**LAPORAN TUGAS AKHIR KEAMANAN APLIKASI BESERTA PEJELASANNYA**



**DIBUAT OLEH:**

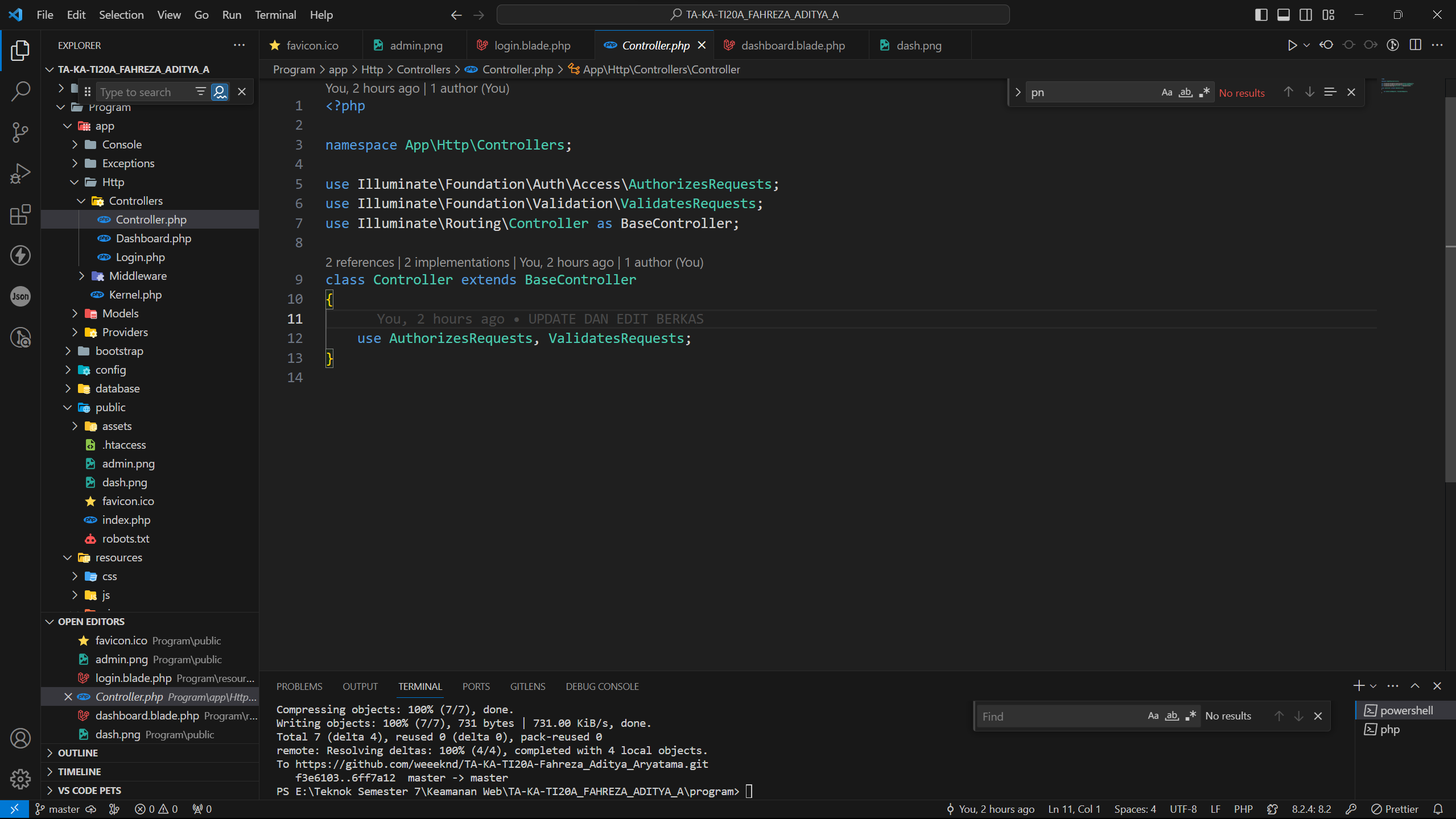
**FAHREZA ADITYA ARYATAMA – 20313009**

**UNIVERSITAS TEKNOKRAT INDONESIA**

**BANDAR LAMPUNG**

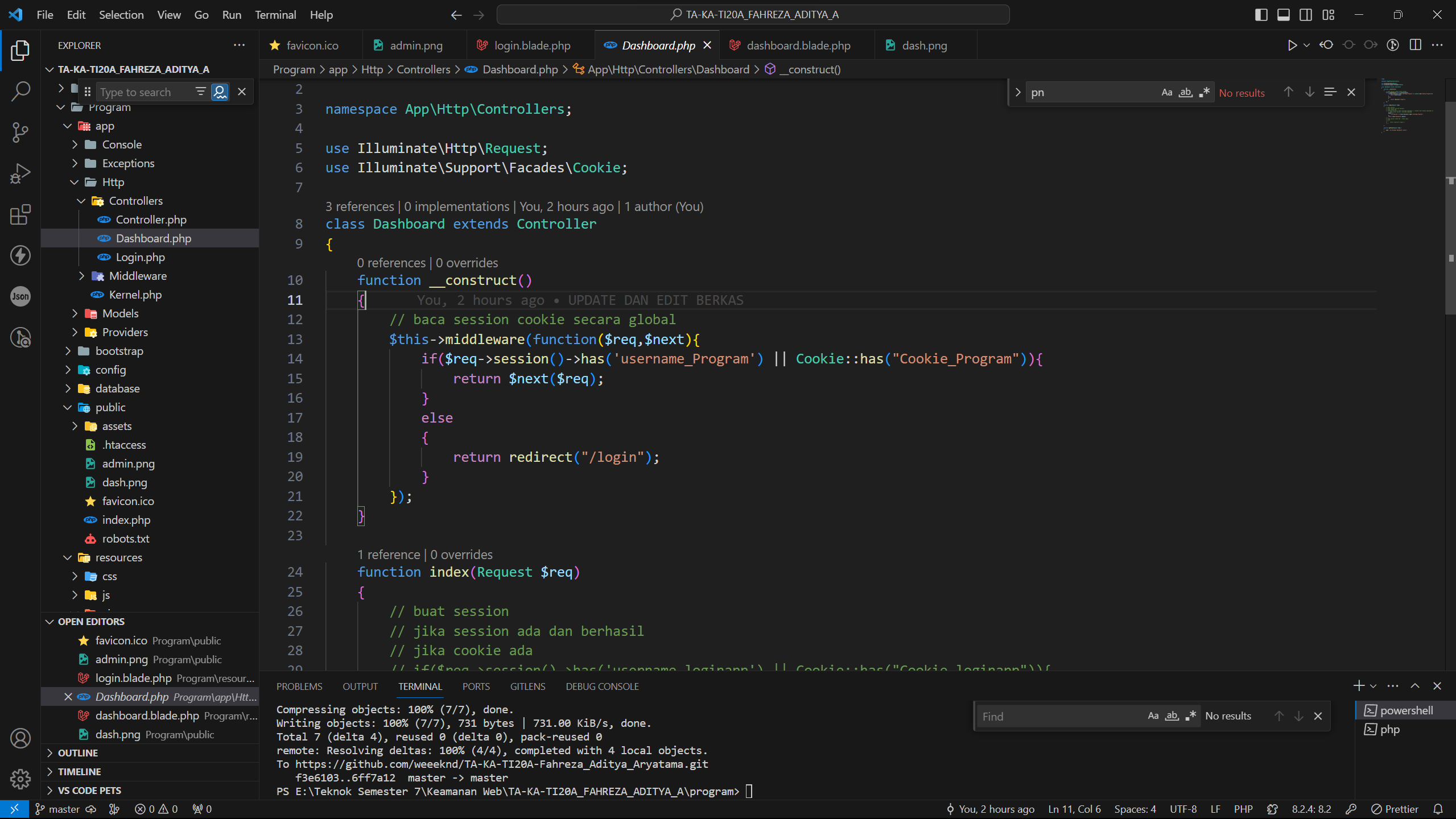
**2024**

1. **Mengedit dan menambahkan fungsi di controllers**



Didalam sebuah Folder **controllers** terdapat beberap fungsi yang di tambah kan diantara lain ada beberpa fungsi yaitu:

* Dashboard



Program pada bagian dashboard ini memiliki beberapa fungsi utama:

1. Middleware untuk Autentikasi:

- Konstruktor controller ini memiliki sebuah middleware yang bertugas untuk memeriksa apakah pengguna telah login atau memiliki cookie yang valid. Jika pengguna memiliki session 'username\_Program' atau cookie 'Cookie\_Program', maka akses diteruskan ke endpoint yang sesuai. Jika tidak, pengguna akan diarahkan kembali ke halaman login.

2. Method `index`:

