah halaman dengan menggunakan properti justify-center dan items-center. Judul tersebut memiliki ukuran teks 3XL dan berat huruf sebesar "font-semibold".

Selanjutnya, terdapat elemen div dengan properti "flex mt-10 justify-center" untuk menempatkan komponen FormTambahData di tengah halaman. Komponen FormTambahData yang telah diimpor digunakan untuk kemungkinan berisi formulir untuk menambahkan data

mahasiswa. Akhirnya, tag untuk komponen Layout ditutup untuk mengakhiri tata letak halaman. Komponen `TambahDataPage` diekspor agar dapat digunakan di tempat lain dalam aplikasi. Keseluruhannya, blok program ini membentuk halaman "Tambah Data Mahasiswa" yang terstruktur dan estetis dalam aplikasi React.

- **Notification.jsx**

```
fe3 > src > components > Notification.jsx > ...
1  import React, { useState, useEffect } from "react";
2
3  function Notification({ message, isError }) {
4    const [isVisible, setIsVisible] = useState(false);
5
6    useEffect(() => {
7      if (message) {
8        setIsVisible(true);
9        setTimeout(() => {
10          setIsVisible(false);
11          // Memperbarui halaman setelah pesan muncul selama 2 detik
12          window.location.reload();
13        }, 2000);
14      }
15    }, [message]);
16
17    const notificationClass = isError ? "bg-red-500" : "bg-green-500";
18    return (
19      <div>
20        {isVisible && (
21          <div
22            className={`text-white z-50 fixed top-0 right-0 mt-20 mr-24 w-48 p-3 rounded ${notificationClass}`}
23          >
24            {message}
25          </div>
26        )}
27      </div>
28    );
29  }
30
31  export default Notification;
```

Kemudian pada blok program diatas ini mengimpor modul React, `useState`, dan `useEffect` dari pustaka React. `useState` digunakan untuk membuat state lokal, sementara `useEffect` digunakan untuk melakukan efek samping (side effect) saat komponen ini dirender.

Fungsi komponen `Notification` didefinisikan untuk menerima dua properti, yaitu `message` untuk menampilkan pesan pemberitahuan, dan `isError` yang menentukan jenis pemberitahuan (gagal atau berhasil). State lokal `isVisible` dibuat untuk mengindikasikan apakah pemberitahuan ditampilkan atau tidak. Menggunakan `useEffect`, program memantau perubahan pada properti `message`. Jika terdapat pesan, `isVisible` diatur menjadi `true`, kemudian diatur kembali menjadi `false` setelah 2 detik. Selanjutnya, halaman diperbarui dengan melakukan `reload` setelah pesan muncul selama 2 detik.

Kelas CSS untuk pemberitahuan ditentukan dengan menggunakan kondisi pada notificationClass. Jika isError bernilai true, maka kelas yang dipilih adalah untuk pemberitahuan kesalahan (bg-red-500), dan jika false, kelas untuk pemberitahuan berhasil (bg-green-500) yang dipilih.

Fungsi komponen mengembalikan elemen React yang akan dirender. Menggunakan kondisi, program menentukan apakah elemen pemberitahuan harus ditampilkan. Jika isVisible bernilai true, maka elemen pemberitahuan akan dirender. Elemen div untuk pemberitahuan dibuat dengan kelas CSS yang telah ditentukan sebelumnya, muncul di sudut kanan atas layar, dan menampilkan pesan pemberitahuan.

Komponen Notification diekspor agar dapat digunakan di tempat lain dalam aplikasi. Keseluruhannya, program ini membentuk komponen pemberitahuan dinamis yang memberikan umpan balik visual kepada pengguna terkait status operasi tertentu dalam aplikasi.

- **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    const deleteMhs = async (mhsId) => {
27      try {
28        await axios.delete(`http://localhost/pweb/be/delete.php/${mhsId}`);
29        setMsg("Data Delete Success");
30        setIsError(false);
31      } catch (error) {
32        setMsg("Data Gagal Edit");
33        setIsError(false);
34      }
35      // 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



- **Tambah Data Page**

The screenshot displays a web application interface for adding student data. The header section includes the text 'JUAN 23' on the left, navigation links 'Home' and 'Data Mahasiswa' in the center, and a 'JUAN 23' button on the right. The main content area is titled 'Tambah Data Mahasiswa' and features a form with three input fields: 'NPM' (with placeholder 'NPM'), 'Nama' (with placeholder 'Nama'), and 'Kelas' (with placeholder 'Kelas'). Below the fields is a blue button labeled 'Tambah Data'.