

# **LAPORAN PRAKTIKUM PEMROGRAMAN WEB**

## **PRAKTIKUM 13 – AUTHENTICATION**



Disusun oleh:

Nama : Alfath Roziq Widhayaka

Nim : L0122012

Kelas : A

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA**

**UNIVERSITAS SEBELAS MARET**

**2024**

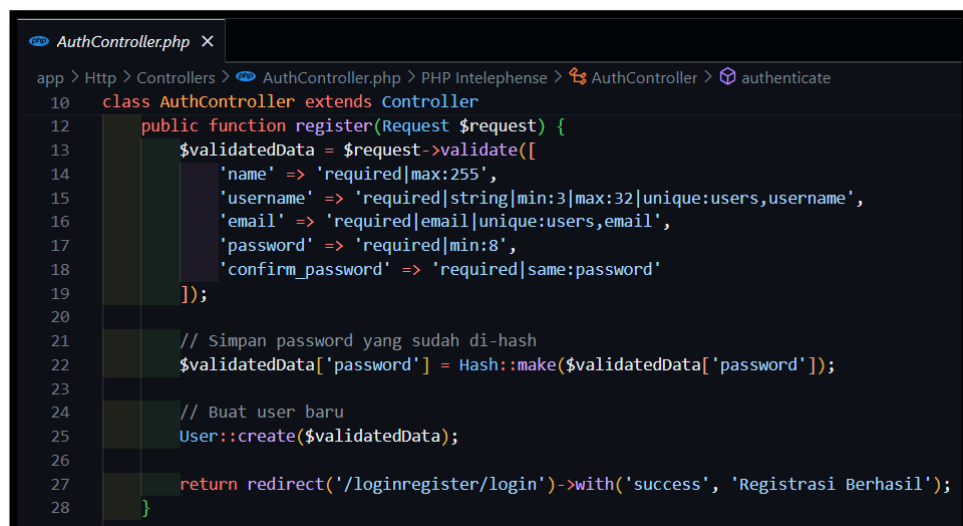
# BAB I

## A. Penjelasan Source Code

Pada praktikum minggu ke 11 ini, saya membuat fitur **User Authentication** menggunakan Laravel yang mencakup proses **Login, Register, dan Logout**. Fitur **Login** memungkinkan pengguna untuk mengakses sistem dengan memvalidasi identitas menggunakan username/email dan password yang terdaftar. Sementara itu, fitur **Register** memungkinkan pengguna untuk membuat akun baru dengan mengisi informasi yang diperlukan seperti nama, email, dan password. Setelah berhasil login, pengguna dapat menggunakan layanan yang memerlukan otentikasi, dan ketika selesai, fitur **Logout** memungkinkan mereka untuk keluar dari sesi aktif mereka dengan aman. Implementasi ketiga fitur ini memastikan keamanan dan kenyamanan pengguna dalam mengelola identitas dan akses ke dalam sistem. Berikut adalah penjelasan source code nya.

### 1. Membuat Controller Authentication

- Register function

A screenshot of a code editor showing the 'register' function in 'AuthController.php'. The code is in PHP and uses Laravel's validation and user creation features. It includes comments in Indonesian explaining the steps: validating input, hashing the password, creating a new user, and redirecting to the login page upon success.

```
10 class AuthController extends Controller
12     public function register(Request $request) {
13         $validatedData = $request->validate([
14             'name' => 'required|max:255',
15             'username' => 'required|string|min:3|max:32|unique:users,username',
16             'email' => 'required|email|unique:users,email',
17             'password' => 'required|min:8',
18             'confirm_password' => 'required|same:password'
19         ]);
20
21         // Simpan password yang sudah di-hash
22         $validatedData['password'] = Hash::make($validatedData['password']);
23
24         // Buat user baru
25         User::create($validatedData);
26
27         return redirect('/loginregister/login')->with('success', 'Registrasi Berhasil');
28     }
```

*Gambar 1 register function*

Fungsi **Register** ini bertanggung jawab untuk menangani pendaftaran pengguna baru. Fungsi ini menerima input dari permintaan HTTP (Request), kemudian memvalidasi data yang diberikan menggunakan aturan yang telah ditetapkan. Validasi mencakup pengecekan nama, username, email, password, dan konfirmasi password. Jika data valid, password akan di-hash menggunakan **Hash::make** sebelum disimpan ke dalam database. Setelah itu, pengguna baru dibuat

menggunakan model User dan data yang telah divalidasi. Terakhir, fungsi ini mengarahkan pengguna ke halaman login dengan pesan sukses.

