**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



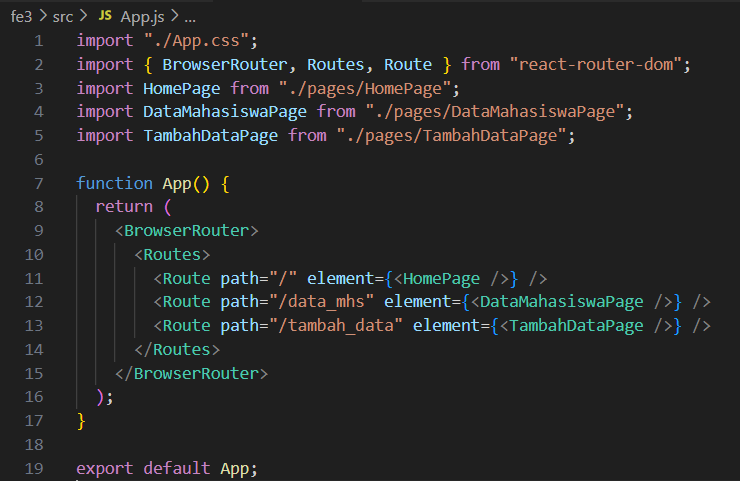
**LABORATORIUM TEKNIK INFORMATIKA**

**UNIVERSITAS GUNADARMA**

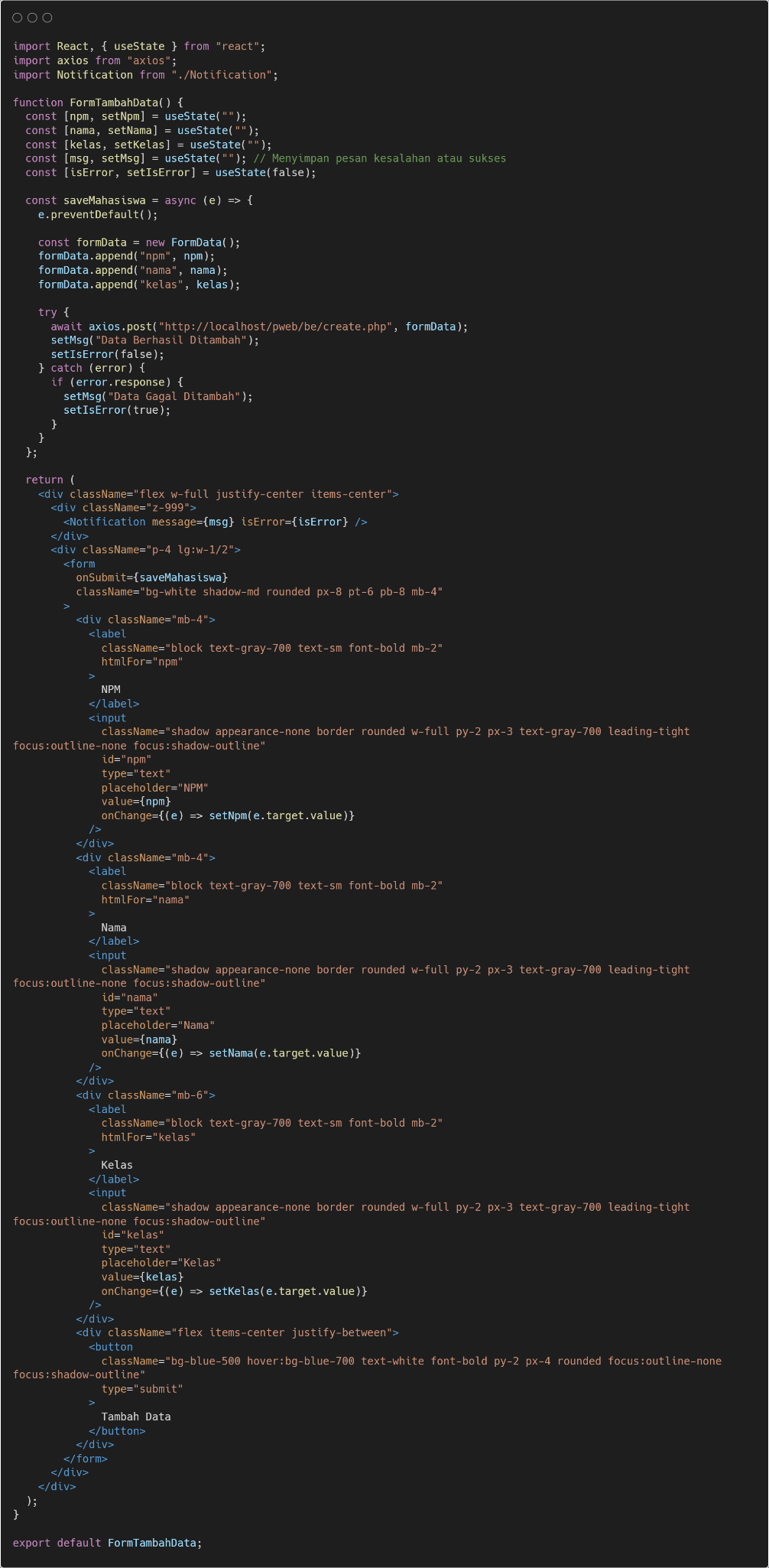
**2023**

**LISTING PROGRAM**

* **app.js**

****

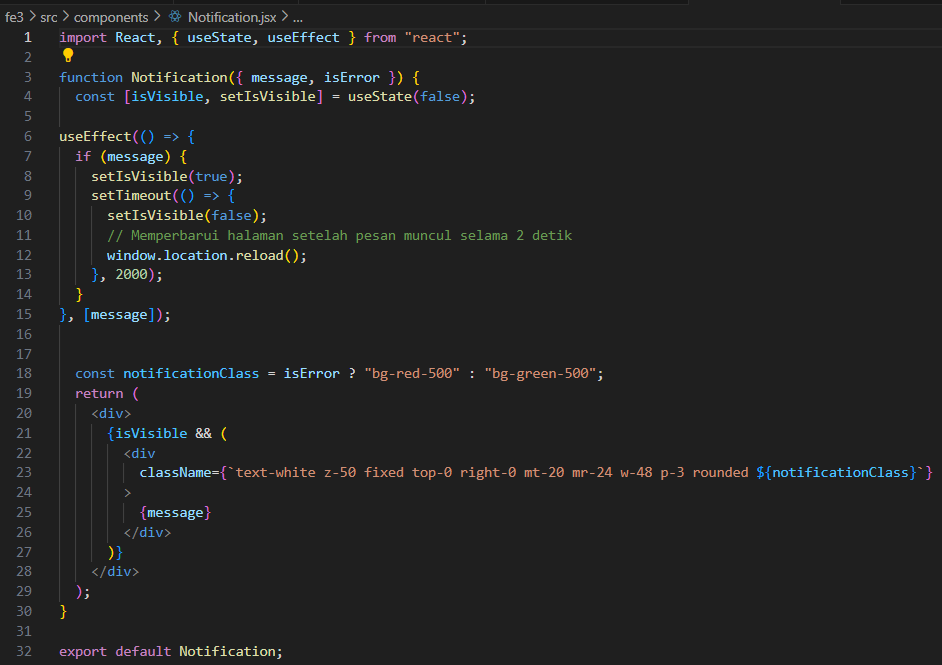
* **FormTambahData.jsx**



* **TambahDataPage.jsx**



* **Notification.jsx**



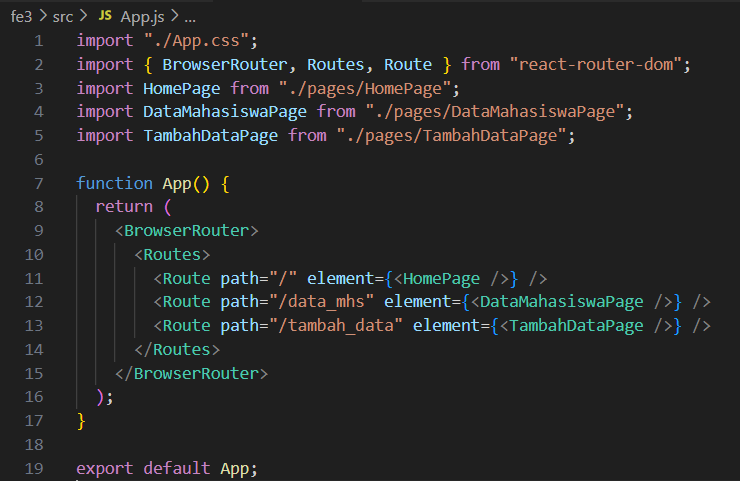