- Metode `index` digunakan untuk menangani permintaan ke halaman dashboard. Jika pengguna sudah login (sesuai dengan session atau cookie), halaman dashboard akan ditampilkan dengan informasi sesuai dengan data sesi, seperti contoh menyertakan "username\_Program" dalam array data.

3. Method `getView`:

- Metode `getView` digunakan untuk menangani permintaan ke halaman dashboard (view). Pada contoh ini, metode hanya mencetak pesan sederhana, "Ini Halaman Dashboard (view)".

4. Session Handling:

- Terdapat penggunaan session Laravel (`$req->session()->get()`) untuk mendapatkan nilai dari session 'username\_Program'. Session ini kemudian dimasukkan ke dalam array data untuk ditampilkan di halaman dashboard.

5. Redirection to Login Page:

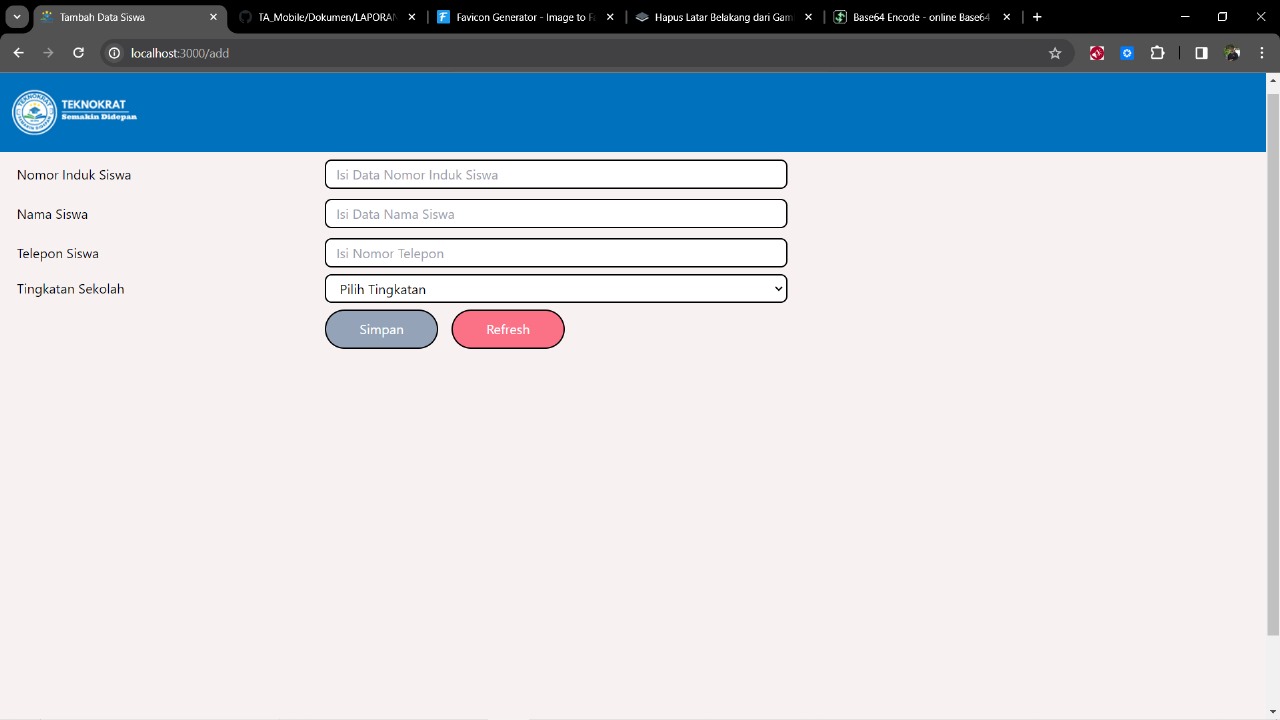
- Jika pengguna tidak memiliki session atau cookie yang valid, mereka akan diarahkan kembali ke halaman login dengan menggunakan fungsi `redirect("/login")`.

6. Cookie Handling:

- Ada penanganan cookie dengan menggunakan `Cookie::has()` untuk memeriksa apakah cookie 'Cookie\_Program' tersedia.