- **Login Function**

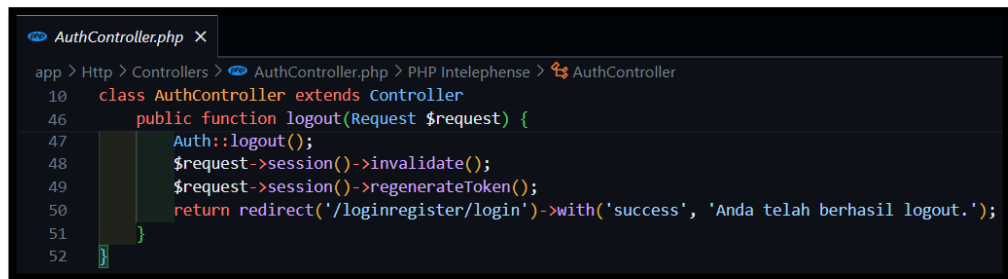
A screenshot of a code editor window titled 'AuthController.php'. The editor shows the following PHP code:

```
10 class AuthController extends Controller
30     public function authenticate(Request $request) {
31         $credentials = $request->validate([
32             'username' => 'required|string',
33             'password' => 'required',
34         ]);
35
36         if(Auth::attempt($credentials)) {
37             $request->session()->regenerate();
38
39             // Redirect ke halaman films.index setelah login berhasil
40             return redirect()->intended(route('films.index'));
41         }
42
43         return back()->with('failed', 'Username atau password salah.');
```

*Gambar 2 login function*

Fungsi **Login** ini menangani proses otentikasi atau login pengguna. Fungsi ini menerima input dari permintaan HTTP dan memvalidasi data yang diberikan untuk memastikan username dan password disediakan. Setelah validasi, fungsi mencoba melakukan otentikasi dengan kredensial yang diberikan menggunakan **Auth::attempt**. Jika otentikasi berhasil, sesi pengguna akan diregenerasi untuk keamanan tambahan dan pengguna diarahkan ke halaman yang dimaksud (default-nya adalah halaman indeks film). Jika otentikasi gagal, pengguna dikembalikan ke halaman sebelumnya dengan pesan kesalahan.

## ▪ Logout Function



```
AuthController.php X
app > Http > Controllers > AuthController.php > PHP Intelephense > AuthController
10 class AuthController extends Controller
46 public function logout(Request $request) {
47     Auth::logout();
48     $request->session()->invalidate();
49     $request->session()->regenerateToken();
50     return redirect('/loginregister/login')->with('success', 'Anda telah berhasil logout.');
```

Gambar 3 logout function

Fungsi **Logout** ini menangani proses logout pengguna. Fungsi ini menggunakan metode **Auth::logout** untuk menghapus sesi autentikasi pengguna saat ini. Setelah itu, sesi pengguna di-invalidasi dan token sesi diregenerasi untuk menghindari serangan CSRF. Terakhir, pengguna diarahkan kembali ke halaman login dengan pesan sukses bahwa mereka telah berhasil logout.

## 2. Melengkapi Route



```
web.php X
routes > web.php > ...
1 <?php
2
3 use App\Http\Controllers\AuthController;
4 use App\Http\Controllers\FilmController;
5 use Illuminate\Support\Facades\Route;
6
7 // Redirect ke halaman login sebagai halaman pertama
8 Route::get('/', function () {
9     return redirect()->route('login');
10 });
11
12 // Route untuk halaman login dan register
13 Route::get('/loginregister/login', function () {
14     return view('loginregister.login');
15 }->name('login');
16
17 Route::post('/login', [AuthController::class, 'authenticate']->name('login.authenticate'));
18
19 Route::get('/loginregister/register', function () {
20     return view('loginregister.register');
21 });
22
23 Route::post('/register', [AuthController::class, 'register']->name('register'));
```

