**LAPORAN PROJEK AKHIR KEAMANAN APLIKASI**

****

**Disusun oleh**

**Yeris Ari Sandi – 20313007**

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

**UNIVERSITAS TEKNOKRAT INDONESIA**

**TAHUN 2023/2024**

Contents

[Tampilan halaman login 3](#_Toc157045757)

[Tampilan Dashbord 5](#_Toc157045758)

[Favicon 7](#_Toc157045759)

[User di Models 8](#_Toc157045760)

[Stock di Models 10](#_Toc157045761)

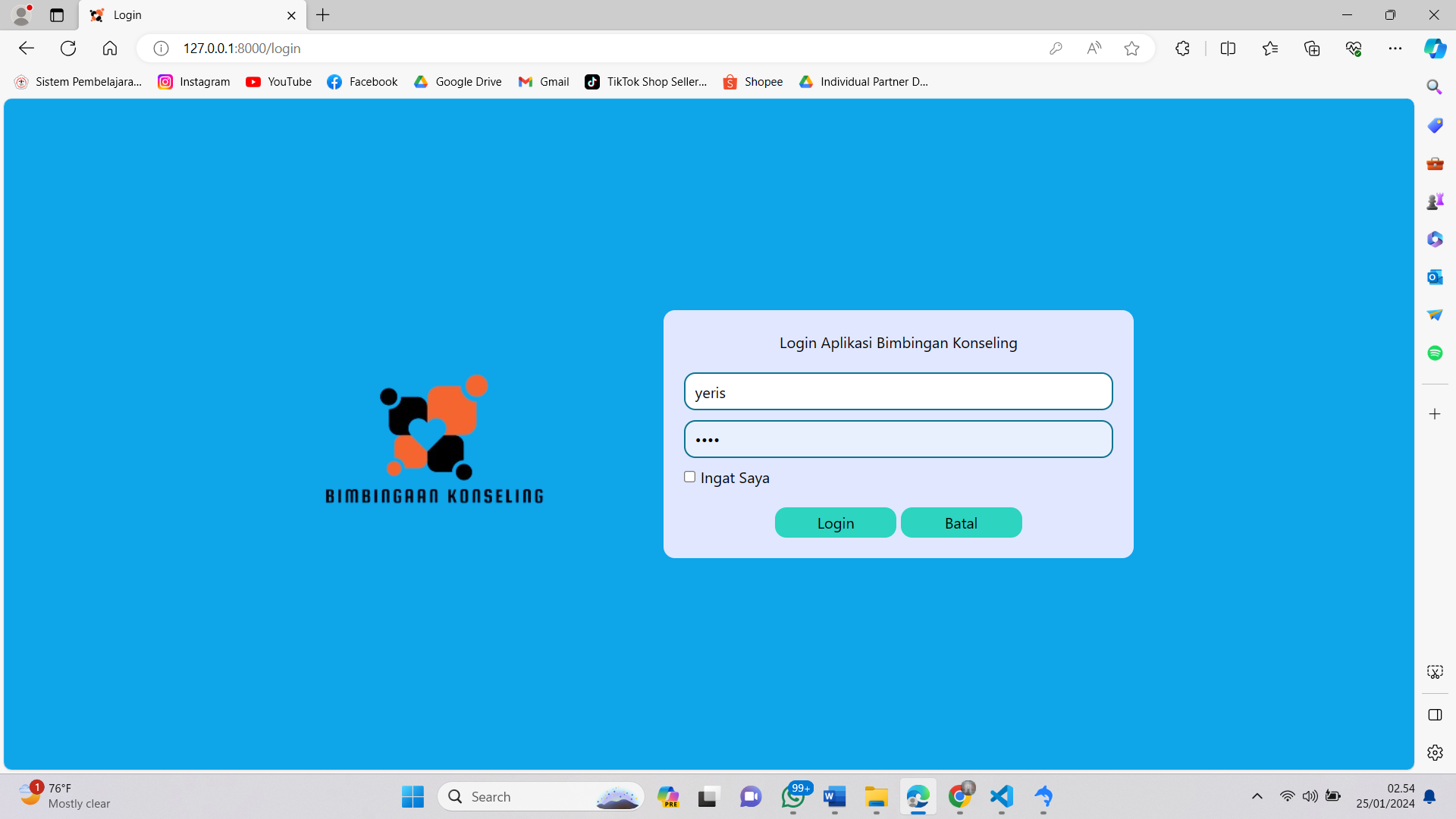
[Mlogin di Models 11](#_Toc157045762)

[Stokcontroller di controller 13](#_Toc157045763)

[Login di controller 14](#_Toc157045764)

[Dashbord di controller 16](#_Toc157045765)

# Tampilan halaman login



1. **Tailwind CSS:**
   * menggunakan Tailwind CSS melalui CDN untuk mendesain tampilan halaman. Pastikan sudah memahami kelas-kelas Tailwind CSS yang digunakan.
2. **CSRF Token:**
   * telah menambahkan meta tag untuk csrf token. Ini penting untuk melindungi aplikasi dari serangan CSRF. Pastikan token ini digunakan dengan benar dalam permintaan POST.

<meta name="csrf\_token" content="{{ csrf\_token() }}">

1. **Event Handler:**
   * menggunakan JavaScript untuk menangani event seperti klik tombol login dan batal. Fungsi **setLogin** akan mengirim data ke server menggunakan **fetch** ketika tombol login ditekan. Pastikan server Laravel memiliki endpoint yang sesuai.

function setLogin() { // ... (sebagaimana dijelaskan dalam komentar di kode) }

1. **Checkbox Ingat Saya:**
   * menangani event klik pada checkbox "Ingat Saya" dan menyimpan nilai statusnya ke dalam variabel **ingat**.