* **DataMahasiswapage.jsx**



**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

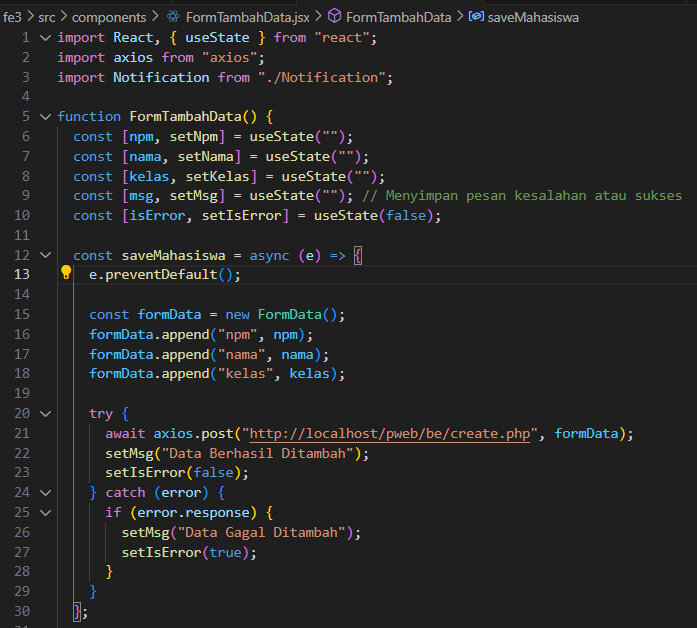
* **app.js**

****

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**

****

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**

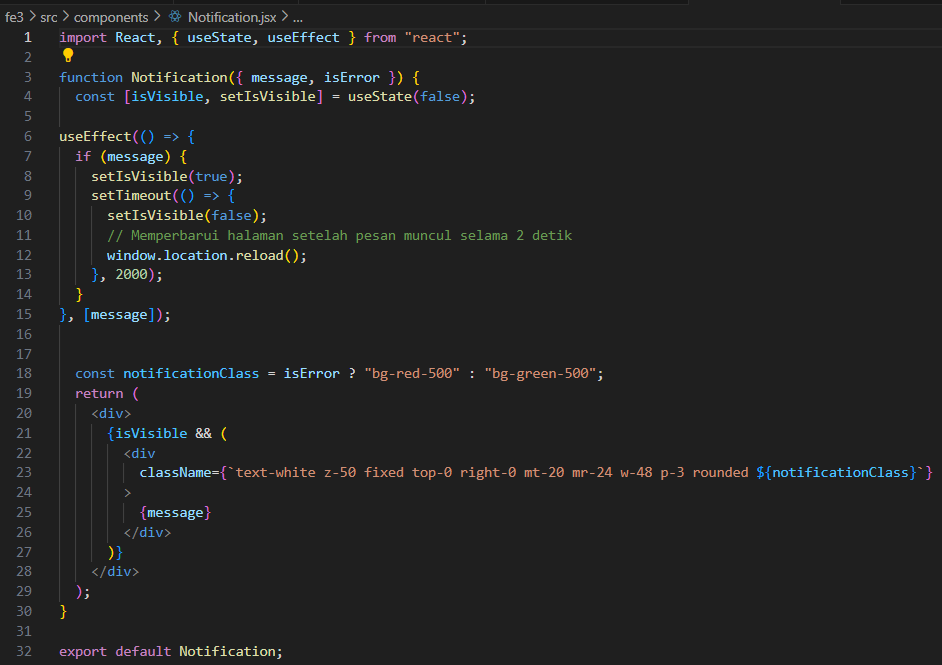


Selanjutnya blok program diatas ini menggunakan modul React untuk membuat dan menggunakan komponen React. Komponen FormTambahData diimpor dari direktori "../components" untuk digunakan dalam halaman ini. Demikian pula, komponen Layout diimpor dari direktori "./Layout" untuk mengatur tata letak halaman. 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**



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**



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.