Controller ini bertindak sebagai filter untuk memastikan bahwa hanya pengguna yang sudah login atau memiliki cookie yang valid yang dapat mengakses halaman dashboard. Jika tidak, mereka akan diarahkan kembali ke halaman login.

* Login

****

Program pada bagian login ini bertujuan untuk mengelola proses otentikasi pengguna, termasuk pemeriksaan kredensial, pembuatan session, pengaturan cookie, dan fungsi logout. Berikut adalah penjelasan fungsi utama dari program pada bagian login:

1. Inisialisasi Model:

- Konstruktor (`\_\_construct`) digunakan untuk membuat objek model `Mlogin` yang digunakan untuk berinteraksi dengan data pengguna pada database.

2. Method `index`:

- Metode ini bertanggung jawab untuk menampilkan halaman login. Ketika URL "/login" diakses, controller ini akan memanggil view "login" dan menampilkannya kepada pengguna.

3. Method `getLogin`:

- Metode ini menangani permintaan login yang dikirimkan dari halaman login menggunakan teknik AJAX (dengan menggunakan fetch API). Beberapa langkah yang dilakukan oleh metode ini adalah:

- Mengumpulkan data input berupa "username", "password", dan "ingat" (checkbox "Ingat Saya").

- Melakukan enkripsi password menggunakan fungsi `base64\_encode(md5($req->password))`.

- Memeriksa apakah kombinasi "username" dan "password" ada dalam database menggunakan model `Mlogin`.

- Jika data ditemukan, membuat session "username\_Program" dengan nilai nama pengguna.

- Jika opsi "Ingat Saya" diaktifkan, membuat cookie "Cookie\_Program" dengan nilai nama pengguna yang akan berlaku selama 120 menit (2 jam).

- Mengembalikan nilai output (1 jika berhasil, 0 jika gagal) dalam bentuk JSON untuk ditangkap oleh JavaScript pada sisi klien.

4. Method `setLogout`:

- Metode ini digunakan untuk melakukan proses logout. Beberapa tindakan yang diambil oleh metode ini meliputi:

- Menghapus session "username\_Program" untuk menandakan pengguna sudah logout.

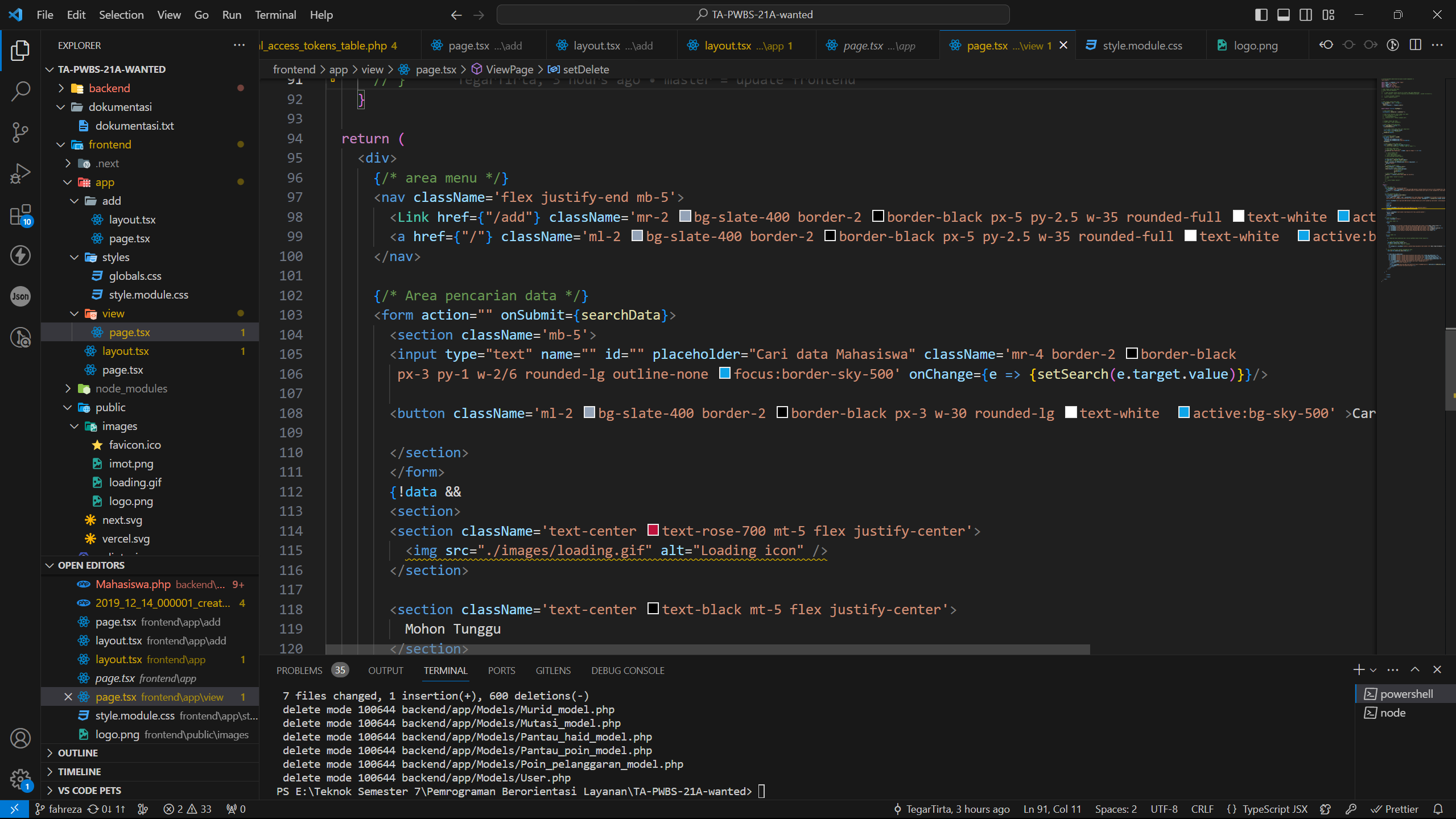
- Menghapus cookie "Cookie\_Program" jika ada.

- Mengarahkan pengguna kembali ke halaman login setelah logout.

Penting untuk diingat bahwa penggunaan enkripsi seperti `base64\_encode(md5($req->password))` mungkin tidak disarankan untuk keamanan yang optimal. Sebaiknya, pertimbangkan untuk menggunakan metode hashing yang lebih aman seperti bcrypt yang telah disediakan oleh Laravel. Selain itu, pastikan untuk menyimpan password dengan benar di database dan hindari menggunakan metode yang tidak aman seperti MD5 untuk keamanan password.

