

LAPORAN PRAKTIKUM PEMROGRAMAN WEB
PRAKTIKUM 13 – LOGIN, REGISTRATION, AUTHENTICATION



Disusun oleh:

Nama : Afif Imam Rahadi

Nim : L0122006

Kelas : A

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA
UNIVERSITAS SEBELAS MARET

2024

1. Screenshot Source Code

Login Controller

```
class LoginController extends Controller
{
    public function index()
    {
        return view('login.index');
    }

    public function authenticate(Request $request)
    {
        $credentials = $request->validate([
            'username' => 'required',
            'password' => 'required'
        ]);

        if(Auth::attempt($credentials))
        {
            $request->session()->regenerate();
            return redirect()->intended('/students');
        }

        return back()->with('loginError', 'Login Failed');
    }

    public function logout(Request $request)
    {
        Auth::logout();

        $request->session()->invalidate();
        $request->session()->regenerateToken();

        return redirect('/');
    }
}
```

Pada bagian awal terdapat namespace dari controller ini, yaitu App\Http\Controllers. Kemudian, ada dua import statement yang mengimpor Request dari Illuminate\Http\Request dan Auth dari Illuminate\Support\Facades\Auth. Selanjutnya, Kelas LoginController mewarisi dari Controller dasar Laravel. Kelas ini memiliki tiga metode utama: index, authenticate, dan logout. Metode index(), Metode ini hanya mengembalikan view login.index.

Berikutnya terdapat Metode authenticate(Request \$request) yang menerima objek Request sebagai parameter, yang mewakili permintaan HTTP. Pertama, data dari permintaan divalidasi untuk memastikan bahwa username dan password ada dan diperlukan. Setelah validasi, kredensial ini digunakan untuk mencoba autentikasi pengguna menggunakan Auth::

attempt(\$credentials). Jika autentikasi berhasil, sesi pengguna diregenerasi untuk mencegah serangan sesi (session fixation attack) dan pengguna diarahkan ke halaman /students. Jika autentikasi gagal, pengguna akan dikembalikan ke halaman login dengan pesan error 'Login Failed'.

Terakhir, terdapat Metode logout(Request \$request) yang menerima objek Request sebagai parameter. Metode ini mengeluarkan pengguna yang sedang login menggunakan Auth::logout(), menginvalkan sesi untuk menghapus semua data sesi, dan meregenerasi token sesi untuk mencegah serangan CSRF (Cross-Site Request Forgery). Terakhir, pengguna diarahkan ke halaman utama.

Login Index

```
<body>
  <div class="container">
    @if (session()->has('success'))
      <div class="alert alert-success alert-dismissible fade show" role="alert">
        {{ session('success') }}
        <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
      </div>
    @endif

    @if (session()->has('loginError'))
      <div class="alert alert-danger alert-dismissible fade show" role="alert">
        {{ session('loginError') }}
        <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
      </div>
    @endif

    <h2 class="my-4" style="text-align:center;">Login</h2>
    <div class="row justify-content-center">
      <div class="col-lg-5">
        <main class="form-signin w-100 m-auto">
          <form action="/login" method="POST">
            @csrf
            <div class="form-floating">
              <input type="text" name="username" class="form-control @error('username') is-invalid @enderror" id="name" placeholder="username" autofocus required value="{{ old('username') }}">
              <label for="name">Username</label>
              @error('username')
                <div class="invalid-feedback">
                  {{ $message }}
                </div>
              @enderror
            </div>
            <div class="form-floating">
              <input type="password" name="password" class="form-control" id="password" placeholder="Password" required>
              <label for="password">Password</label>
            </div>
            <button class="btn btn-primary w-100 py-2" type="submit">Login</button>
          </form>
          <small class="d-block text-center mt-3">Didn't Have An Account? <a href="/register">Register</a></small>
        </main>
      </div>
    </div>
  </div>
```

Ini adalah code html yang digunakan untuk membuat halaman login. Pada Bagian <body> berisi elemen <div class="container"> untuk membungkus konten halaman. Selanjutnya terdapat pengecekan kondisi menggunakan @if untuk menampilkan pesan sukses atau kesalahan login berdasarkan kondisi session. Jika ada pesan sukses, sebuah alert Bootstrap dengan kelas alert-success ditampilkan; jika ada pesan kesalahan login, sebuah alert dengan kelas alert-danger ditampilkan.