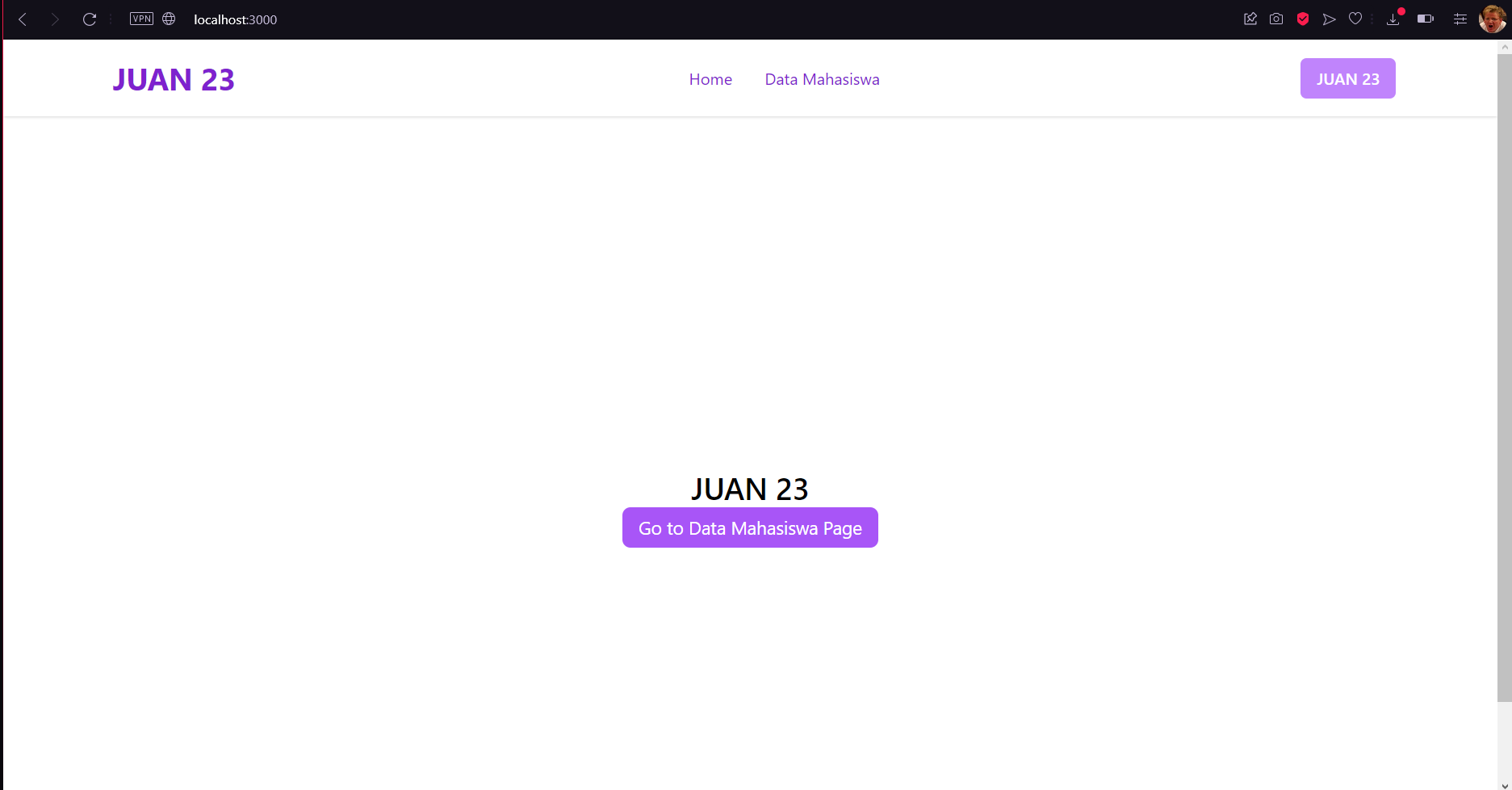
Dalam useEffect, fungsi getData dipanggil untuk mengambil data mahasiswa dari server menggunakan Axios. Hasil respons dari server digunakan untuk mengupdate state mahasiswaData, yang berisi array data mahasiswa. Jika data berhasil diambil, array tersebut akan digunakan untuk merender tabel yang menampilkan 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.

