**LAPORAN TUGAS AKHIR KEAMANAN APLIKASI BESERTA PEJELASANNYA**



**DIBUAT OLEH:**

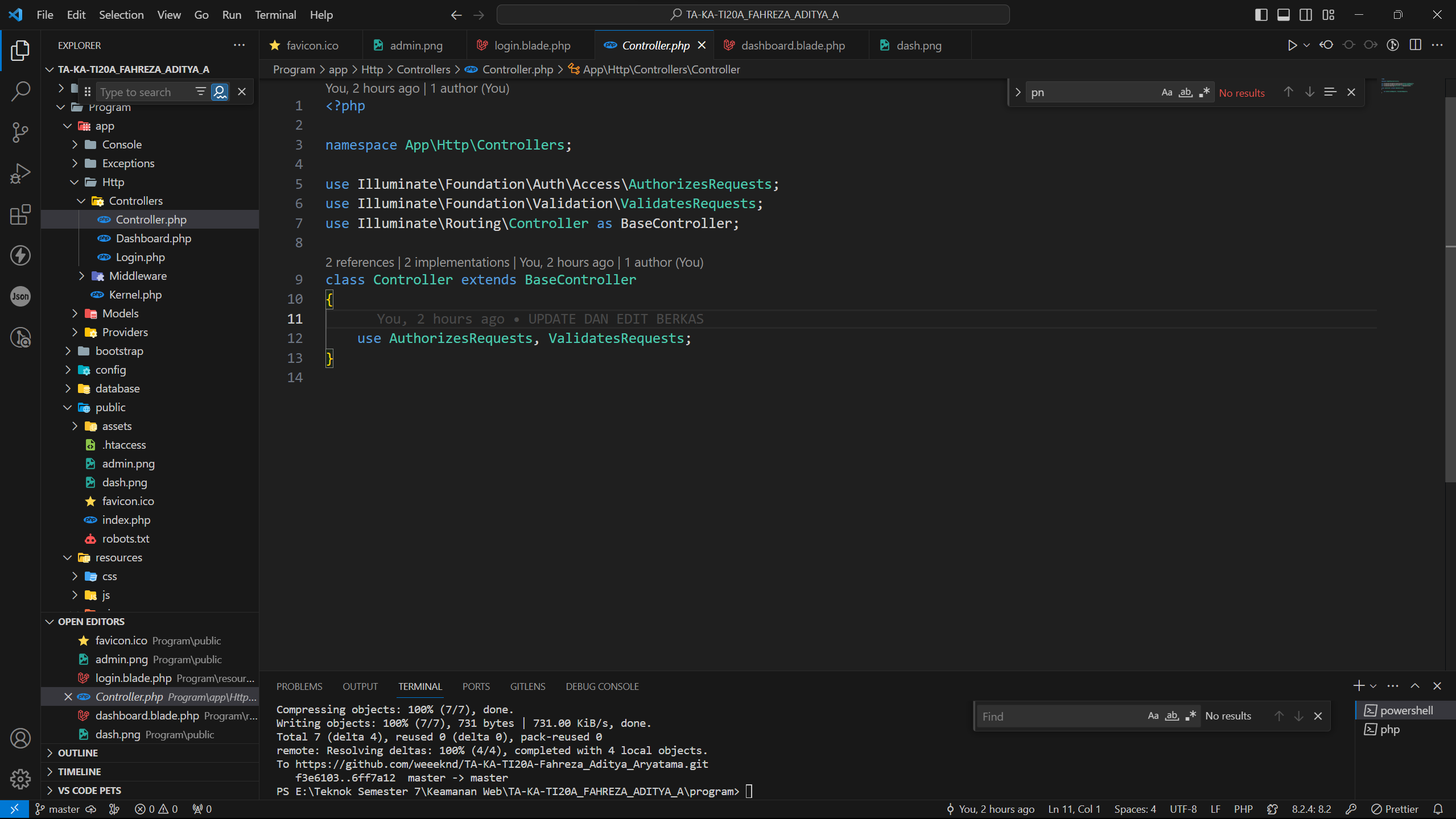
**FAHREZA ADITYA ARYATAMA – 20313009**