1. **Mengedit merancang sistem di folder style**

* **File page\_tsx**

****

Berkas `**page.tsx**` dalam folder "view" sepertinya merupakan berkas TypeScript yang berisi komponen React, kemungkinan mewakili suatu halaman atau tampilan dalam aplikasi web. Berikut adalah deskripsi dari fungsionalitas utama dalam kode yang diberikan:

1. programImports:program

- Berkas dimulai dengan mengimpor modul dan pustaka yang diperlukan, termasuk React, Link dari Next.js, useSWR untuk pengambilan data, Md5 untuk pembuatan hash MD5, dan Result dari 'postcss'.

2. programPengambilan Data (`fetchData` function):program

- Terdapat fungsi `fetchData` yang didefinisikan, yang merupakan fungsi bantu yang digunakan dengan `useSWR` untuk mengambil data dari endpoint API yang ditentukan. Ini menggunakan API `fetch` dan mengembalikan respons JSON yang telah diuraikan.

3. programFungsionalitas Komponen (`ViewPage` component):program

- Komponen `ViewPage` didefinisikan sebagai komponen fungsional dalam TypeScript.

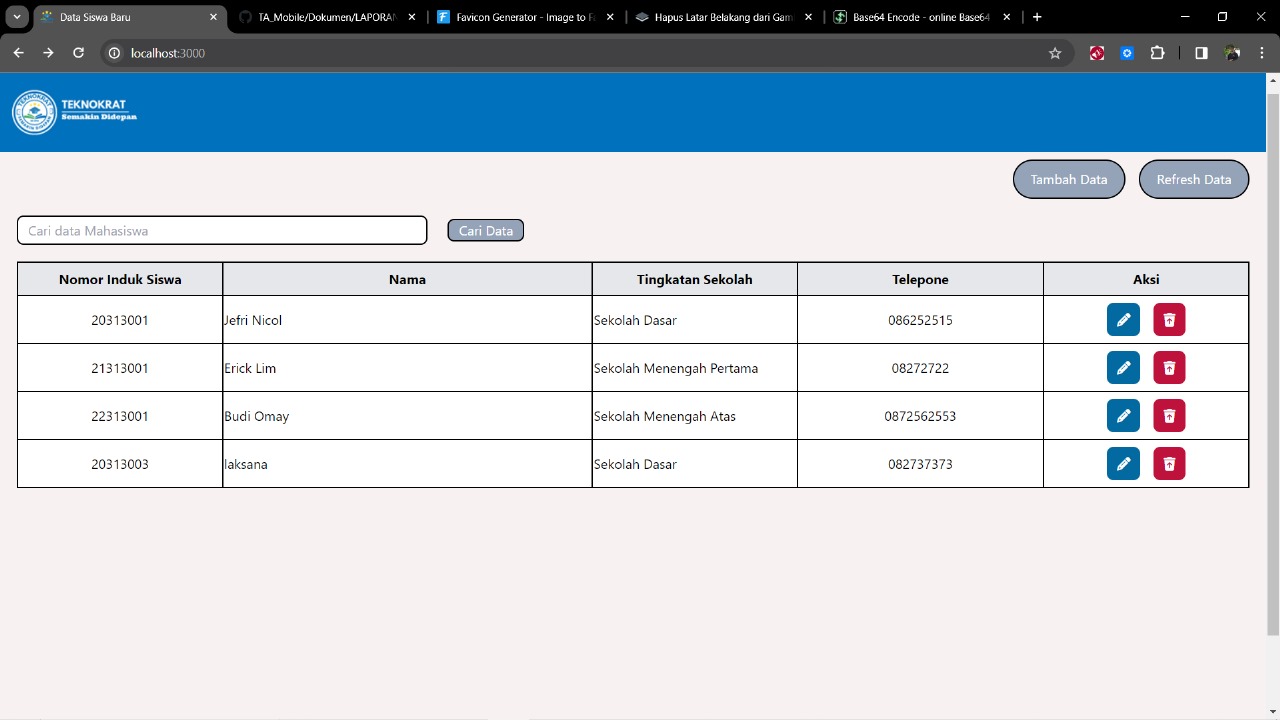
- Menggunakan hook `useState` untuk mengelola keadaan dari pencarian (`search`).

- Terdapat fungsi `getData` yang di komentar, yang sepertinya merupakan fungsi asinkron untuk mengambil data dari server. Namun, saat ini tidak digunakan.