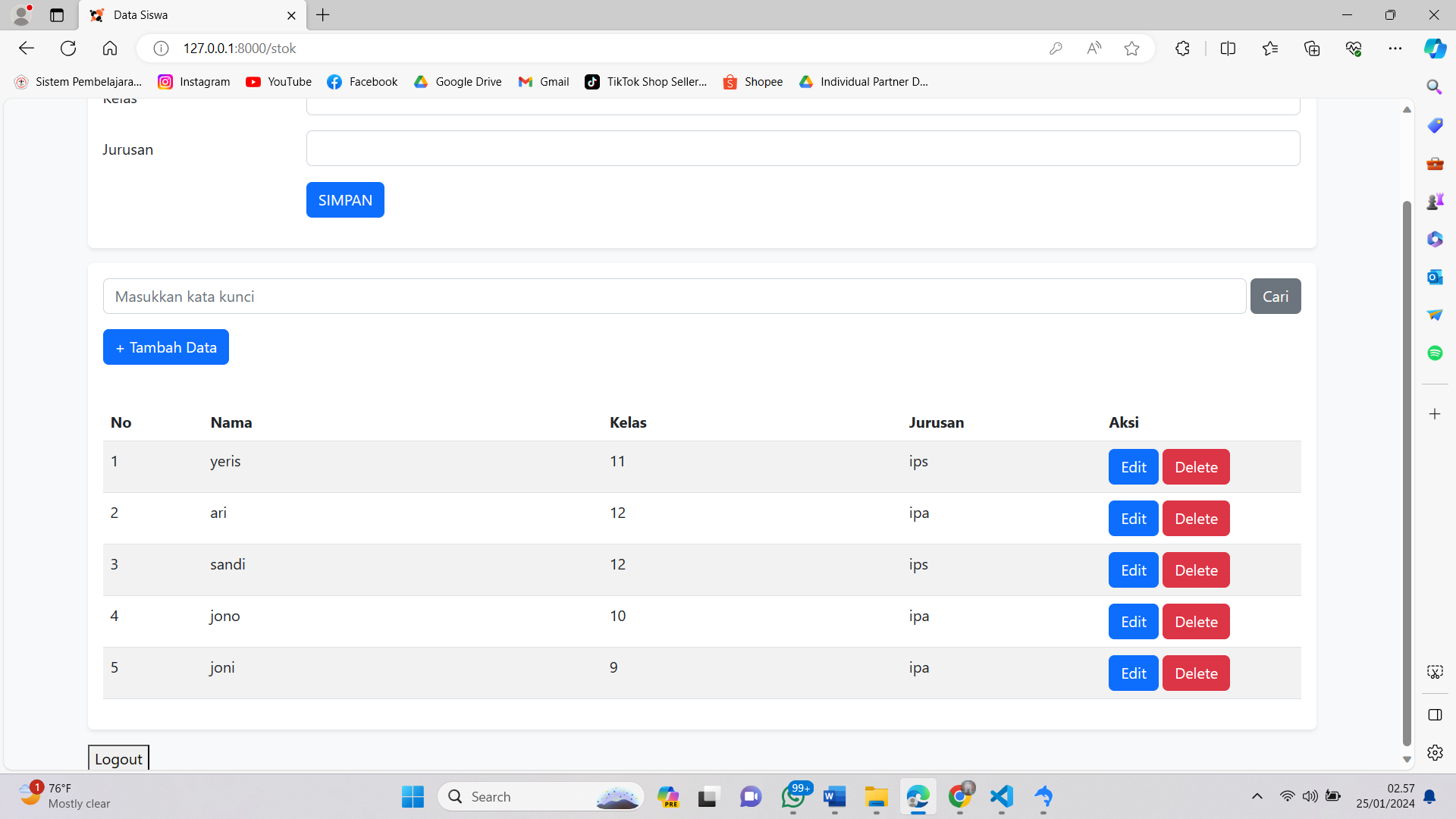
chk\_ingat.addEventListener("click", function(){ // ... });

1. **Fungsi Set Refresh:**
   * Fungsi **setRefresh** mengarahkan pengguna kembali ke halaman login saat tombol "Batal" ditekan.

function setRefresh() { location.href = "{{ url('/login') }}"; }

1. **Fungsi Set Filter Input:**
   * Fungsi **setFilterInput** digunakan untuk membatasi karakter tertentu dalam inputan, seperti "=" atau spasi.

# Tampilan Dashbord



1. **Form Input Data:**
   * Terdapat sebuah form untuk memasukkan data siswa, namun tipe input "nama" pada elemen input pertama sebaiknya diganti menjadi "text".

<input type="text" class="form-control" name='nim' id="nim">

1. **Form Pencarian:**
   * Terdapat form pencarian yang menggunakan metode GET dengan parameter "katakunci". Pastikan endpoint untuk menangani pencarian sudah disiapkan pada sisi server.
2. **Tombol Tambah Data:**
   * Tombol "+ Tambah Data" mengarahkan ke suatu URL (link kosong **'href=''**). Pastikan link tersebut mengarahkan ke halaman yang sesuai untuk menambahkan data siswa baru.