**UNIVERSITAS TEKNOKRAT INDONESIA**

**BANDAR LAMPUNG**

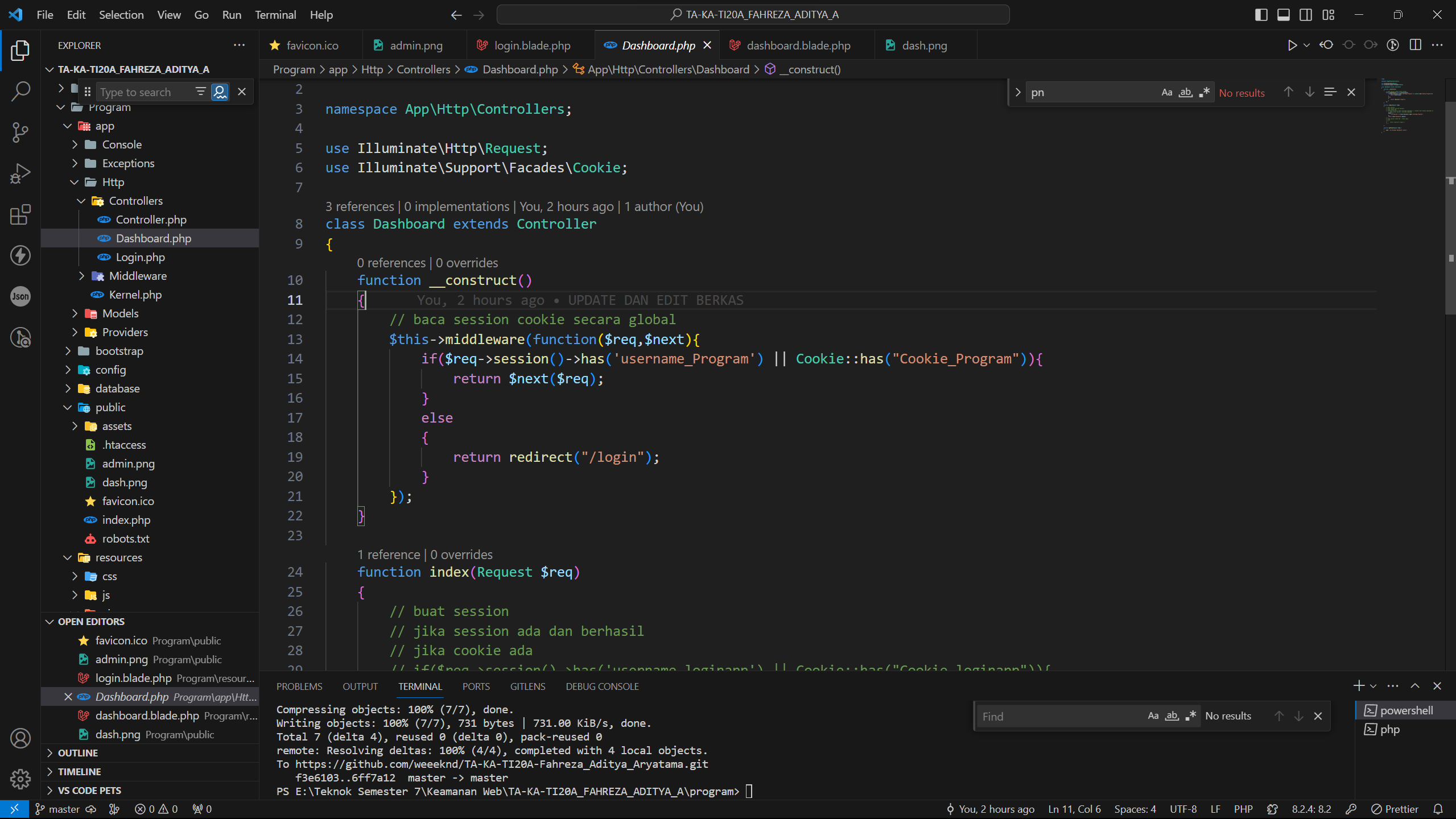
**2024**

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:

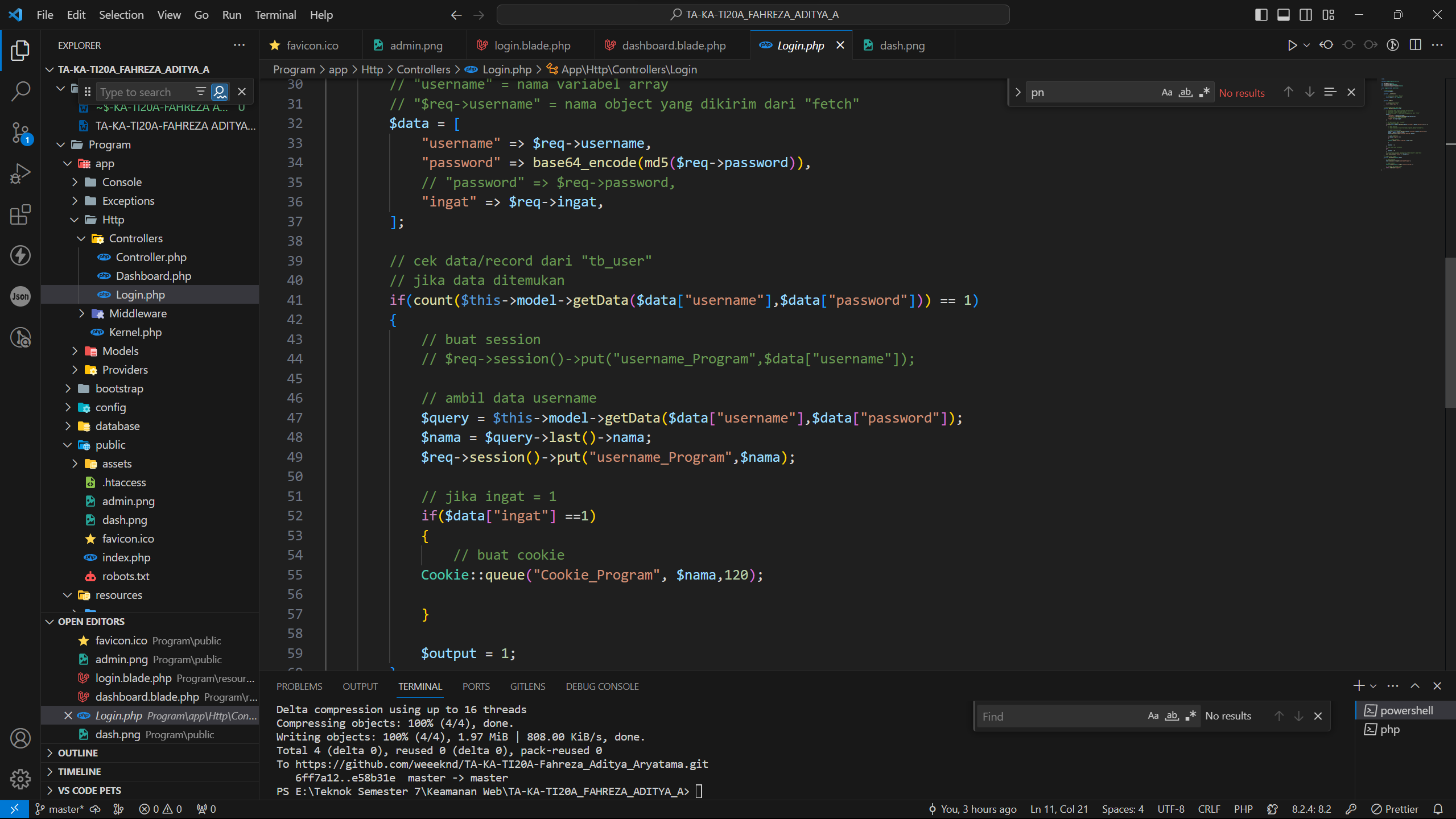
- Jika pengguna tidak memiliki session atau cookie yang valid, mereka akan diarahkan kembali ke halaman login dengan menggunakan fungsi `redirect("/login")`.

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.