Selanjutnya, sebuah judul dan elemen div yang digunakan untuk container dan column untuk form login. Form login diimplementasikan menggunakan atribut action="/login" untuk mengirimkan data ke URL /login dengan metode POST. Token CSRF disertakan menggunakan @csrf untuk keamanan. Form terdiri dari dua input, yaitu username dan password. Setiap input memiliki validasi Laravel untuk menampilkan pesan kesalahan jika ada, ditandai dengan kelas is-invalid dan elemen div dengan kelas invalid-feedback yang menampilkan pesan

kesalahan. Terakhir, sebuah tombol submit yang digunakan untuk mengirimkan formulir, dan sebuah link untuk pendaftaran pengguna baru ditampilkan di bawah tombol submit.

Register Controller

```
<?php

namespace App\Http\Controllers;

use App\Models\User;
use Illuminate\Http\Request;

class RegisterController extends Controller
{
    public function index()
    {
        return view('register.index');
    }

    public function store(Request $request)
    {
        $validated_data = $request->validate([
            'name' => 'required|max:255',
            'username' => 'required|min:3|max:255|unique:users',
            'email' => 'required|email:dns|unique:users',
            'password' => 'required'
        ]);

        $validated_data['password'] = bcrypt($validated_data['password']);

        User::create($validated_data);
        return redirect('/login')->with('success', 'Registration success, Please Login!');
    }
}
```

Bagian awal terdapat Namespace dan Imports yang digunakan untuk controller. Selanjutnya, terdapat Kelas RegisterController yang memiliki dua metode Utama yaitu, index dan store. Metode index() mengembalikan view register.index. Metode store(Request \$request) menerima objek Request sebagai parameter, yang mewakili permintaan HTTP. Data yang diterima dari permintaan divalidasi menggunakan metode validate(), yang memastikan bahwa nama diperlukan dan memiliki panjang maksimum 255 karakter, username diperlukan dengan panjang minimal 3 dan maksimal 255 karakter serta unik di tabel users, email diperlukan, harus sesuai format DNS, dan juga unik di tabel users, serta password diperlukan. Setelah validasi, password yang diterima di-hash menggunakan bcrypt untuk keamanan. Data yang telah divalidasi dan di-hash kemudian disimpan ke database dengan menggunakan metode create() dari model User. Setelah pengguna berhasil dibuat, pengguna diarahkan ke halaman login

(/login) dengan pesan sukses 'Registration success, Please Login!' menggunakan with() untuk menambahkan pesan ke session.

Register index

```
<body>
  <div class="container">
    <h2 class="my-4" style="text-align:center">Register</h2>
    <div class="row justify-content-center">
      <div class="col-lg-4">
        <form action="/register" method="POST">
          <csrf>
            <div class="form-floating">
              <input type="text" name="name" class="form-control rounded-top @error('name') is-invalid @enderror" id="name" placeholder="Name" required value="{{ old('name') }}">
              <label for="name">Name</label>
              @error('name')
                <div class="invalid-feedback">
                  {{ $message }}
                </div>
              @enderror
            </div>
            <div class="form-floating">
              <input type="text" name="username" class="form-control @error('username') is-invalid @enderror" id="username" placeholder="Username" required value="{{ old('username') }}">
              <label for="username">Username</label>
              @error('username')
                <div class="invalid-feedback">
                  {{ $message }}
                </div>
              @enderror
            </div>
            <div class="form-floating">
              <input type="email" name="email" class="form-control @error('email') is-invalid @enderror" id="email" placeholder="name@example.com" required value="{{ old('email') }}">
              <label for="email">Email address</label>
              @error('email')
                <div class="invalid-feedback">
                  {{ $message }}
                </div>
              @enderror
            </div>
            <div class="form-floating">
              <input type="password" name="password" class="form-control rounded-bottom @error('password') is-invalid @enderror" id="password" placeholder="Password" required>
              <label for="password">Password</label>
              @error('password')
                <div class="invalid-feedback">
                  {{ $message }}
                </div>
              @enderror
            </div>
            <div>
              <button class="btn btn-primary w-100 py-2 mt-3" type="submit">Register</button>
            </div>
            <small class="d-block text-center mt-3">Already Have An Account? <a href="/login">Login</a></small>
          </div>
        </form>
      </div>
    </div>
  </div>
```