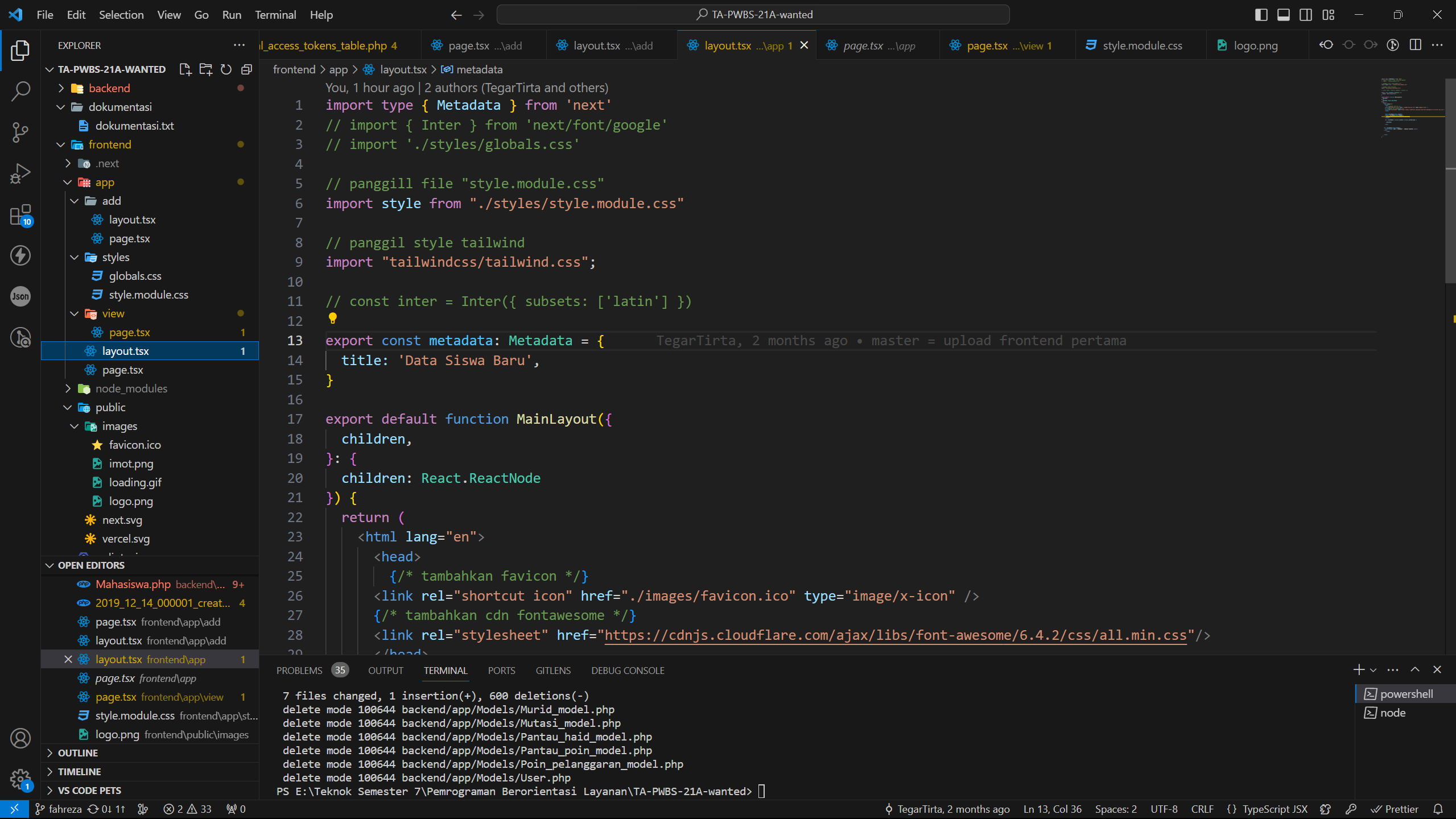
- Komponen merender bagian navigasi dengan tautan untuk menambahkan data dan menyegarkan data.

- Menyertakan formulir pencarian yang memungkinkan pengguna memasukkan kueri untuk menyaring data yang ditampilkan.

**Tampilan dari interface program**



1. **Mengedit merancang sistem di folder style**



Berkas `layout.tsx` dalam folder "view" kemungkinan besar merupakan komponen React yang bertindak sebagai tata letak atau layout utama untuk halaman-halaman dalam aplikasi. Berikut adalah deskripsi dari fungsionalitas utama dalam kode yang diberikan:

1. programImport Tipe dan Modul:program

- Program mengimpor tipe `Metadata` dari 'next', menunjukkan penggunaan metadata Next.js.

- `style.module.css` diimpor sebagai objek `style`.

- "tailwindcss" diimpor menggunakan "tailwind.css".

2. programMetadata:program

- Ada definisi metadata menggunakan konstanta `metadata`. Metadata ini mungkin digunakan untuk mengatur judul halaman atau properti metadata lainnya.

3. programKomponen `MainLayout`:program

- Komponen utama yang didefinisikan dalam berkas ini adalah `MainLayout`.

- Menerima prop `children`, yang merupakan elemen React yang akan di-render di dalam layout.

- Merupakan elemen dasar HTML dengan tag `<html>`, mencakup elemen `<head>` dan `<body>`.

4. programHead Section:program

- Menambahkan favicon dengan tag `<link>` untuk ikon situs.

- Mengimpor ikon Font Awesome menggunakan tag `<link>` untuk mengakses ikon-ikon dari Font Awesome melalui CDN.

- Saat ini, penggunaan `metadata` tidak terlihat dalam penentuan judul halaman.

5. programBody Section:program

- Memiliki tag `<body>` dengan kelas `style.layout`.

- Mencakup elemen `<header>` dengan kelas `style.header`, yang berisi logo situs.

- Memiliki div dengan kelas `style.content` dan `style.content\_bg` untuk menentukan gaya konten halaman.

- Menampilkan elemen `children` di dalam div tersebut, yang mewakili konten aktual halaman yang menggunakan layout ini.

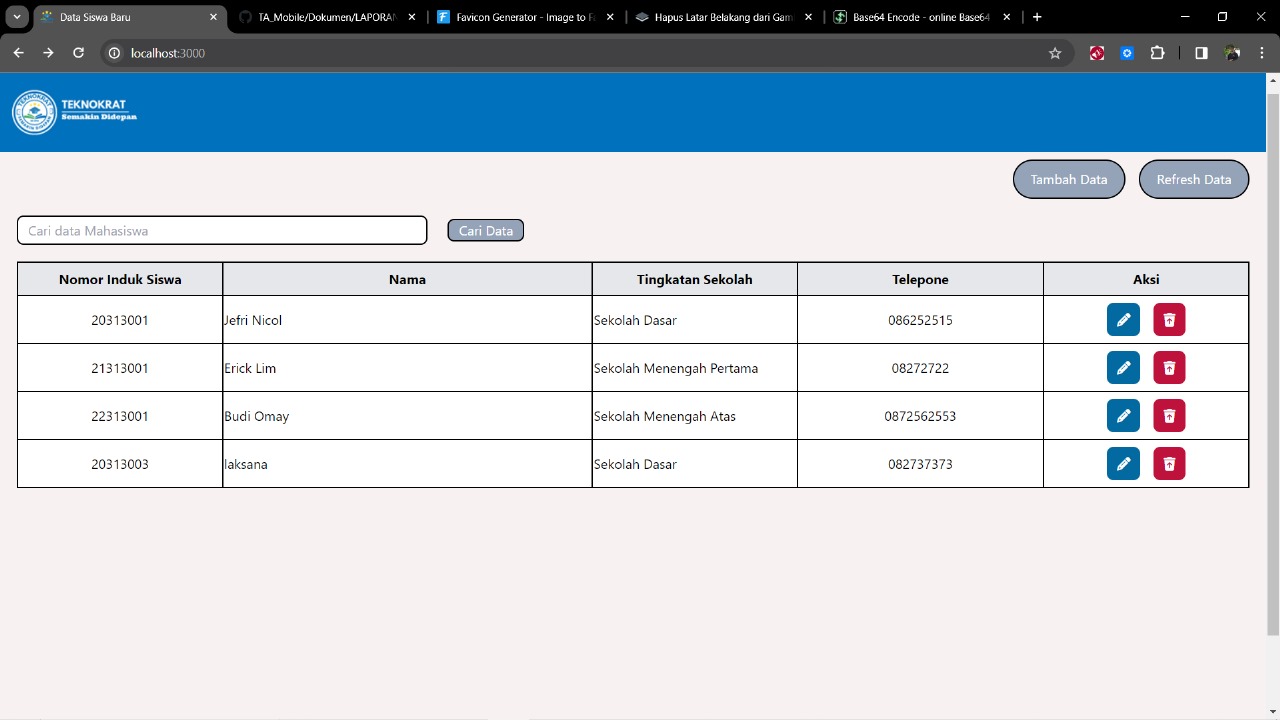
6. programFooter Section:program

- Memiliki div dengan kelas `style.footer` yang berisi elemen `<footer>`. Footer ini menampilkan informasi hak cipta dan pesan branding.

