**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



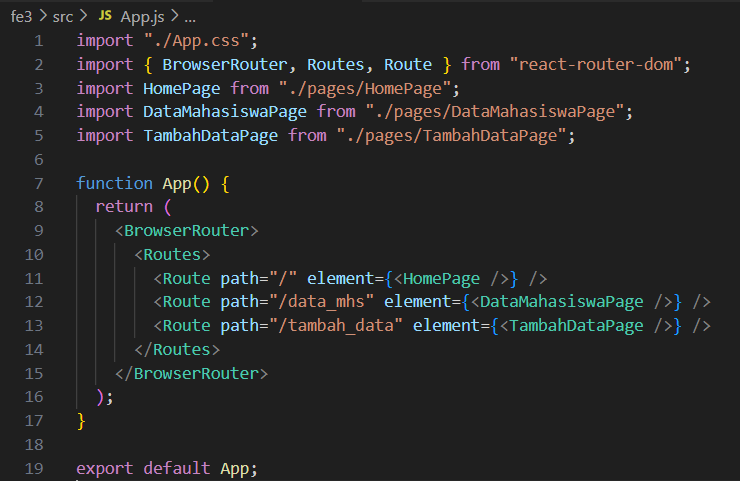
**LABORATORIUM TEKNIK INFORMATIKA**

**UNIVERSITAS GUNADARMA**

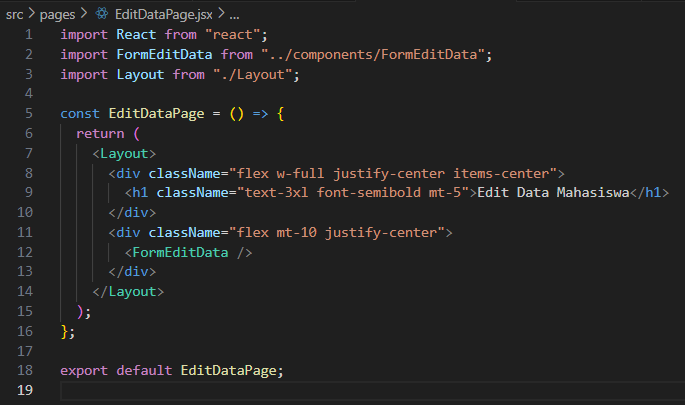
**2024**

**LISTING PROGRAM**

* **app.js**

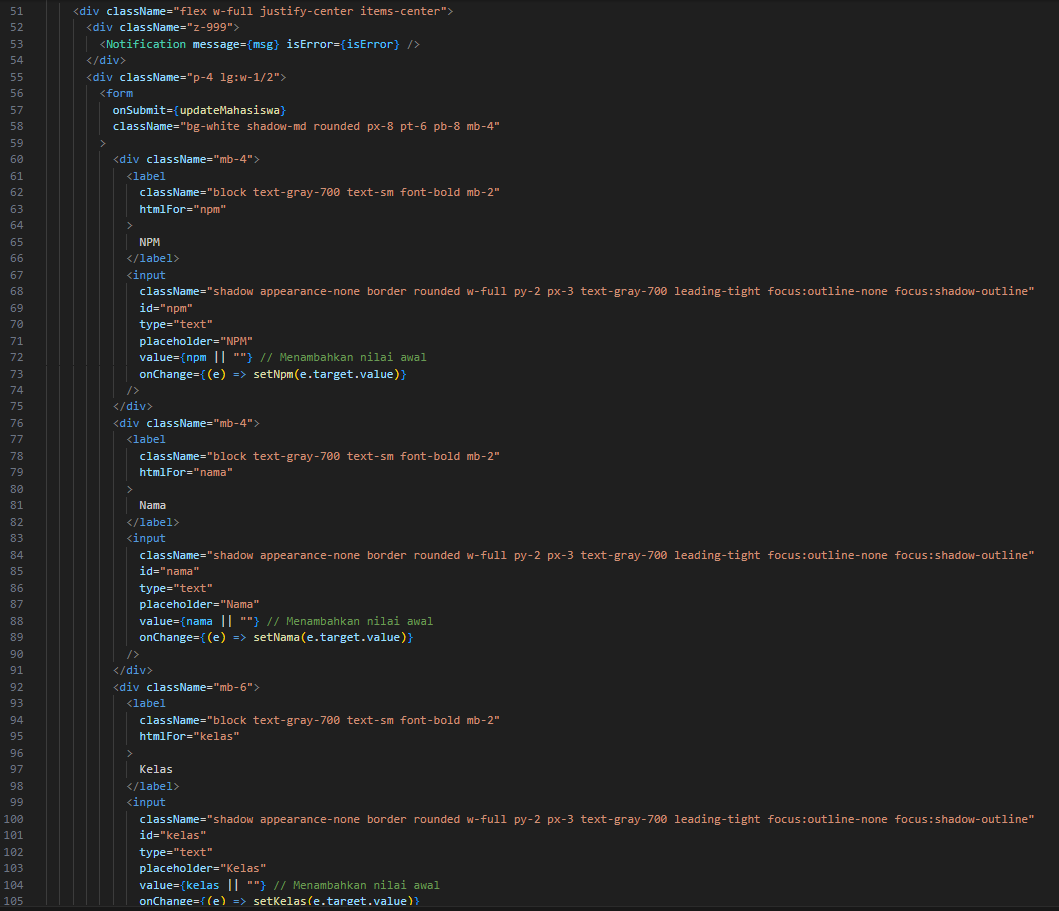
****

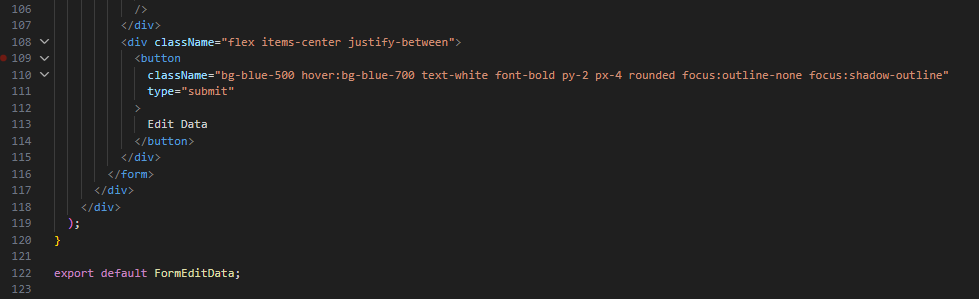