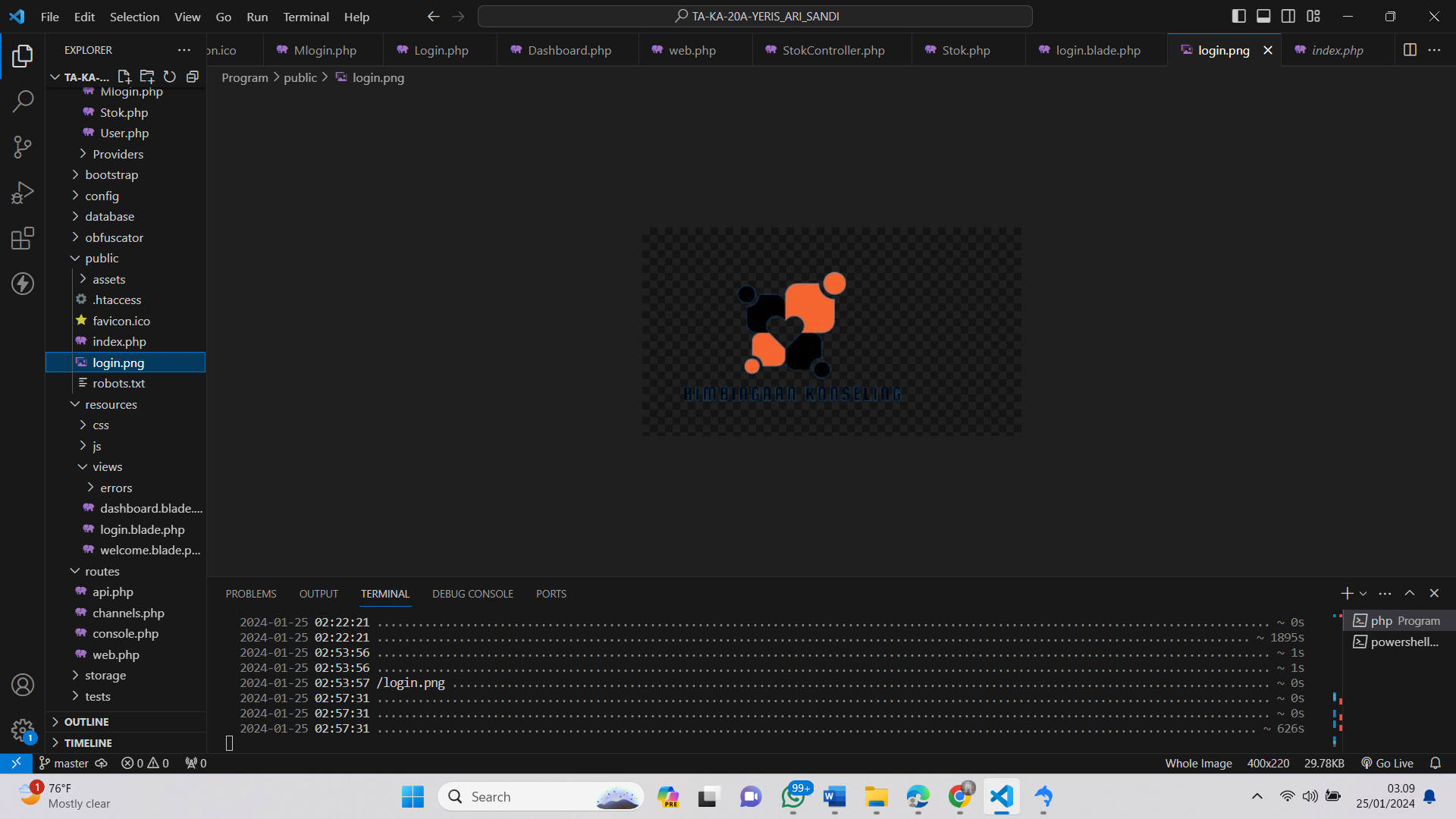
<a href='' class="btn btn-primary">+ Tambah Data</a>

1. **Tabel Data Siswa:**
   * Terdapat tabel yang menampilkan data siswa. Bagian ini menggunakan looping **@foreach** untuk menampilkan data dari variabel **$stoks**.
2. **Button Logout:**
   * Terdapat sebuah tombol "Logout" yang memiliki event click untuk mengarahkan ke halaman logout. Pastikan halaman logout tersedia dan telah diimplementasikan.

<button id="btn\_logout">Logout</button>

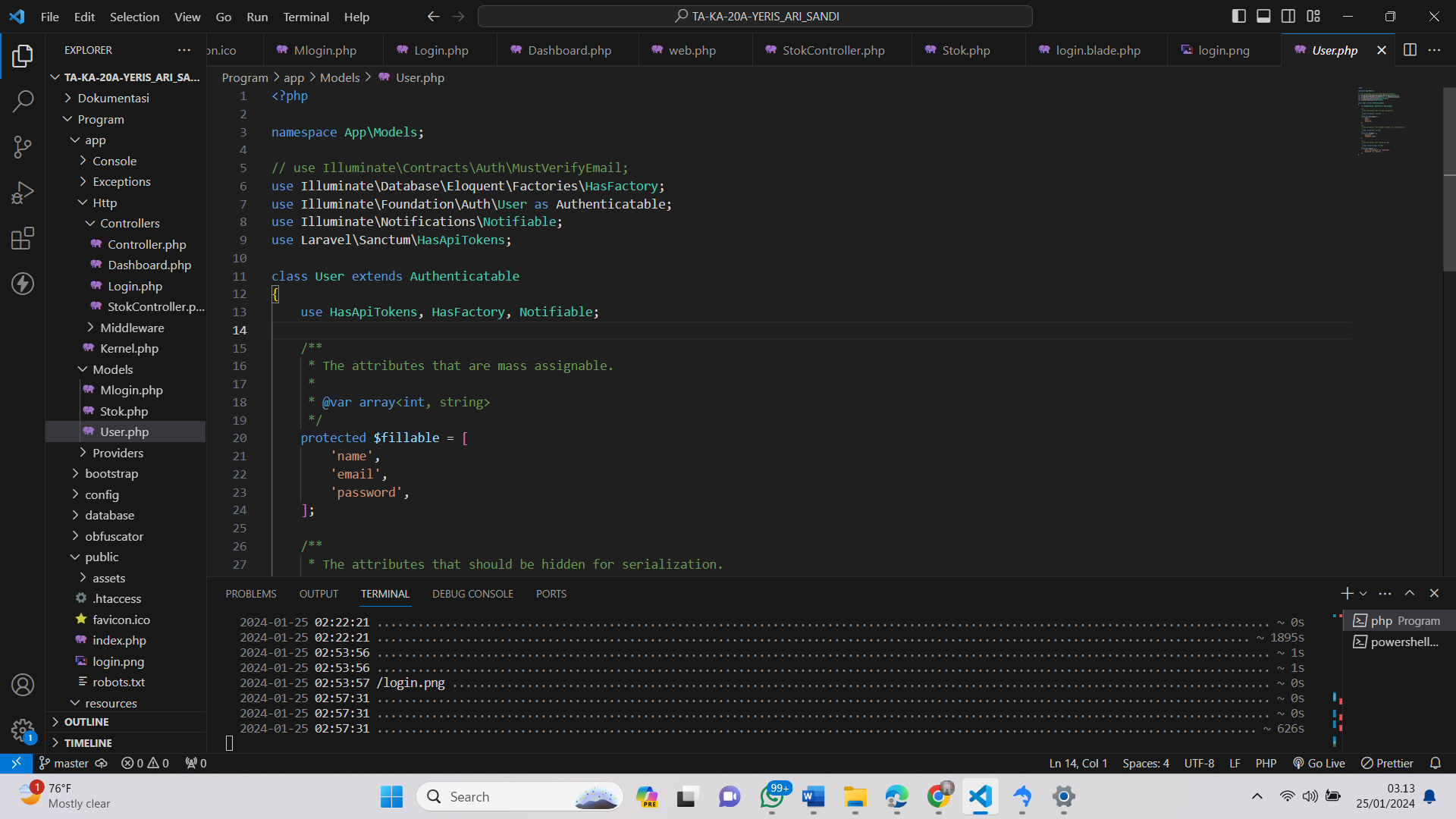
1. **Script JavaScript:**
   * Ada juga script JavaScript yang menangani event klik pada tombol logout.