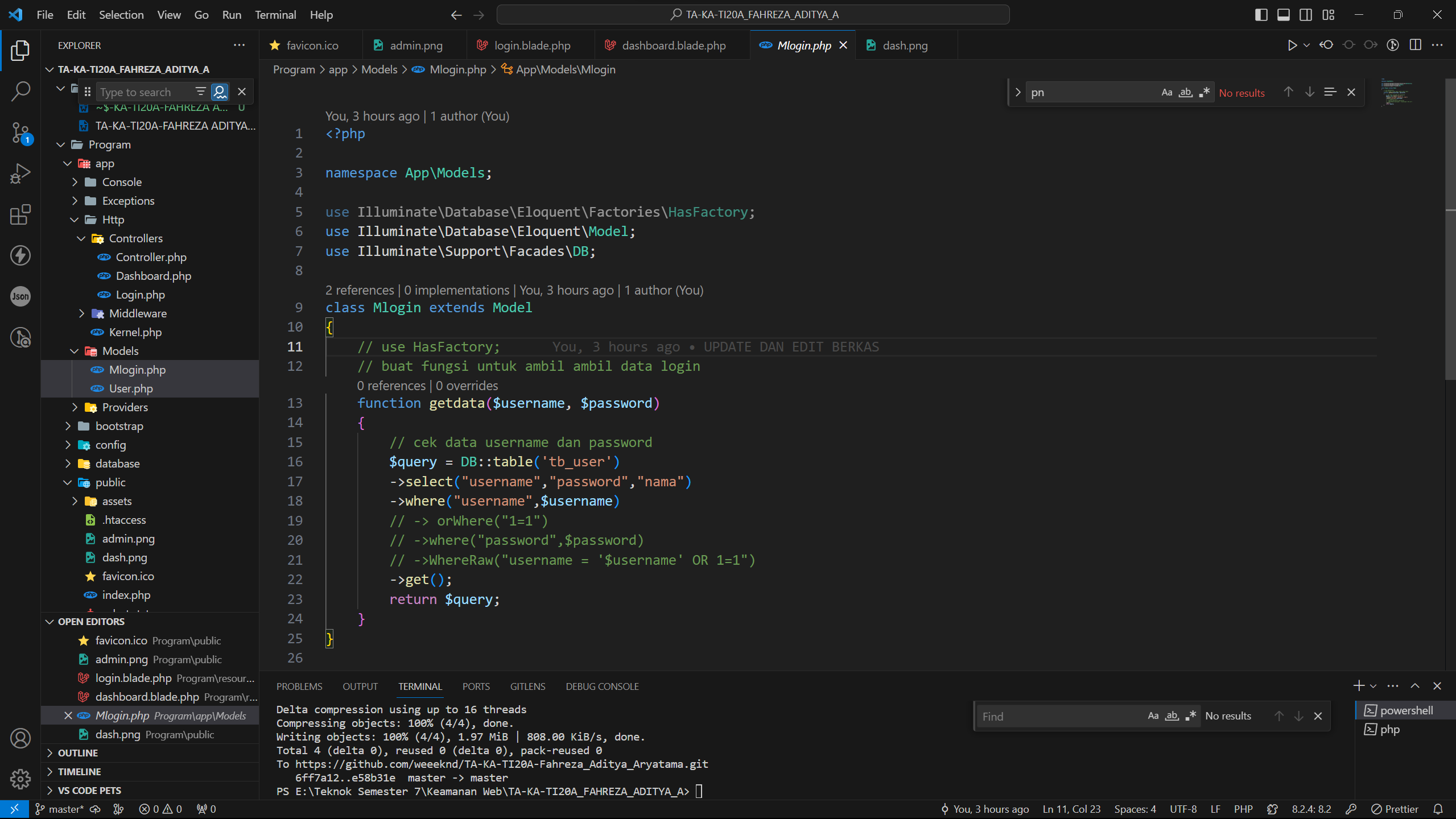
- Menghapus cookie "Cookie\_Program" jika ada.

- 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 Models**

* **File Mlogin**

****

Program pada bagian model (`Mlogin`) dalam folder `Models` bertanggung jawab untuk berinteraksi dengan database dan mengambil data login dari tabel `tb\_user`. Di bawah ini adalah deskripsi lebih rinci tentang fungsi model ini:

**Model `Mlogin`**

Model ini merupakan representasi dari tabel `tb\_user` dalam basis data. Dalam konteks aplikasi login, model ini digunakan untuk mengelola data pengguna yang akan digunakan dalam proses otentikasi. Berikut adalah komponen-komponen kunci dalam model ini:

1. Fungsi `getData`:

- Fungsi ini menerima dua parameter, yaitu `$username` dan `$password`.

- Menggunakan metode `select` untuk memilih kolom-kolom tertentu dari tabel, yaitu "username", "password", dan "nama".

- Menggunakan metode `where` untuk menentukan kondisi pencarian, di mana kita mencari data yang memiliki kolom "username" yang sama dengan nilai `$username` dan kolom "password" yang sama dengan nilai `$password`.

- Menggunakan metode `get` untuk menjalankan query dan mengambil hasilnya.

- Mengembalikan hasil query dalam bentuk objek koleksi (objek `Illuminate\Support\Collection`).