Gambar 4 web.php

Pada route pertama, **Route::get('/', function () { return redirect()->route('login'); });**, didefinisikan bahwa saat pengguna mengakses root URL (/), mereka akan secara otomatis diarahkan (redirect) ke halaman login. Ini dilakukan dengan memanggil fungsi anonim yang mengembalikan redirect ke route yang diberi nama **'login'**. Route

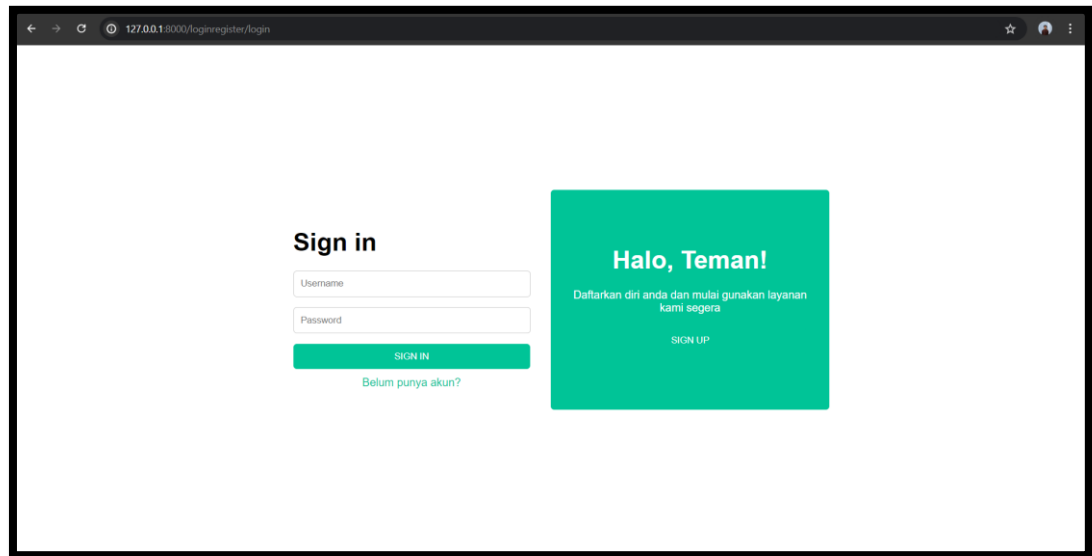
ini berguna untuk memastikan bahwa pengguna diarahkan langsung ke halaman login ketika mengakses aplikasi tanpa memberikan URL yang lebih spesifik.

Untuk mengelola halaman login dan register, terdapat beberapa route lainnya. Route **`Route::get('/loginregister/login', function () { return view('loginregister.login'); })->name('login');`** mendefinisikan bahwa saat pengguna mengakses URL `/loginregister/login`, mereka akan disajikan tampilan (view) `loginregister.login`, yang biasanya merupakan halaman login. Route ini juga diberi nama 'login', yang memudahkan penggunaan redirect dan rujukan dalam kode. Route **`Route::post('/login', [AuthController::class, 'authenticate'])->name('login.authenticate');`** menangani permintaan POST ke `/login`, yang memanggil metode `authenticate` dari `AuthController` untuk memproses login pengguna. Selain itu, route **`Route::get('/loginregister/register', function () { return view('loginregister.register'); });`** dan **`Route::post('/register', [AuthController::class, 'register'])->name('register');`** masing-masing menangani penyajian halaman registrasi dan pemrosesan data registrasi yang dimasukkan pengguna. Semua route ini bekerja bersama untuk mengelola alur autentikasi pengguna, dari menampilkan form login dan registrasi hingga memproses data yang dimasukkan.