# Favicon



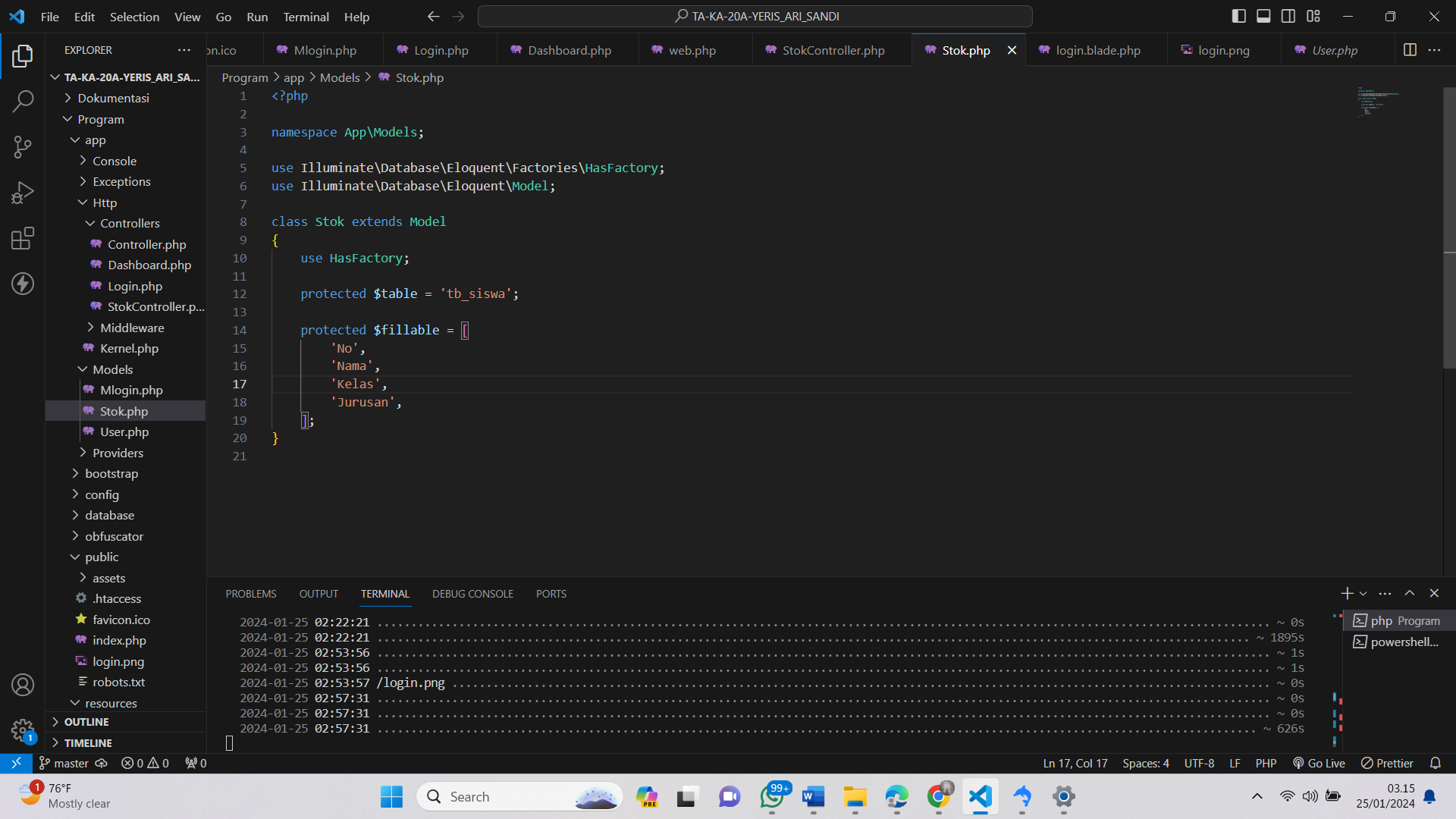
Favicon, singkatan dari "favorite icon," adalah ikon kecil yang digunakan untuk mengidentifikasi suatu situs web atau halaman web dalam antarmuka pengguna peramban web. Favicon biasanya muncul di tab peramban, bookmark, dan dalam sejumlah konteks lain di mana identifikasi visual singkat diperlukan.