2. Catatan Tambahan:

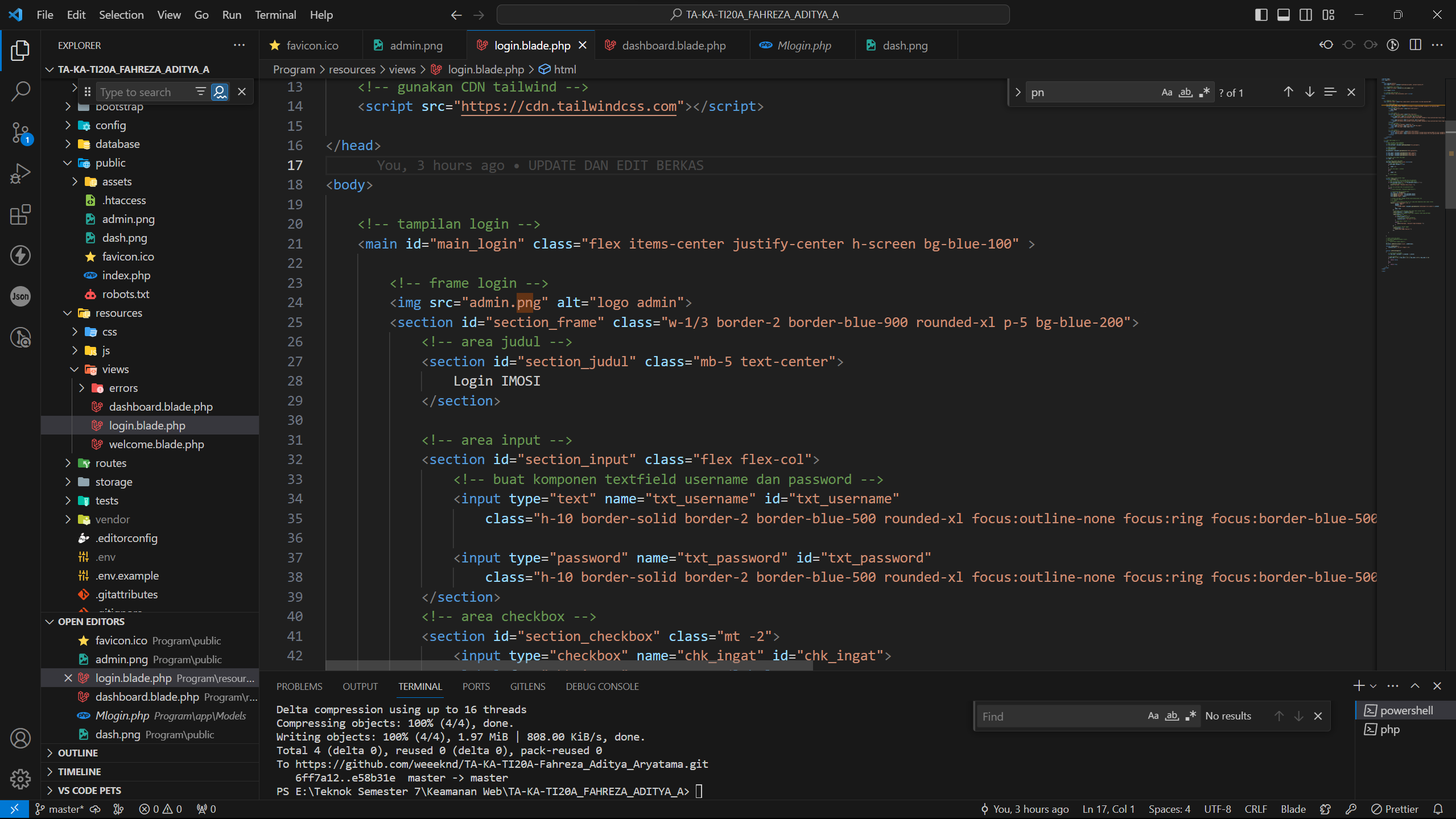
- Model ini menggunakan Query Builder Laravel (`DB::table`) untuk berinteraksi dengan database, yang merupakan cara yang umum digunakan dalam Laravel untuk melakukan operasi database.

- Meskipun model ini hanya mengambil data dari tabel, dalam aplikasi yang lebih kompleks, Anda mungkin ingin menggunakan Eloquent, ORM (Object-Relational Mapping) Laravel, untuk menyederhanakan dan memudahkan operasi database.

- Penggunaan `md5` dan `base64\_encode` pada password kemungkinan tidak disarankan untuk keamanan yang lebih baik. Sebaiknya, pertimbangkan untuk menggunakan metode hash yang lebih aman seperti bcrypt yang telah disediakan oleh Laravel. Penggunaan bcrypt dapat dilakukan pada level controller atau bahkan dalam model jika menggunakan Eloquent.