* **Editdata.jsx**

****

* **Formeditdata.jsx**

****

****

****

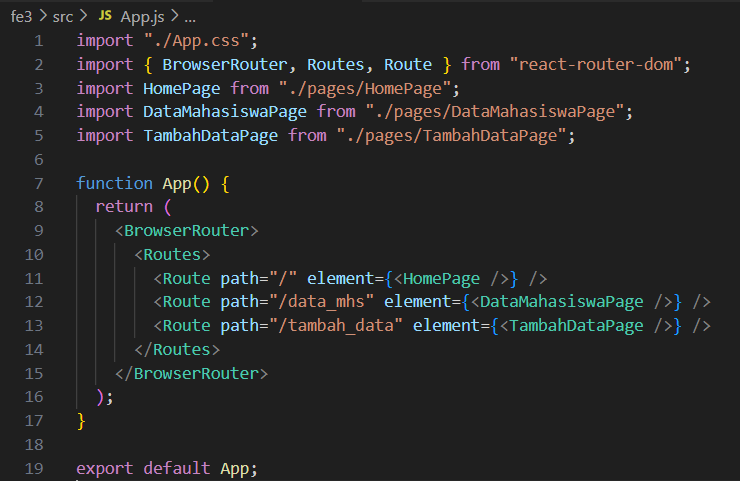
* **DataMahasiswapage.jsx**



**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 ACT4

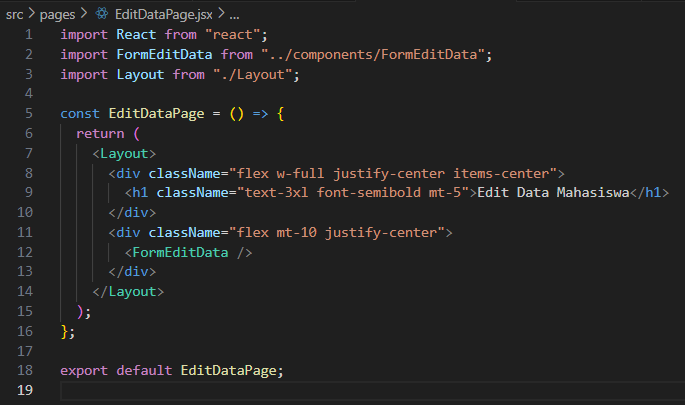
* **app.js**

****

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.