# User di Models



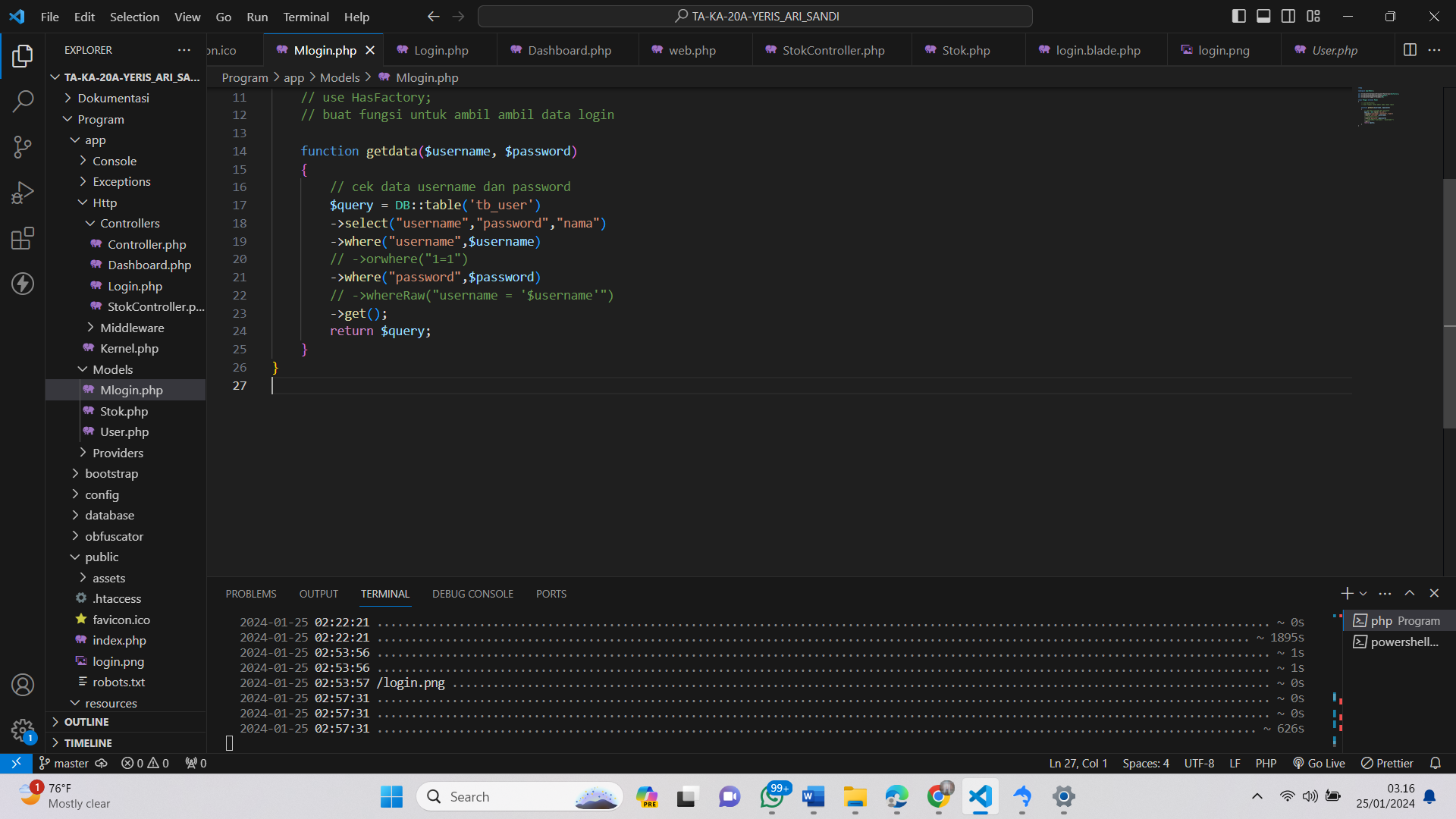
1. **Autentikasi Pengguna:**
   * Model **User** menyediakan fitur autentikasi pengguna. Dengan menggunakan model ini, Anda dapat melakukan operasi autentikasi seperti login dan logout.
2. **HasApiTokens:**
   * Trait **HasApiTokens** memberikan model kemampuan untuk mengeluarkan dan mengelola token API menggunakan Laravel Sanctum atau Passport. Token API ini dapat digunakan untuk otentikasi pengguna dalam penggunaan API.
3. **HasFactory:**
   * Trait **HasFactory** memungkinkan model untuk menggunakan factory, yang digunakan untuk menghasilkan data tes atau seeding database dengan data dummy.
4. **Notifiable:**
   * Trait **Notifiable** memungkinkan model untuk mengirim notifikasi. Dalam konteks autentikasi, notifikasi sering digunakan untuk mengirim email verifikasi atau reset password.
5. **fillable:**
   * Properti **$fillable** menentukan kolom-kolom dalam tabel pengguna yang dapat diisi secara massal. Ini diperlukan ketika menyimpan data pengguna dalam database.
6. **hidden:**