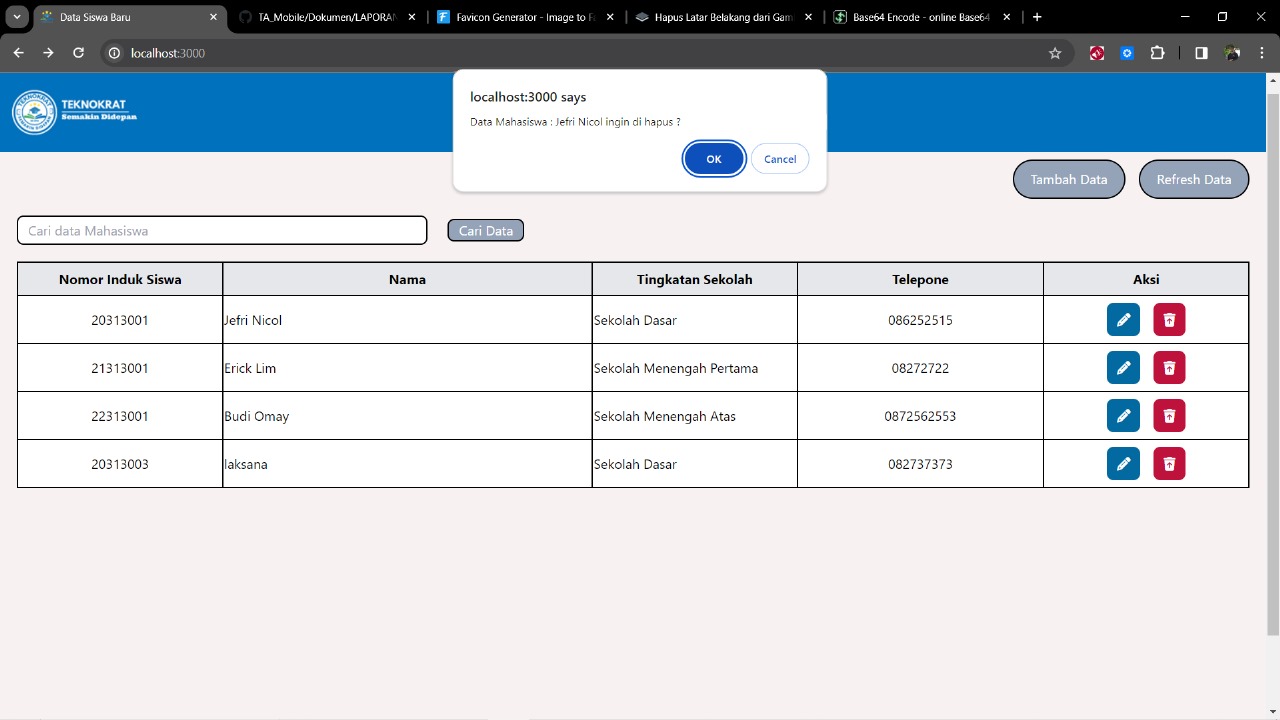
Berdasarkan struktur tersebut, `MainLayout` tampaknya dirancang untuk memberikan tata letak konsisten dan elemen umum di seluruh halaman dalam aplikasi. Elemen seperti header, konten, dan footer diatur dengan menggunakan gaya dari berkas `style.module.css`.