* **Editdata.jsx**

****  
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**

****

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 estetis.

* **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.

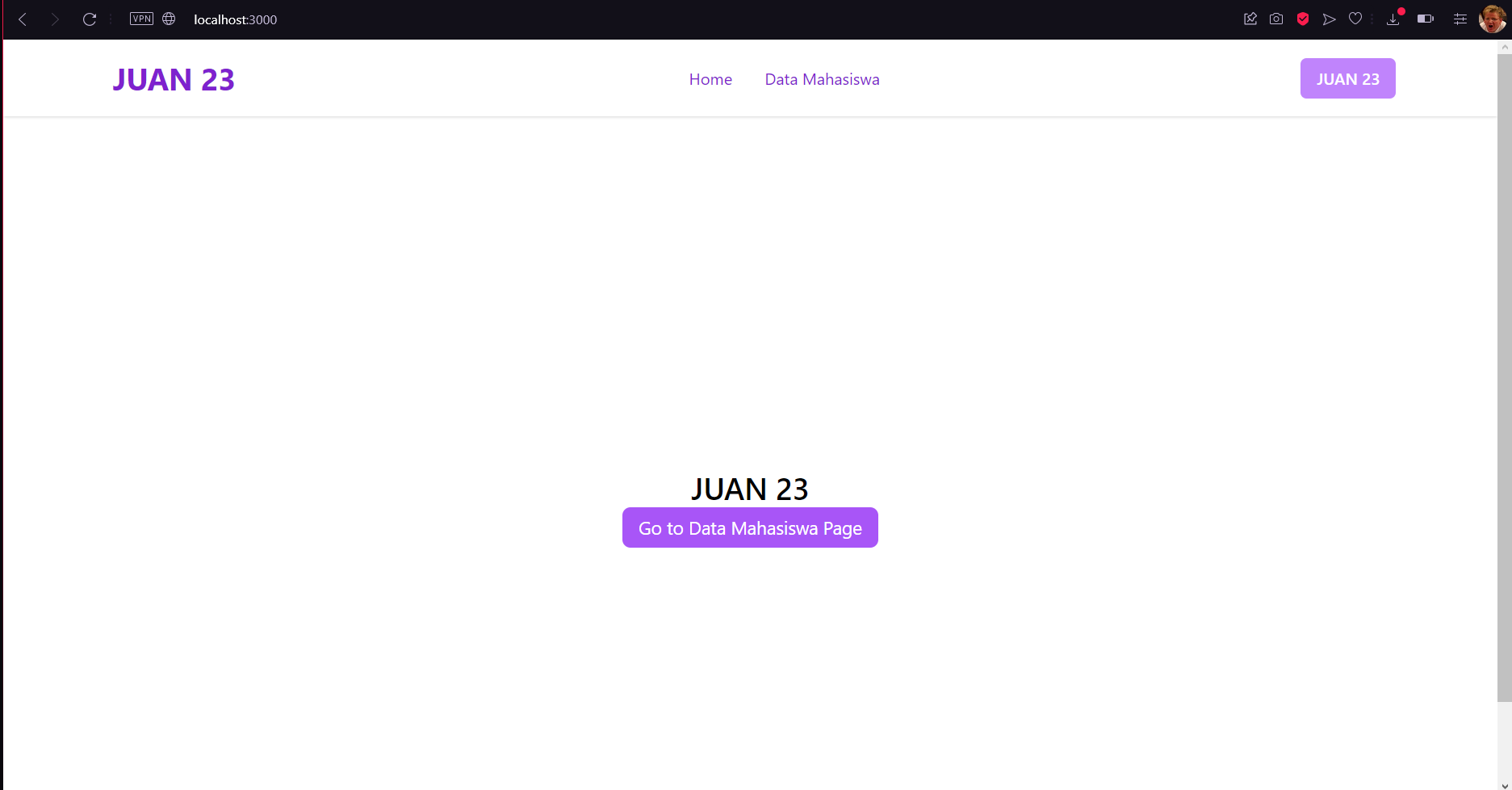
Dalam useEffect, fungsi getData dipanggil untuk mengambil data mahasiswa dari server menggunakan Axios. Hasil respons dari server digunakan untuk mengupdate state mahasiswaData, yang berisi array data mahasiswa. Jika data berhasil diambil, array tersebut akan digunakan untuk merender tabel yang menampilkan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