   * Properti **$hidden** menentukan kolom-kolom yang disembunyikan saat objek model di-serialisasi. Pada umumnya, informasi sensitif seperti password tidak akan ditampilkan dalam respons JSON.
7. **casts:**
   * Properti **$casts** digunakan untuk mengkonversi tipe data kolom. Pada contoh ini, **email\_verified\_at** di-cast ke tipe **datetime**, dan **password** di-cast sebagai hashed. Hashing password dilakukan untuk meningkatkan keamanan.

# Stock di Models



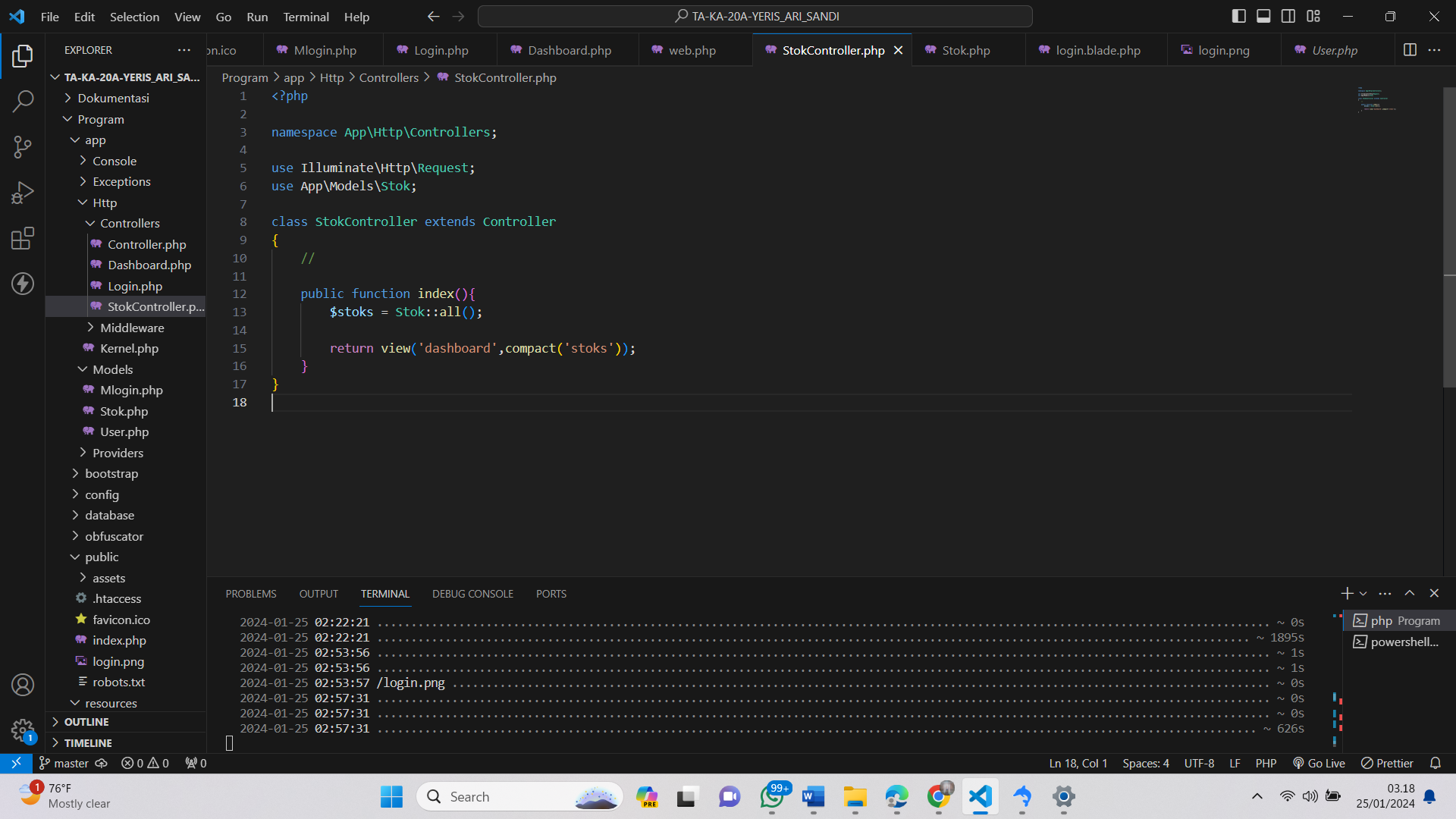
1. **Namespace:**
   * Model berada dalam namespace **App\Models**, sesuai dengan struktur direktori default Laravel.
2. **Traits:**
   * Menggunakan trait **HasFactory** untuk memfasilitasi penggunaan factory dalam menghasilkan data tes atau seeding database.
3. **$table Property:**
   * Property **$table** menentukan nama tabel dalam database yang akan diakses oleh model. Dalam kasus ini, model ini terhubung dengan tabel **tb\_siswa**.
4. **$fillable Property:**
   * Property **$fillable** menentukan kolom-kolom dalam tabel **tb\_siswa** yang dapat diisi secara massal. Atribut yang terdaftar di sini dapat diisi ketika menyimpan data baru atau memperbarui data menggunakan metode **create** atau **update** pada model.