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

* **Login\_blade\_php**

****

Kode yang Anda bagikan adalah halaman login dalam format Blade pada aplikasi Laravel. Berikut adalah deskripsi dan fungsi dari beberapa bagian utama dalam kode tersebut:

Login (`login.blade.php`)

1. Head Section:

- Menetapkan karakter set, viewport, title halaman, dan menambahkan CSRF token menggunakan `{{ csrf\_token() }}` untuk melindungi formulir dari serangan Cross-Site Request Forgery.

2. Main Section:

- Membuat elemen-elemen dalam halaman menggunakan Tailwind CSS classes untuk styling.

- Menyertakan logo admin dan membuat sebuah frame login.

3. Input Fields:

- Membuat elemen input untuk username dan password.

- Menggunakan classes dari Tailwind CSS untuk memberikan styling yang responsif dan menarik.

4. Ingat Saya (Remember Me) Checkbox:

- Menambahkan checkbox "Ingat Saya" dengan elemen `<input>` dan `<label>`.

- Menggunakan event listener untuk menangkap perubahan pada checkbox dan menyimpan statusnya di variabel `ingat`.

5. Tombol Login dan Batal:

- Menambahkan dua tombol, yaitu "Login" dan "Batal".

- Tombol "Login" memiliki event listener yang memanggil fungsi `setLogin` saat diklik.

6. JavaScript Section:

- Membuat script JavaScript untuk menangani logika pada halaman login.

- Variabel JavaScript seperti `txt\_username`, `txt\_password`, `btn\_login`, `btn\_batal`, dan `chk\_ingat` digunakan untuk merepresentasikan elemen-elemen di halaman.

7. Fungsi `setLogin`:

- Mengecek apakah username dan password telah diisi. Jika tidak, menampilkan pesan kesalahan.

- Jika sudah diisi, membuat objek FormData dan menambahkan username, password, dan status "Ingat Saya" ke dalamnya.

- Menggunakan `fetch` untuk mengirim data ke endpoint `/login/get` dengan metode POST.

- Mengelola respons JSON dari server untuk menentukan apakah login berhasil atau tidak.

- Jika login berhasil (output = 1), pengguna diarahkan ke halaman utama (`'/'`), jika tidak, menampilkan pesan kesalahan.

8. Fungsi `setRefresh`:

- Dijalankan saat tombol "Batal" diklik dan mengarahkan pengguna kembali ke halaman login.

9. Fungsi `setFilterInput`:

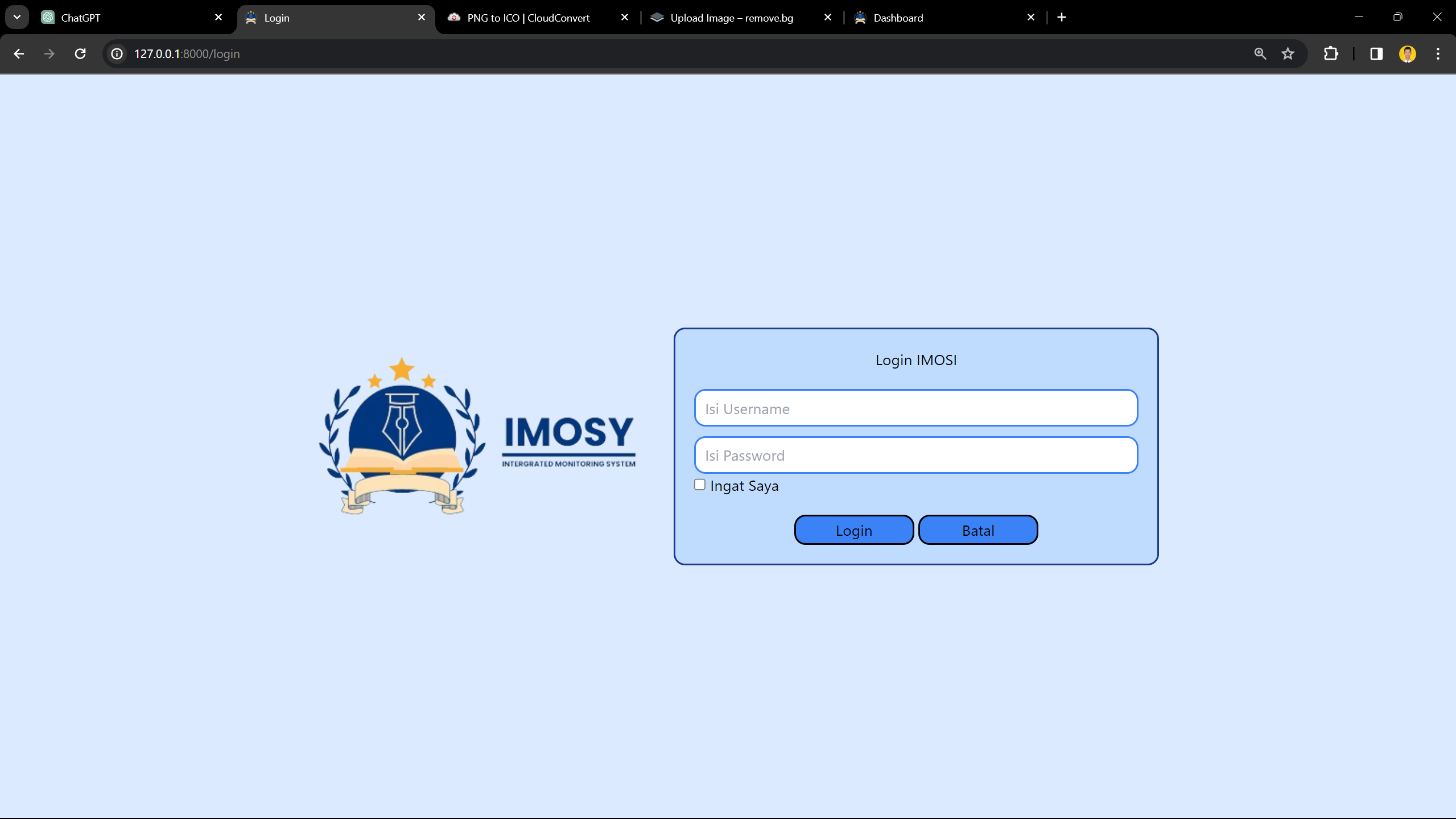
- Membatasi karakter yang dapat dimasukkan pada input, mengizinkan hanya karakter tertentu dan mencegah karakter spesial tertentu.