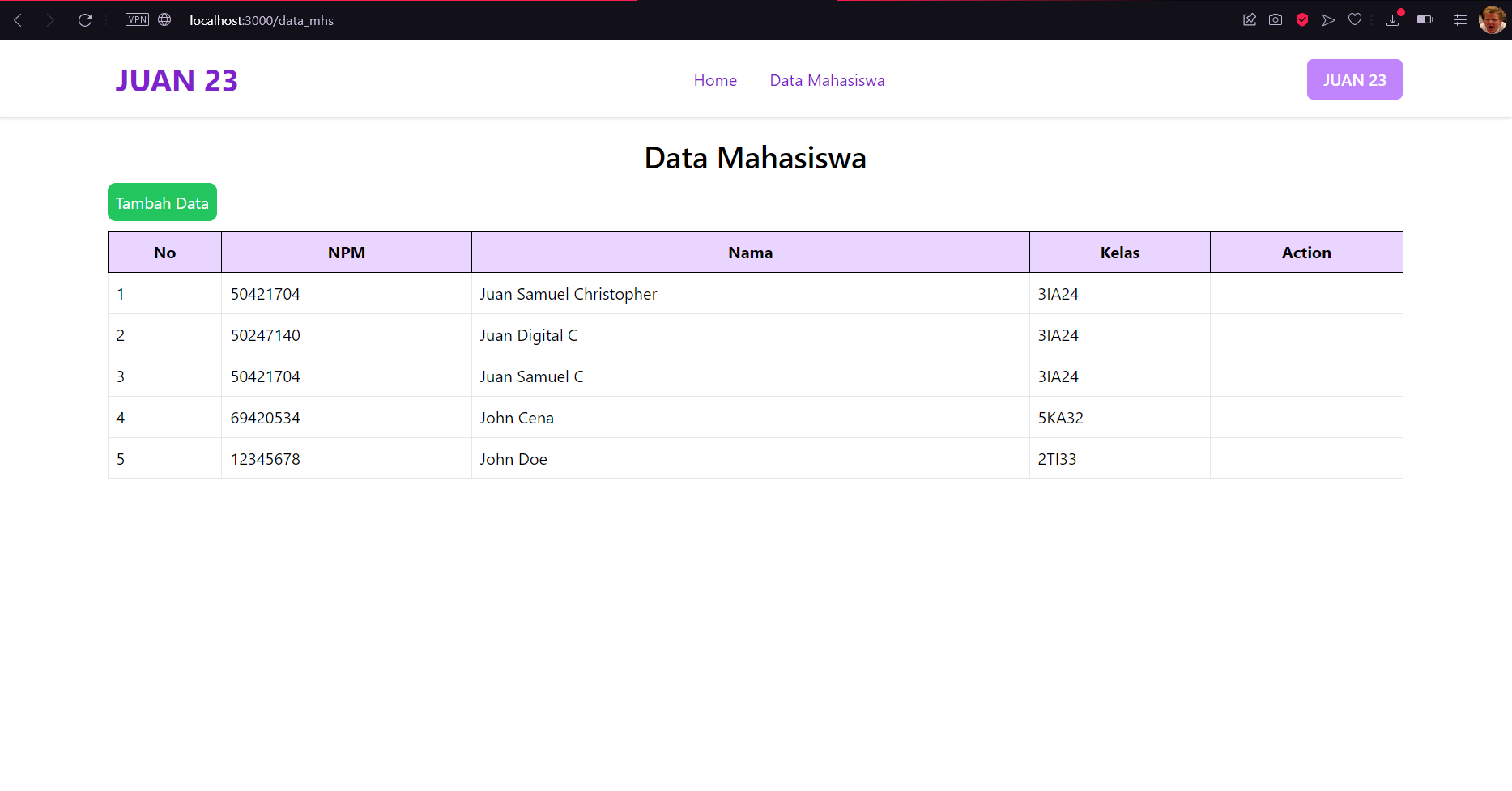
Tata letak halaman ini diatur menggunakan komponen Layout, dan terdapat tombol "Tambah Data" yang mengarahkan pengguna ke halaman penambahan data baru. Tabel ditampilkan dengan menggunakan elemen <table>, <thead>, dan <tbody>, serta menggunakan kondisi untuk menangani apakah terdapat data mahasiswa yang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