# Mlogin di Models



1. **Namespace:**
   * Model ini berada dalam namespace **App\Models**, sesuai dengan konvensi direktori default Laravel.
2. **Fungsi getdata:**
   * Fungsi **getdata** berfungsi untuk mengambil data pengguna berdasarkan username dan password yang diberikan.
3. **Metode Pemanggilan Database:**
   * Menggunakan Query Builder Laravel (**DB::table**) untuk membangun query yang diinginkan.
4. **Query Builder:**
   * Membuat query SQL yang memilih kolom **username**, **password**, dan **nama** dari tabel **tb\_user** yang memenuhi kondisi tertentu.
5. **Parameter Input:**
   * Fungsi menerima dua parameter, yaitu **$username** dan **$password**, yang digunakan sebagai kondisi dalam query untuk mencocokkan data login.
6. **Hasil Query:**
   * Mengembalikan hasil query menggunakan metode **get()**. Hasil query ini dapat berupa objek koleksi (collection) yang berisi data pengguna yang sesuai dengan kriteria pencarian.

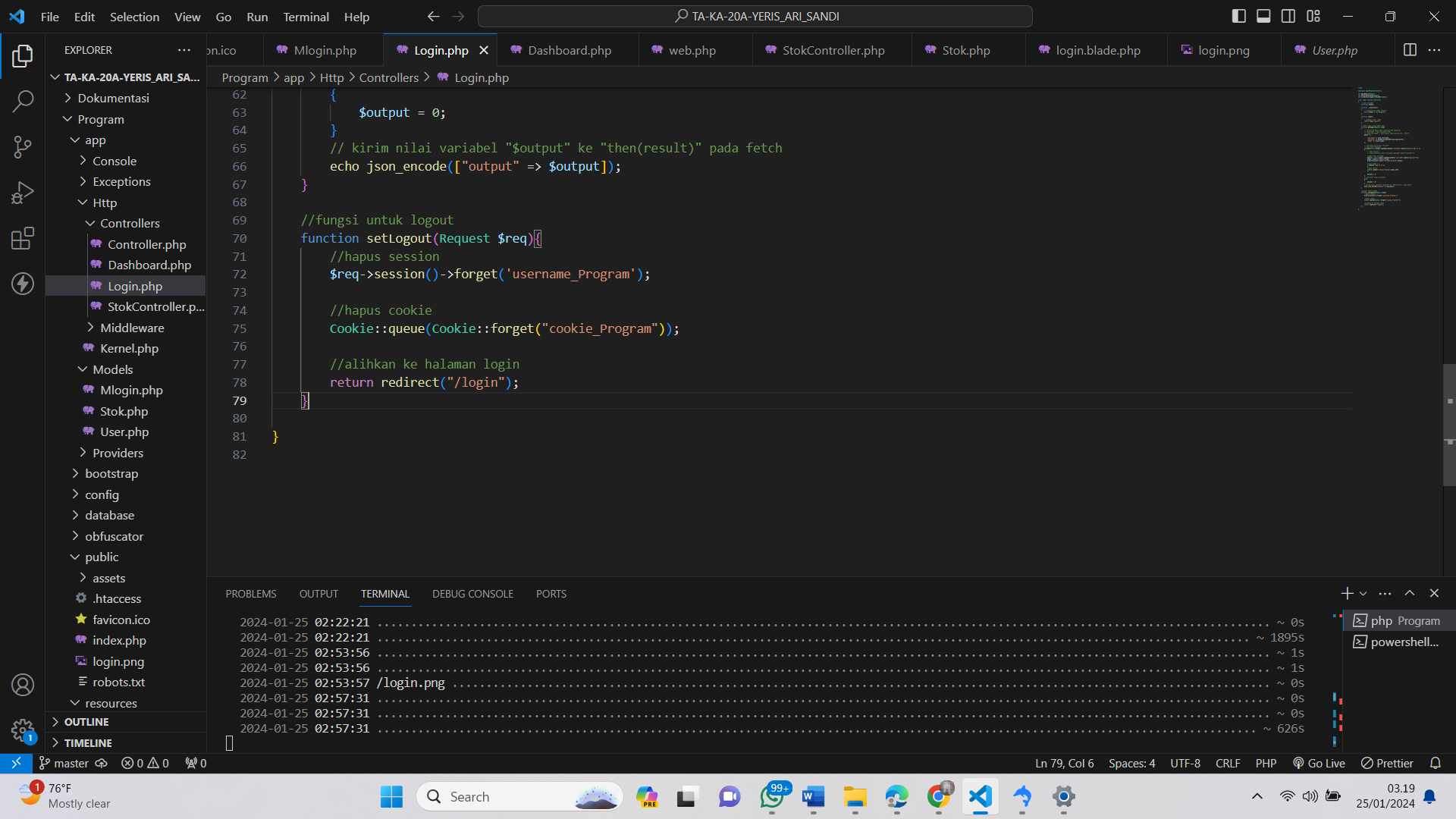
# Stokcontroller di controller



1. **Namespace:**
   * Controller ini berada dalam namespace **App\Http\Controllers**, sesuai dengan konvensi direktori default Laravel.
2. **Use Statement:**
   * Menggunakan **use Illuminate\Http\Request** untuk mengimpor kelas Request dari Laravel. Ini dibutuhkan untuk mengakses data dari permintaan HTTP.
3. **Menggunakan Model:**
   * Menggunakan model **Stok** dengan pernyataan **use App\Models\Stok**. Model ini digunakan untuk berinteraksi dengan data stok dalam database.
4. **Fungsi index:**
   * Fungsi **index** digunakan untuk menampilkan semua data stok dari model **Stok**. Data stok kemudian dikirimkan ke view 'dashboard' untuk ditampilkan.
5. **View Rendering:**
   * Menggunakan fungsi **view** untuk merender view 'dashboard' dan menyertakan data stok dalam variabel **$stoks**.

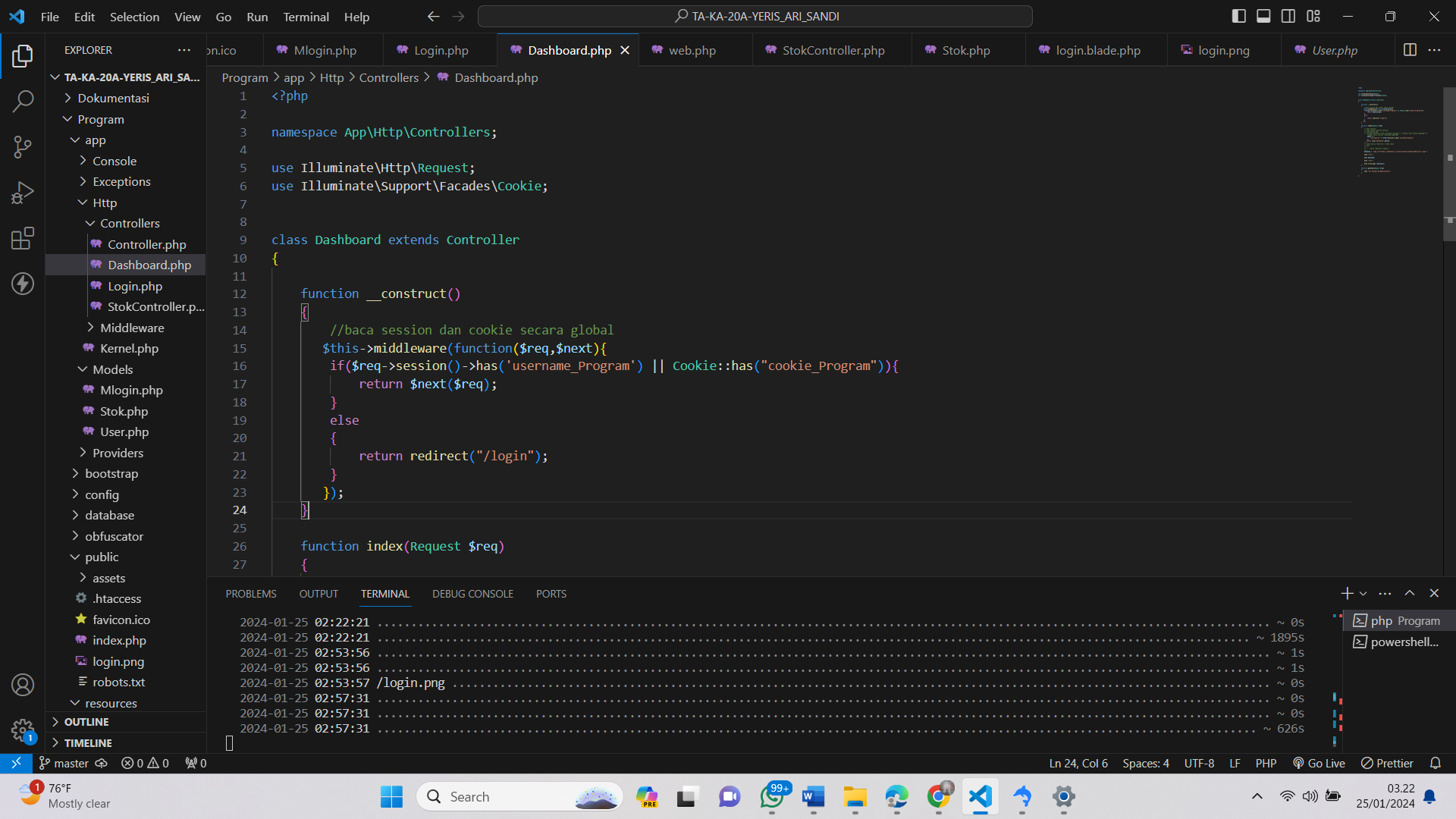
Controller ini dihubungkan dengan view dan mungkin diakses melalui route tertentu dalam aplikasi Laravel. Pada umumnya, controller berperan sebagai perantara antara model dan view, dan digunakan untuk mengelola logika aplikasi terkait data stok.