## BAB II

### A. Hasil Output

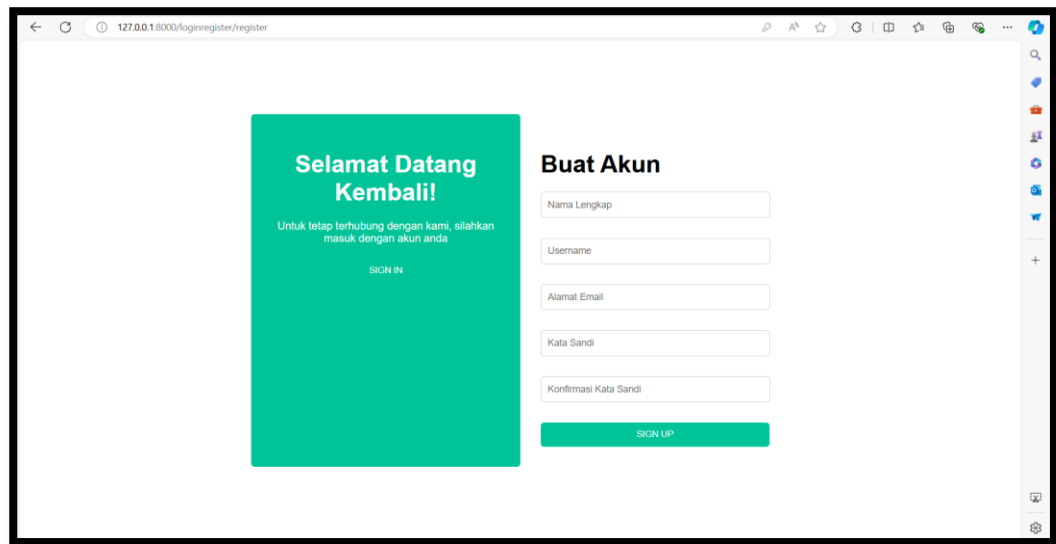
#### 1. Halaman Login (Awal)



*Gambar 5 halaman login*

Gambar diatas merupakan tampilan ketika user masuk ke halaman awal pada aplikasi laravel ini. Gambar diatas menampilkan halaman login untuk user melakukan login, untuk user yang sudah memiliki akun diharapkan mengisi username dan passwordnya. Jika user tidak memiliki akun, user dapat menekan tombol **Sign Up** atau **Belum punya akun?** Yang nantinya akan diarahkan menuju halaman register.

## 2. Halaman Register



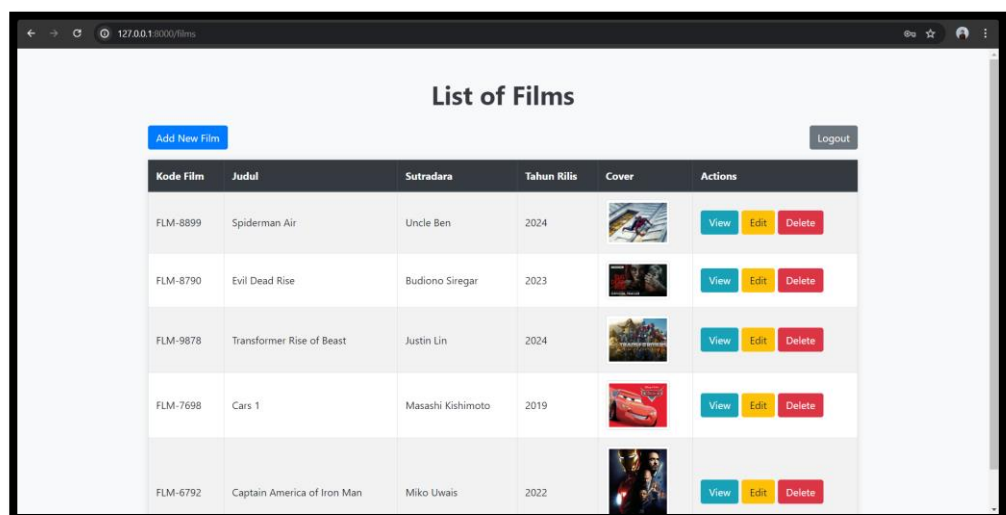
The screenshot shows a web browser window with the URL `127.0.0.1:8000/loginregister/register`. The page has a white background. On the left, there is a green rectangular box with the text "Selamat Datang Kembali!" in white, followed by "Untuk tetap terhubung dengan kami, silahkan masuk dengan akun anda" and a "SIGN IN" button. On the right, under the heading "Buat Akun", there is a form with five input fields: "Nama Lengkap", "Username", "Alamat Email", "Kata Sandi", and "Konfirmasi Kata Sandi". Below these fields is a green "SIGN UP" button. The browser's address bar and various icons are visible at the top.

*Gambar 6 halaman register*






Gambar diatas merupakan tampilan ketika user ingin melakukan registrasi akun. Pada tampilan ini user diwajibkan mengisi semua kolom seperti **Nama Lengkap**, **Username**, **Email**, dan **Kata Sandi**. Jika sudah maka pilih tombol **SIGN UP**. Ketika user sudah membuat akun, user dapat kembali ke halaman login dan melakukan login.