1. **Tutorial Penggunaan Website**

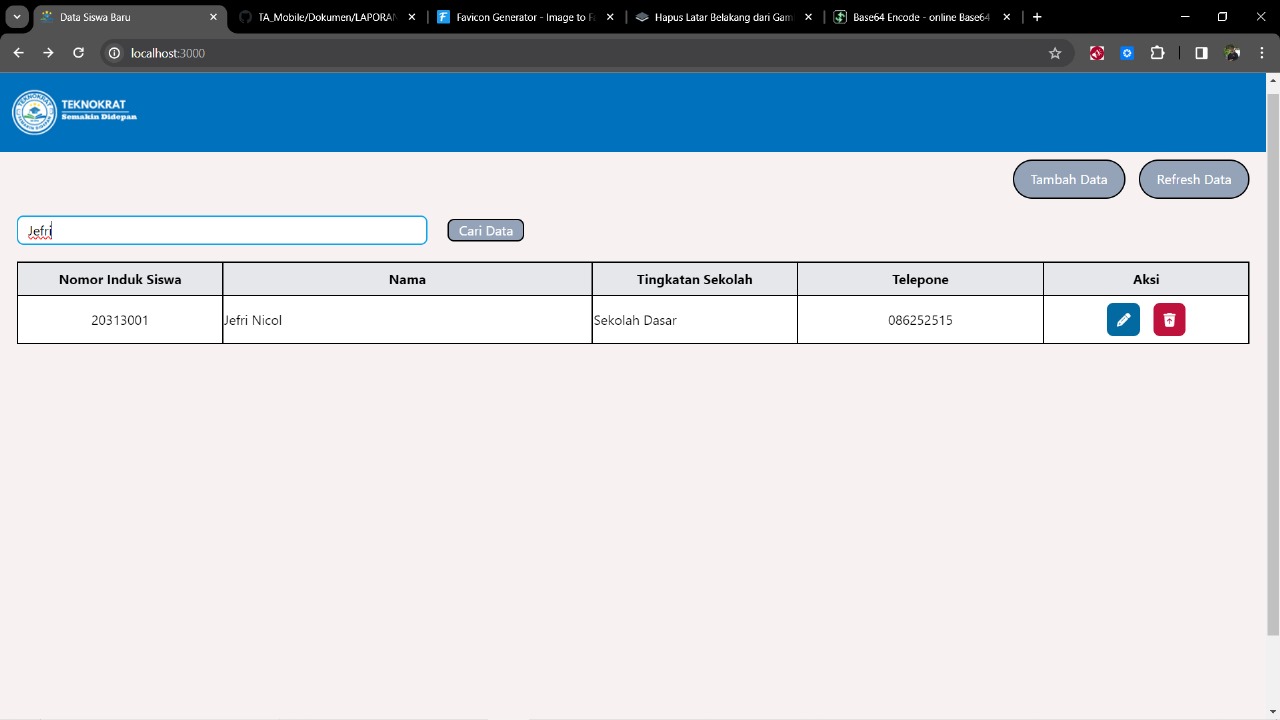
* **Tampilan Awal dari website**

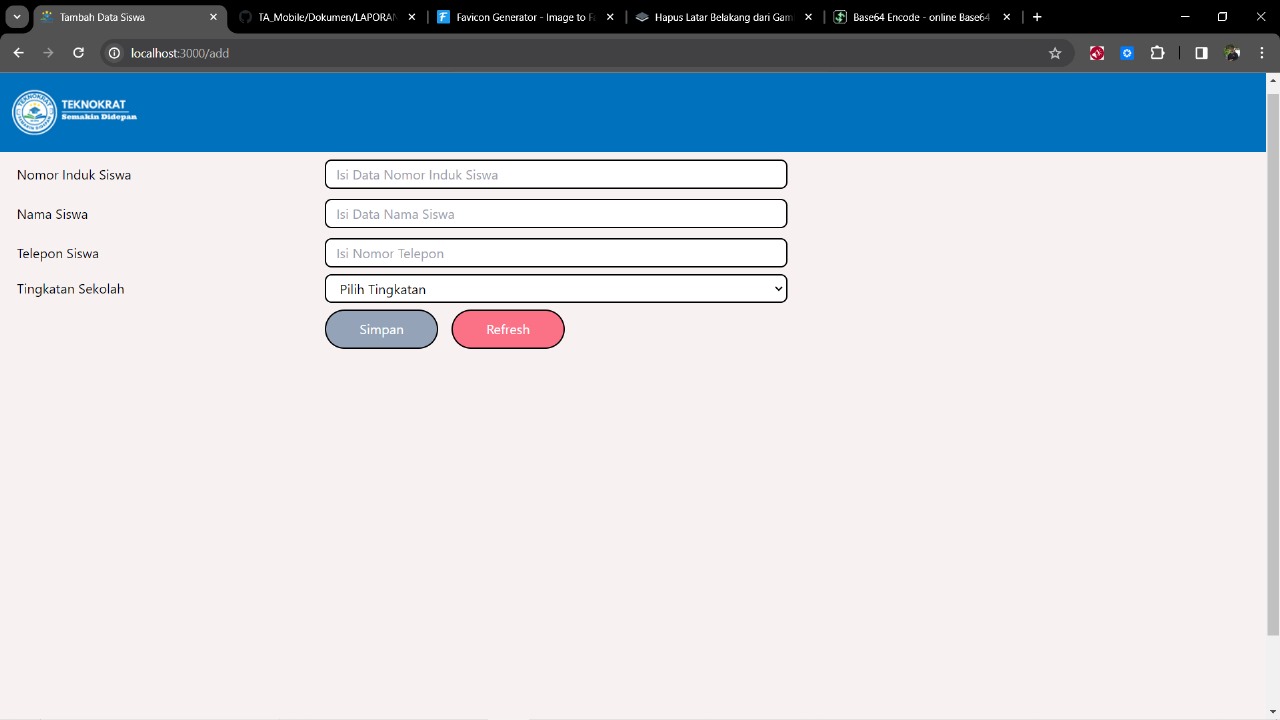
****

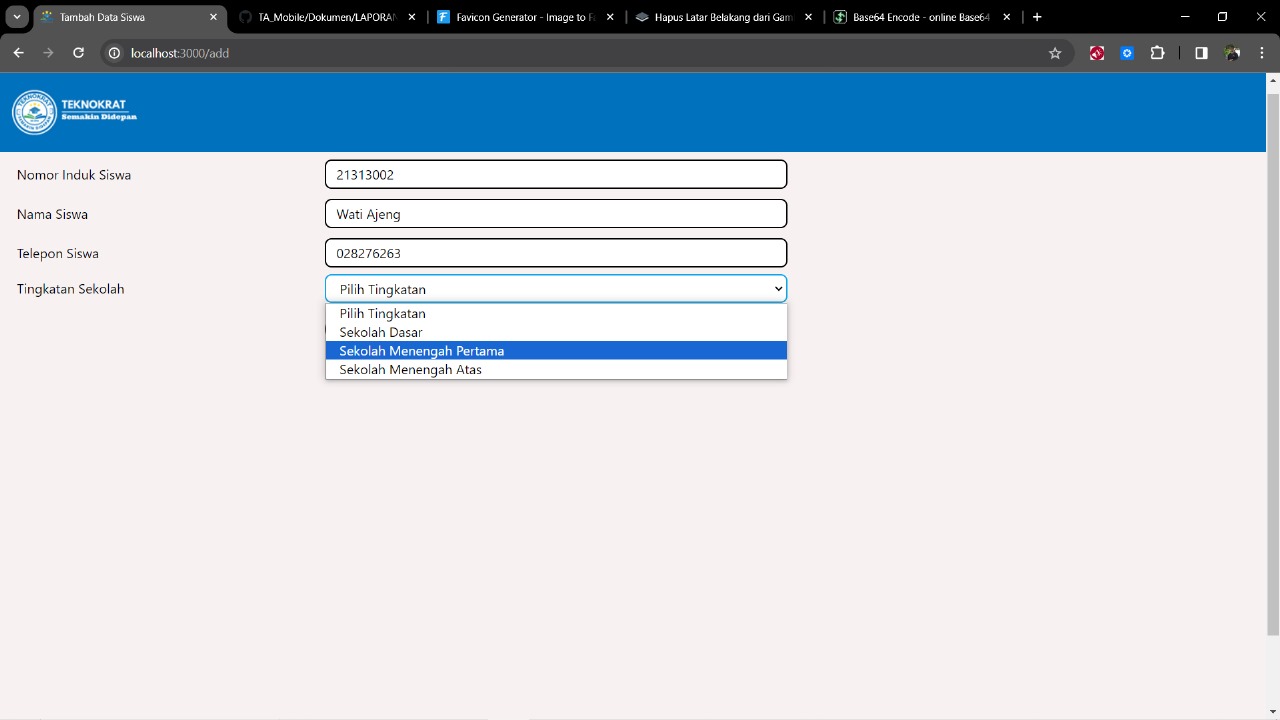
* **Tes fungsi delet pada aplikasi**

****

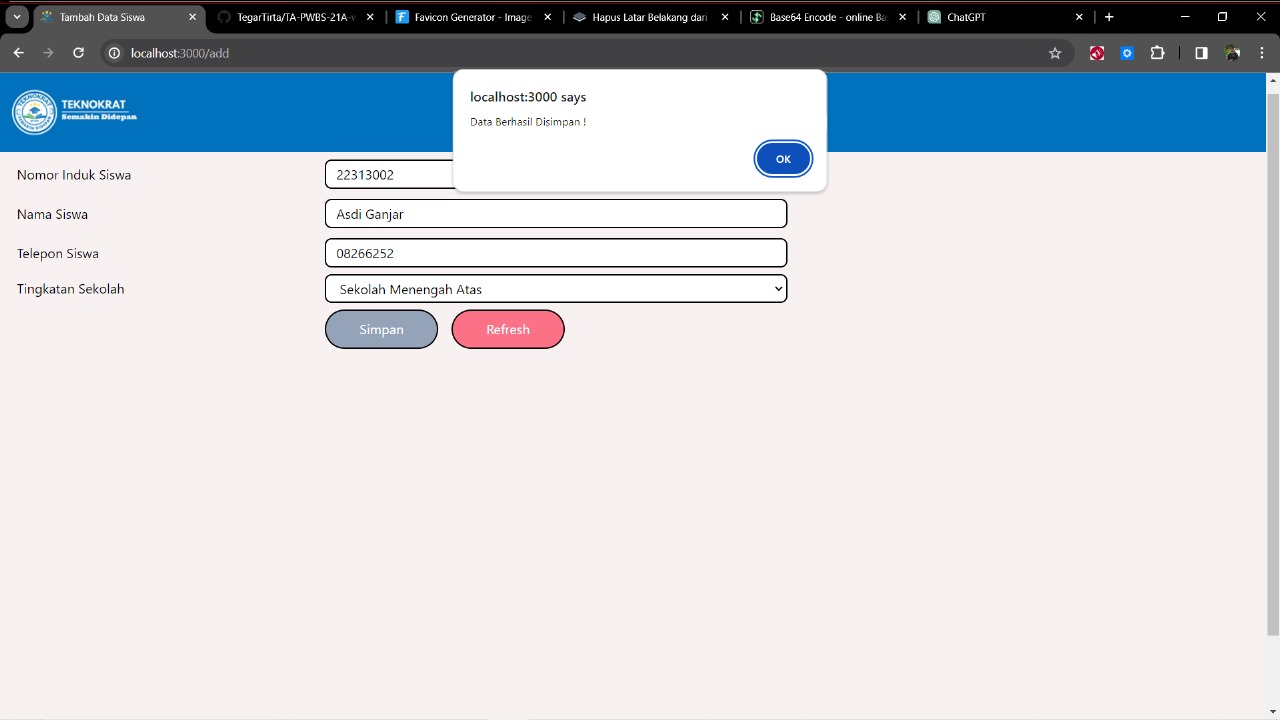