Gambar di atas merupakan code dari halaman register. terdapat div untuk membungkus konten halaman yang di dalamnya terdapat judul. Lalu diikuti, form registrasi. Di dalam Form registrasi terdapat atribut action="/register" untuk mengirimkan data ke URL /register dengan metode POST. Token CSRF disertakan menggunakan @csrf untuk keamanan. Form ini terdiri dari empat input yaitu, nama, username, email, dan password. Tombol submit digunakan untuk mengirimkan formulir, dan sebuah link untuk login pengguna yang sudah memiliki akun ditampilkan di bawah tombol submit, dengan teks "Already Have An Account? Login".

Routes

```
// login
Route::get('/login', [LoginController::class, 'index']->name('login'));
Route::post('/login', [LoginController::class, 'authenticate']);

// logout
Route::post('/logout', [LoginController::class, 'logout']);

// register
Route::get('/register', [RegisterController::class, 'index']);
Route::post('/register', [RegisterController::class, 'store']);
```

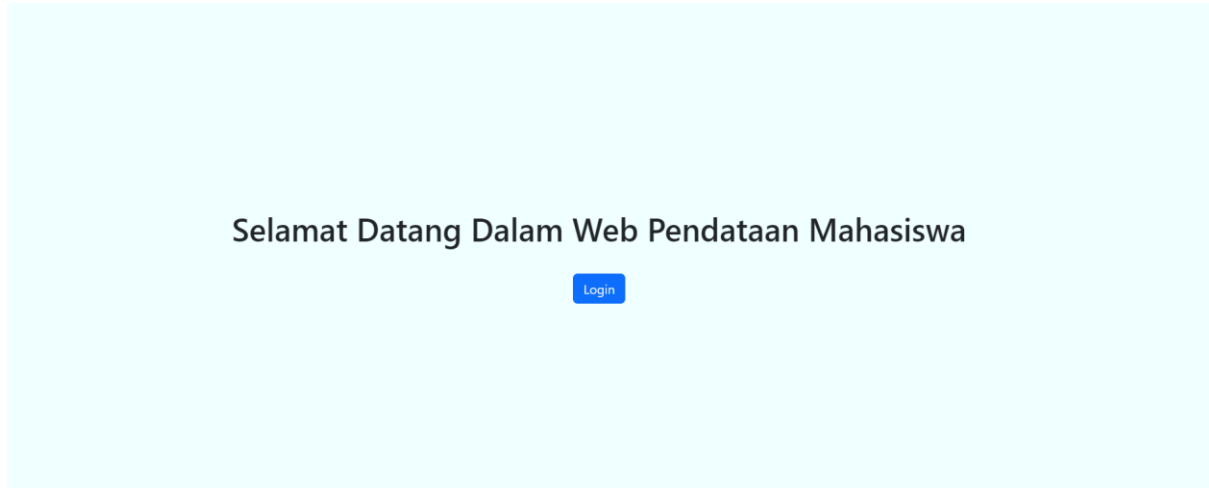
Pertama terdapat route login dengan method get yang diberi nama login. Baris ini mendefinisikan rute /login. Ketika pengguna mengakses URL ini, metode index dari LoginController akan dipanggil dan akan menampilkan halaman login. Kedua, terdapat route login dengan method post. Baris ini mendefinisikan rute POST untuk URL /login. Ketika pengguna mengirimkan form login, data akan dikirimkan ke rute ini, dan metode authenticate dari LoginController akan dipanggil untuk memproses login (validasi kredensial pengguna dan melakukan login).