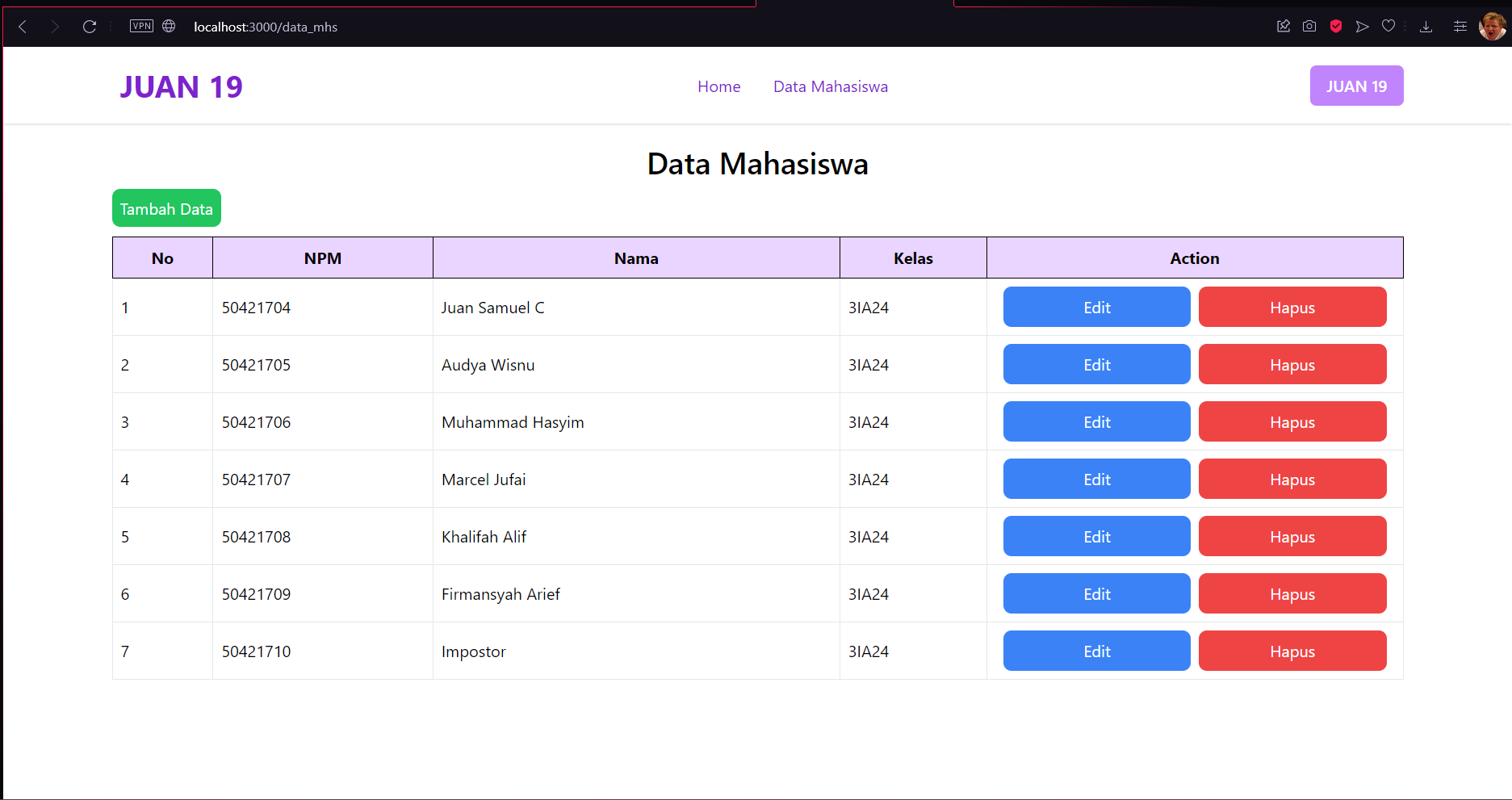
Tata letak halaman ini diatur menggunakan komponen Layout, dan terdapat tombol "Tambah Data" yang mengarahkan pengguna ke halaman penambahan data baru. Tabel ditampilkan dengan menggunakan elemen <table>, <thead>, dan <tbody>, serta menggunakan kondisi untuk menangani apakah terdapat data mahasiswa yang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