**OUTPUT PROGRAM**

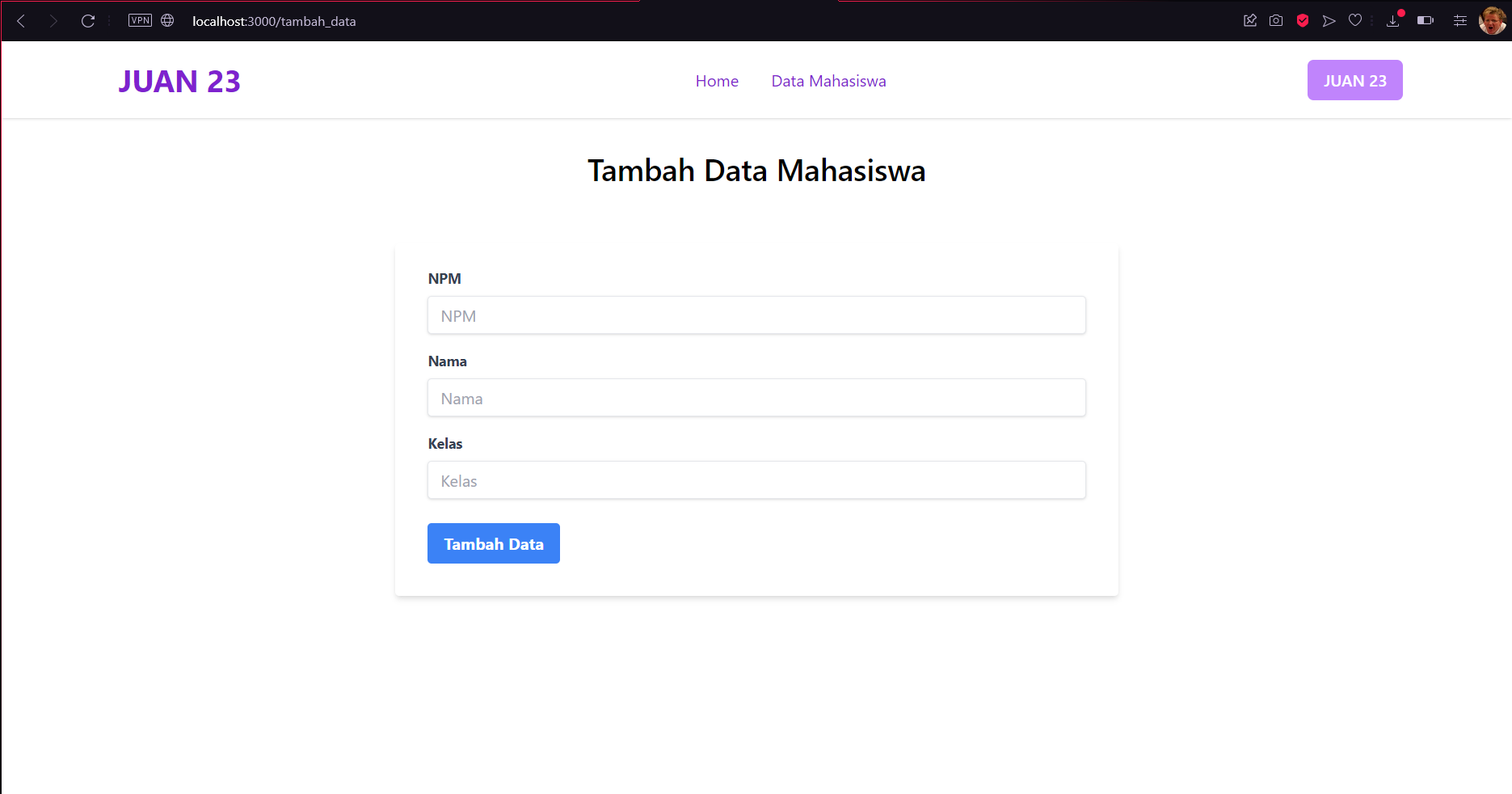
* **Homepage**

****

* **Data Mahasiswa Page**

****

* **Tambah Data Page**

****