Catatan:

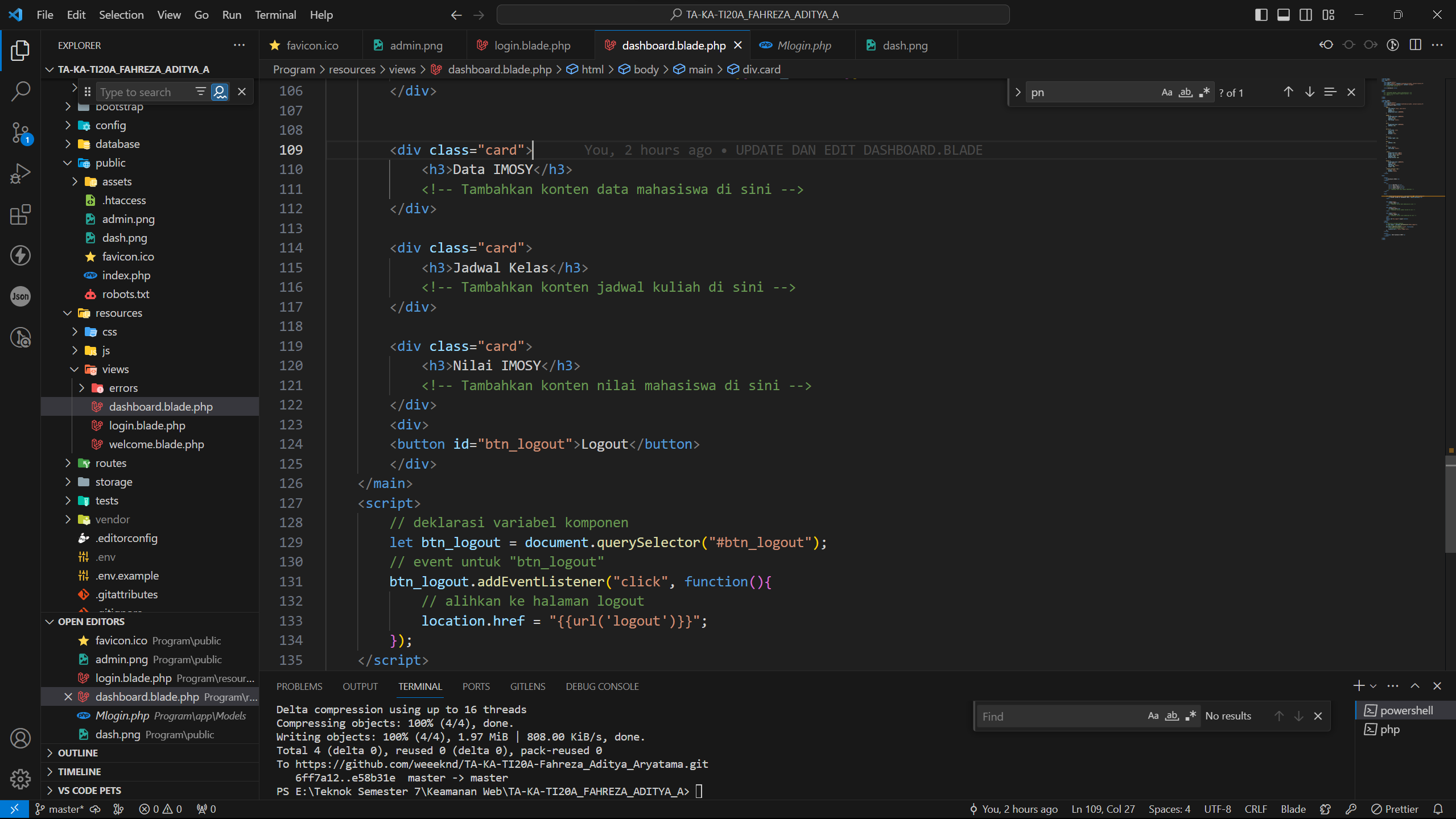
- Halaman ini menggunakan Tailwind CSS untuk styling, memberikan tampilan yang responsif dan modern.