**OUTPUT PROGRAM**

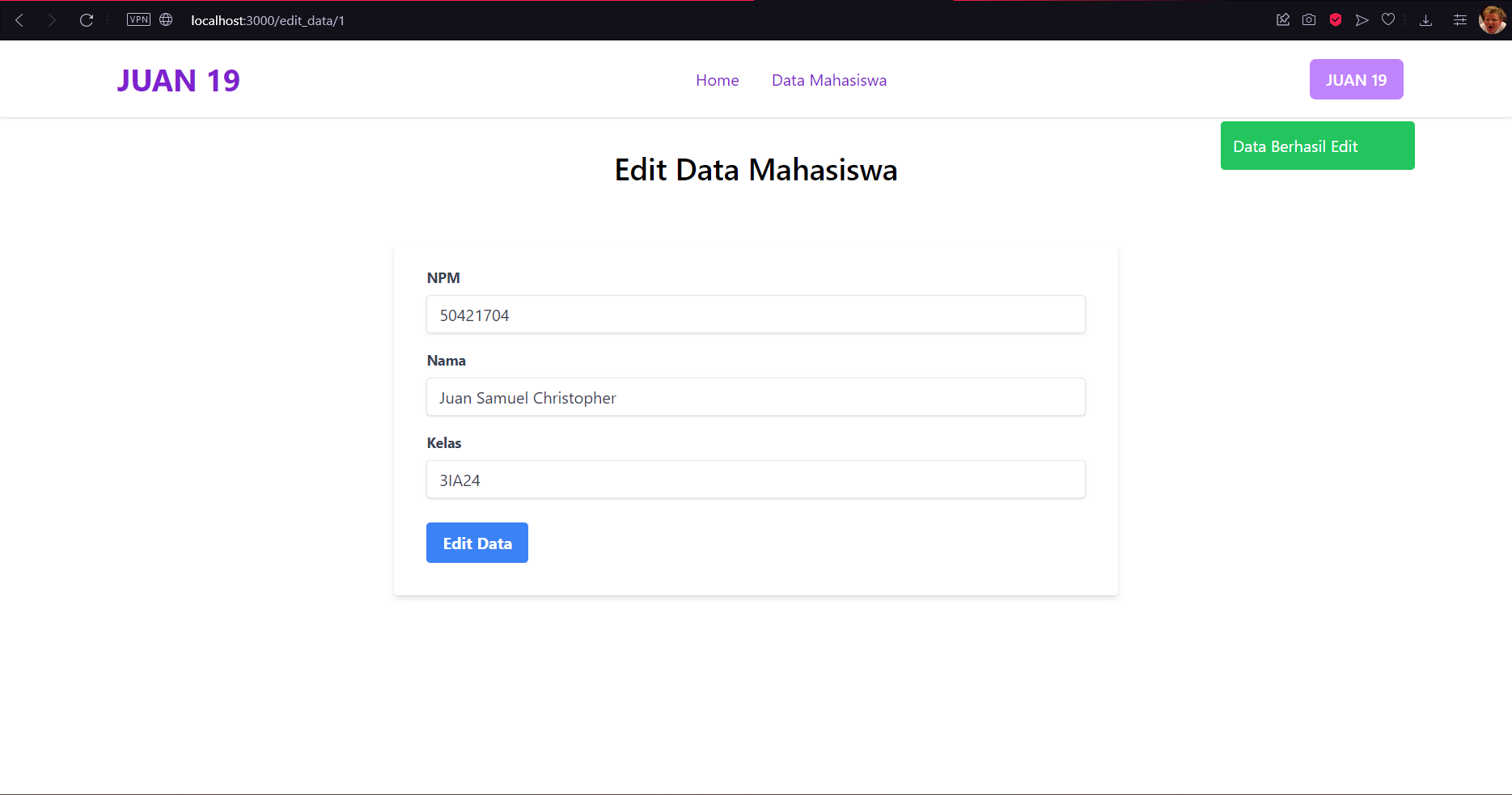
* **Homepage**

****