* **Tes fungsi “search” pada website**

****

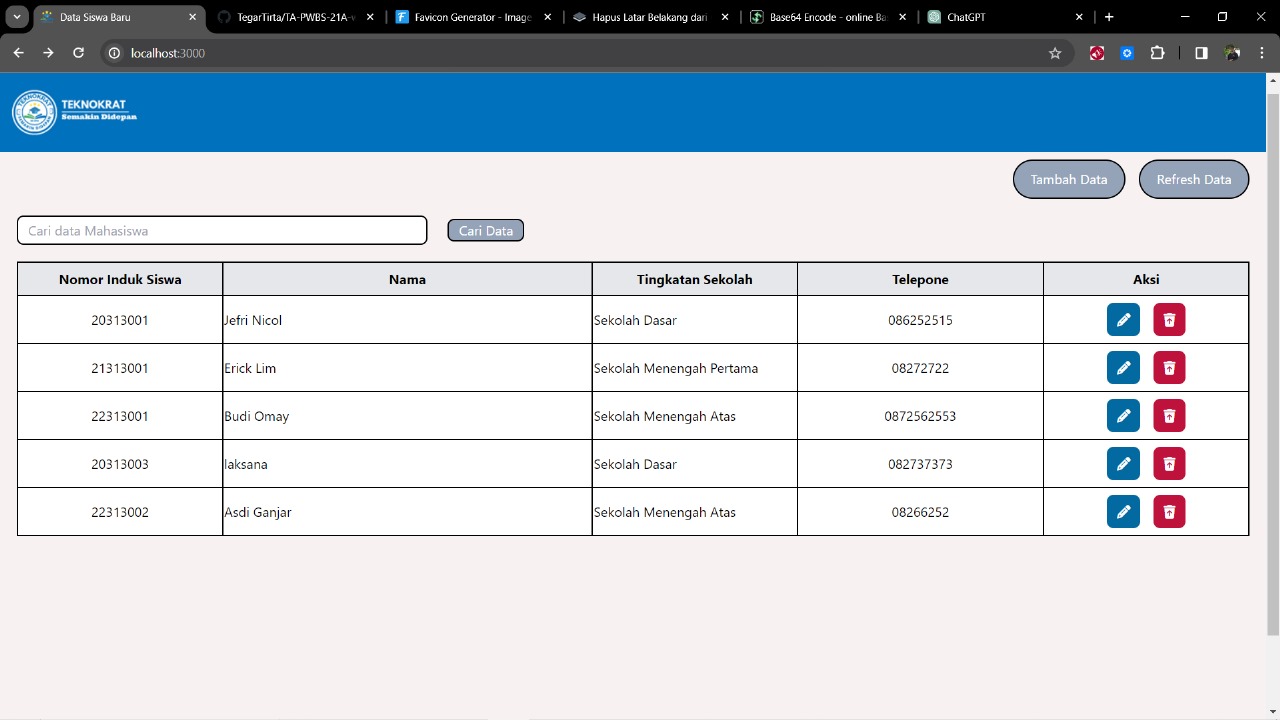
* **Tes fungsi “Tambah Data Siswa”**
* **Tes fungsi isi data pada “Tambah Data Mahasiswa”**

****

* **Berhasil Menambahkan data siswa**



* **Data telah terdaftar pada tabel siswa**

****