- Kode JavaScript digunakan untuk menangani beberapa interaksi pengguna, seperti checkbox "Ingat Saya" dan pengiriman data login ke server.

* Tampilan Dari login\_blade.php



* Dashboard\_blade.php



Halaman `dashboard.blade.php` bertindak sebagai tampilan (view) dari dashboard dalam aplikasi Laravel. Berikut adalah deskripsi fungsi dan komponen utama dari kode pada halaman `dashboard.blade.php`:

1. Struktur HTML Dasar:

- Membuat struktur dasar HTML dengan tag `html`, `head`, dan `body`.

- Menetapkan karakter set dan viewport.

- Menyertakan judul halaman.

2. Styles dan Layout:

- Menyertakan beberapa gaya CSS langsung di dalam tag `<style>` untuk merancang tata letak dan tampilan halaman. Ini mencakup warna latar belakang, warna teks, dan gaya header, nav, main, dan footer.

3. Header:

- Mendefinisikan elemen `<header>` dengan judul "Dashboard IMOSI".

- Memberikan tata letak dan gaya sesuai dengan gaya CSS yang telah ditentukan.

4. Navigation Bar (`nav`):

- Menyertakan navigasi (`nav`) dengan menu yang terdiri dari beberapa item menggunakan tag `<ul>` dan `<li>`.

- Setiap item menu dirancang dengan tautan (`<a>`) yang dapat dikustomisasi sesuai dengan kebutuhan.

5. Main Content (`main`):

- Memiliki elemen `<main>` sebagai area utama konten halaman.

- Menyertakan elemen `<div>` dengan class `center-container` yang berisi gambar dan pesan selamat datang.

- Terdapat beberapa elemen `<div>` dengan class `card` yang mewakili blok-blok konten terpisah di dalam dashboard, seperti "Data IMOSY", "Jadwal Kelas", dan "Nilai IMOSY".

6. Logout Button:

- Menambahkan tombol Logout (`<button id="btn\_logout">Logout</button>`) dengan fungsi JavaScript. Saat tombol ini diklik, pengguna akan diarahkan ke halaman logout (`{{url('logout')}}`).

7. JavaScript:

- Mendeklarasikan variabel `btn\_logout` yang menangkap elemen tombol logout.

- Menambahkan event listener pada tombol logout untuk mengarahkan pengguna ke halaman logout saat tombol diklik.

8. Footer:

- Memiliki elemen `<footer>` dengan informasi hak cipta dan tahun "© 2024 Dashboard IMOSY".

9. JavaScript Block:

- Ditempatkan di akhir halaman untuk memastikan bahwa elemen HTML sudah terload sebelum script dijalankan.

Catatan:

- Halaman ini menggunakan blade templating engine, yang memungkinkan integrasi variabel PHP (seperti `{{$isi\_session}}`) dan logika pemrograman di dalam HTML.

- CSS inline digunakan untuk menentukan gaya halaman secara langsung.

- Script JavaScript digunakan untuk memberikan fungsionalitas pada tombol logout.

* Tampilan dari dashboard\_blade.php