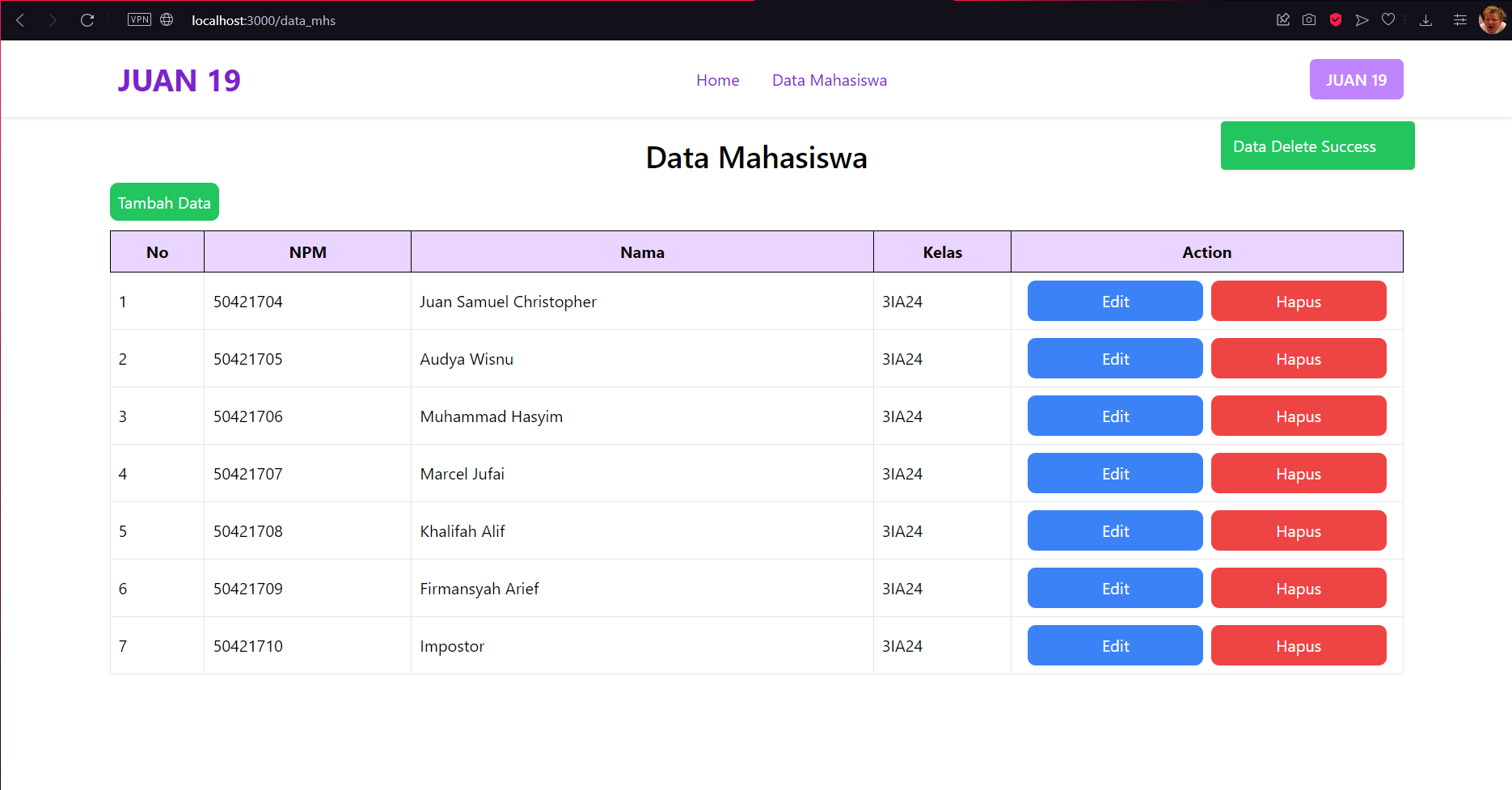
* **Data Mahasiswa Page**

****

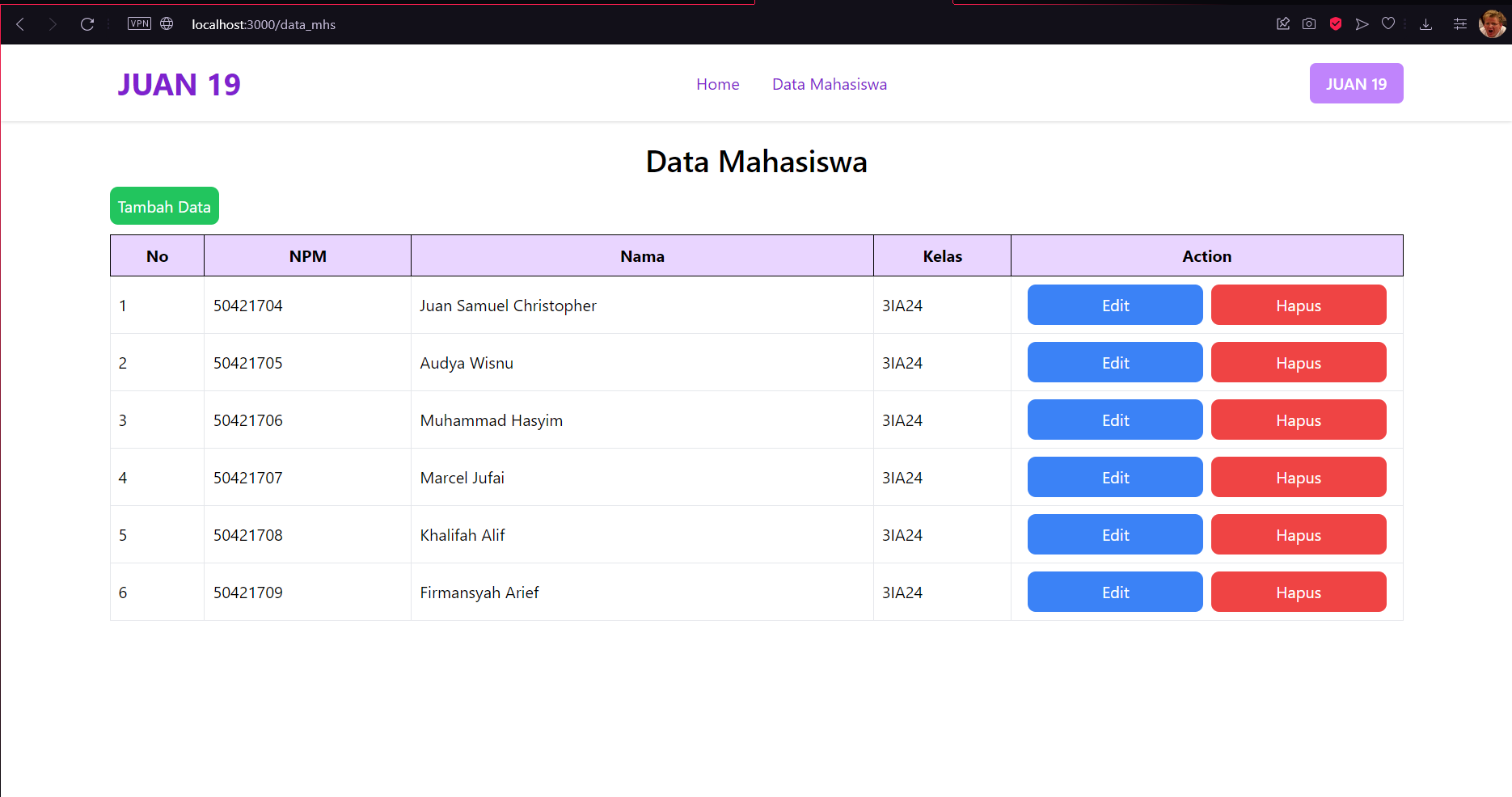
* **Edit Data**

****

* **Hapus Data**

****

* **Data akhir mahasiswa**

****