Berikutnya terdapat route untuk logout. Route ini menggunakan method post. Baris ini mendefinisikan rute POST untuk URL /logout. Ketika pengguna mengklik tombol logout, permintaan POST akan dikirim ke rute ini, dan metode logout dari LoginController akan dipanggil untuk melakukan logout (menghapus sesi pengguna dan mengarahkan pengguna ke halaman yang sesuai).

Terakhir terdapat route untuk register. Pertama route menggunakan method get. Baris ini mendefinisikan rute GET untuk URL /register. Ketika pengguna mengakses URL ini, metode index dari RegisterController akan dipanggil dan akan menampilkan halaman registrasi. Selanjutnya, route dengan method post. Baris ini mendefinisikan rute POST untuk URL /register. Ketika pengguna mengirimkan form registrasi, data akan dikirimkan ke rute ini, dan metode store dari RegisterController akan dipanggil untuk memproses pendaftaran (validasi data pengguna dan menyimpan pengguna baru ke database).

2. Screenshot Ouput

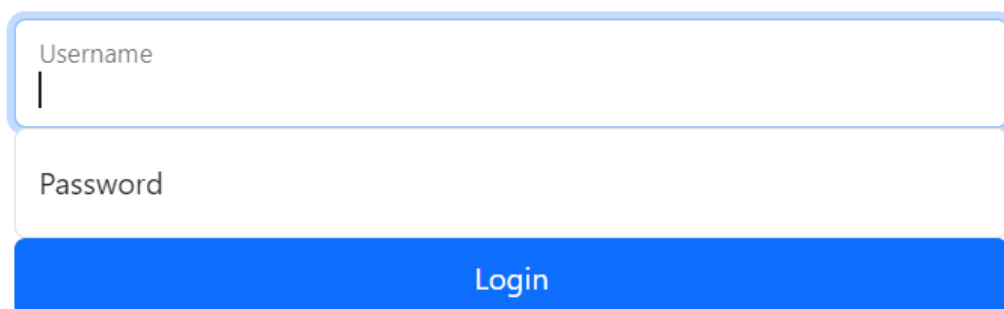
Home



Ini adalah halaman home. Halaman home sangat sederhana yang berisi tulisan selamat datang. Di bawahnya, terdapat button login. Jika button ini di klik, maka program akan mengarahkannya ke halaman login. Pada halaman utama, hanya ada button login yang berarti semua user untuk melihat data dan melakukan CRUD harus melakukan login/register.

Login

Login

The screenshot shows a login form with two input fields. The first field is labeled "Username" and has a vertical cursor inside. The second field is labeled "Password". Below these fields is a large blue button with the word "Login" in white text.

Didn't Have An Account? [Register](#)

Ini adalah halaman login, halaman login terdiri dari username dan password. Pada bagian username terdapat autofocus yang membuat user langsung dapat mengisi field username secara langsung.

Login Failed ×

Login

Username

Password

Login

Didn't Have An Account? [Register](#)

Terdapat juga pesan error jika username atau password salah atau belum terdaftar pada database.

Register

Name

Username

Email address

Password

Register

Already Have An Account? [Login](#)

Ini adalah halaman register yang terdiri dari 4 inputan berupa nama, username, email, dan password. Pada bagian ini, user akan melakukan register agar masuk ke dalam database.

Register

Name
ajdbawjbdawbdjabd

Username
adbajdbadbajb

Email address
tes@ddd

The email field must be a valid email address.

Password

Register

Already Have An Account? [Login](#)

Terdapat juga pesan error untuk setiap field jika tidak sesuai dengan yang diinginkan. Dan semua field ini required yang harus diisi oleh user.

Index

Data Mahasiswa

No	Nama	NIM	Prodi	Kelas	Angkatan	Aksi
Tambah Data	Logout					

Ini adalah halaman utama dari data mahasiswa. Sama seperti sebelumnya yang menampilkan data nama, nim, prodi, kelas Angkatan, dan terdapat aksi untuk melakukan edit dan delete. Pada bagian bawah terdapat button untuk melakukan penambahan data dan juga melakukan logout. Jika button logut dipencet, maka akan masuk ke halaman awal yang berupa judul.