## 3. Fungsi Logout

### 1. Sebelum Logout



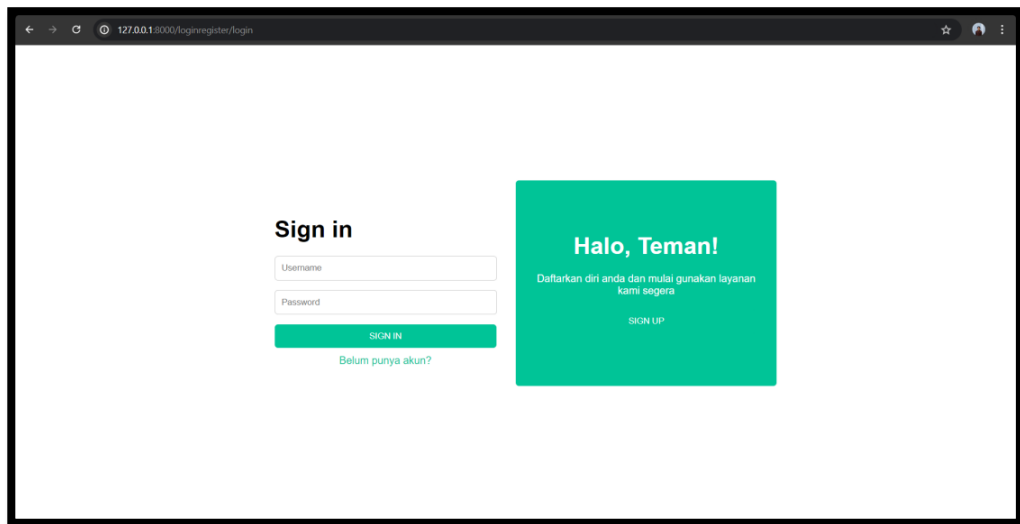
The screenshot shows a web browser window with the URL `127.0.0.1:8000/films`. The page has a light gray background. At the top, there is a "List of Films" heading. Below it, there is a table with the following columns: "Kode Film", "Judul", "Sutradara", "Tahun Rilis", "Cover", and "Actions". The table contains five rows of movie data. In the top right corner, there is a "Logout" button. In the top left corner, there is a blue button labeled "Add New Film".

Kode Film	Judul	Sutradara	Tahun Rilis	Cover	Actions
FLM-8899	Spiderman Air	Uncle Ben	2024		<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
FLM-8790	Evil Dead Rise	Budiono Siregar	2023		<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
FLM-9878	Transformer Rise of Beast	Justin Lin	2024		<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
FLM-7698	Cars 1	Masashi Kishimoto	2019		<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>
FLM-6792	Captain America of Iron Man	Miko Uwais	2022		<a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a>

*Gambar 7 sebelum logout*

Ini merupakan tampilan ketika user berhasil login, halaman yang dituju adalah halaman daftar film sama seperti tugas praktikum sebelumnya. Terdapat tombol **Logout** yang dapat digunakan untuk user keluar dari halaman ini.

## 2. Setelah Logout



*Gambar 7 setelah setelah logout*

Ini merupakan tampilan ketika user sudah berhasil **Logout**, user diharapkan masuk kembali melalui login untuk dapat masuk kembali.

## BAB III

### A. Kesimpulan

Kesimpulan dari praktikum ini adalah bahwa implementasi fitur **User Authentication** menggunakan Laravel telah berhasil dilakukan dengan mencakup proses **Login, Register, dan Logout**. Fitur-fitur ini memastikan bahwa pengguna dapat **mendaftar, mengautentikasi, dan keluar** dari aplikasi dengan aman. Validasi data yang ketat dan penggunaan hashing pada password meningkatkan keamanan aplikasi. Penerapan routing yang tepat memastikan pengguna diarahkan ke halaman yang sesuai, baik saat mengakses halaman login, register, maupun saat keluar dari sesi. Dengan adanya fitur-fitur ini, aplikasi menjadi lebih aman dan user-friendly, memungkinkan pengelolaan identitas dan akses yang efektif dalam sistem.