# Login di controller

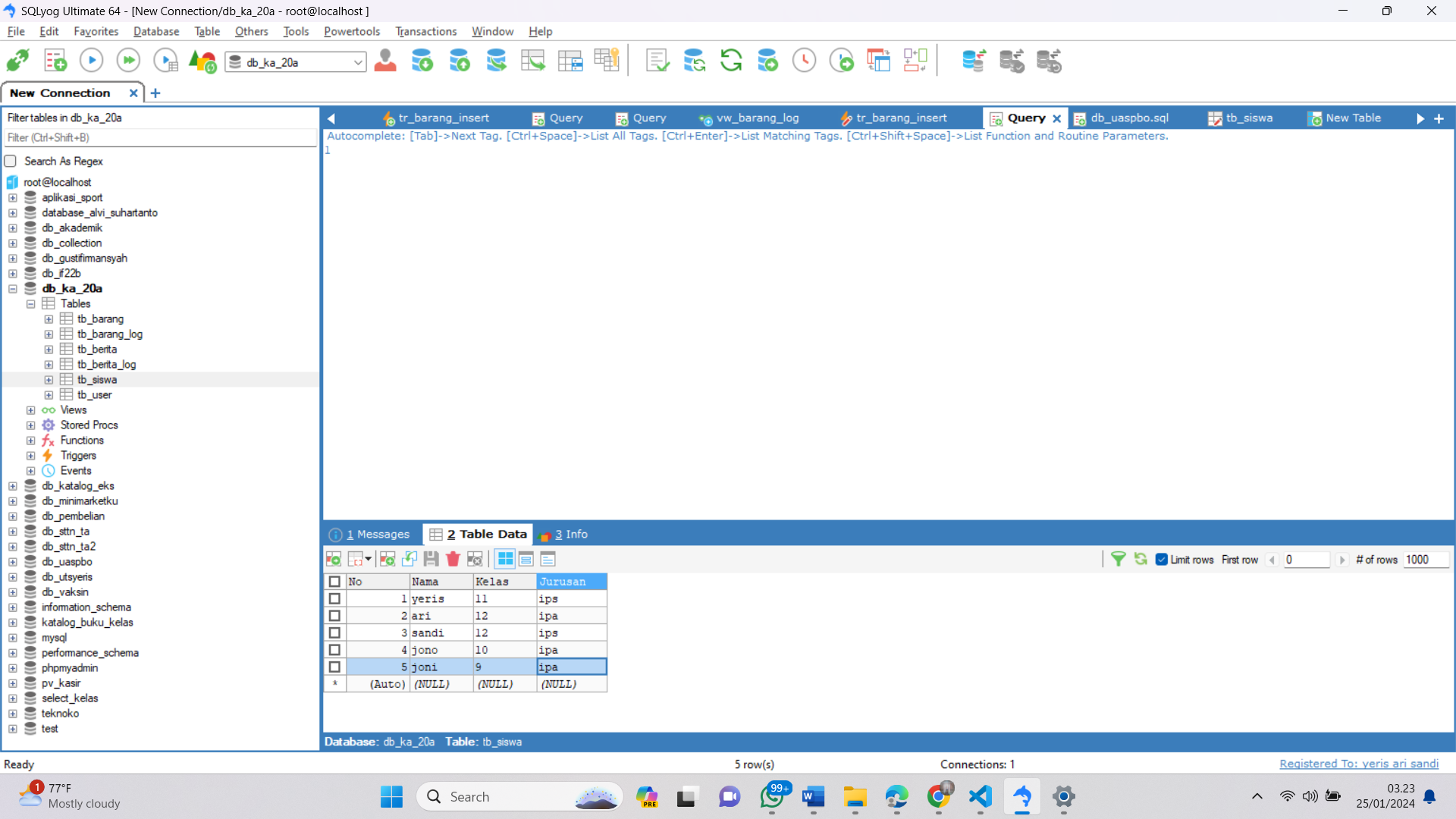


1. **Namespace:**
   * Controller ini berada dalam namespace **App\Http\Controllers**, sesuai dengan konvensi direktori default Laravel.
2. **Menggunakan Model:**
   * Menggunakan model **Mlogin** dengan inisialisasi di dalam konstruktor. Model ini mungkin dirancang untuk melakukan operasi terkait data login.
3. **Fungsi Konstruktor:**
   * Konstruktor menginisialisasi model **Mlogin**, memungkinkan controller untuk menggunakan metode yang didefinisikan dalam model tersebut.
4. **Fungsi index:**
   * Fungsi **index** menangani permintaan untuk menampilkan halaman login. Mengembalikan view dengan nama "login".
5. **Fungsi getLogin:**
   * Fungsi **getLogin** menangani permintaan login. Menerima data username, password, dan opsi "Ingat Saya" dari permintaan HTTP.
   * Menggunakan model **Mlogin** untuk memeriksa data login. Jika data ditemukan, membuat session dengan nama "username\_Program" dan mungkin membuat cookie jika opsi "Ingat Saya" diaktifkan.
   * Mengembalikan hasil dalam format JSON ke bagian frontend aplikasi.
6. **Fungsi setLogout:**
   * Fungsi **setLogout** menangani permintaan logout. Menghapus session "username\_Program" dan cookie "cookie\_Program", lalu mengarahkan pengguna kembali ke halaman login.

# Dashbord di controller



1. **Konstruktor (\_\_construct):**
   * Konstruktor digunakan untuk inisialisasi, dalam hal ini, untuk membaca session dan cookie secara global.
   * Middleware yang didefinisikan di dalam konstruktor memastikan bahwa pengguna hanya dapat mengakses halaman dashboard jika sesi "username\_Program" ada atau jika cookie "cookie\_Program" ada.
2. **Fungsi index:**
   * Fungsi ini menangani permintaan untuk menampilkan halaman dashboard.
   * Memeriksa apakah sesi "username\_Program" ada atau apakah cookie "cookie\_Program" ada. Jika salah satu dari keduanya ada, pengguna dianggap sudah login.
   * Jika pengguna sudah login, data sesi diambil dan disertakan dalam view "dashboard". Jika pengguna belum login, mereka diarahkan kembali ke halaman login.
   * Terdapat contoh penggunaan variabel **$display** yang berisi tag gambar HTML. Output dari variabel ini kemudian di-echo dan di-strip tag HTML-nya sebelum ditampilkan.
3. **Fungsi getView:**
   * Fungsi ini mungkin dirancang untuk menangani permintaan khusus terkait tampilan dashboard. Namun, dalam contoh ini, hanya mencetak teks "Ini halaman dashboard(view)".
4. Table data siswa



1. Table data user login

