

PEMROGRAMAN FRAMEWORK

LAPORAN PRAKTIKUM MODUL 8



Disusun Oleh:
Ferdi Zahi Robbani
1204220104

PROGRAM STUDI S1 SISTEM INFORMASI
FAKULTAS REKAYASA INDUSTRI
TELKOM UNIVERSITY SURABAYA
2024

1. Authentication via Laravel UI

- Membuat project Laravel baru dengan melihat bagian controllers, views dan routes dan masuk ke project Laravel yang telah dibuat.
- Install package Laravel UI via composer.

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.3527]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ferdi\Documents\Modul_8\Modul8>composer require laravel/ui
```

- Generate Scaffolding Bootstrap dengan fitur Authentification pada project Laravel kemudian perhatikan bagian controllers, views dan routes.

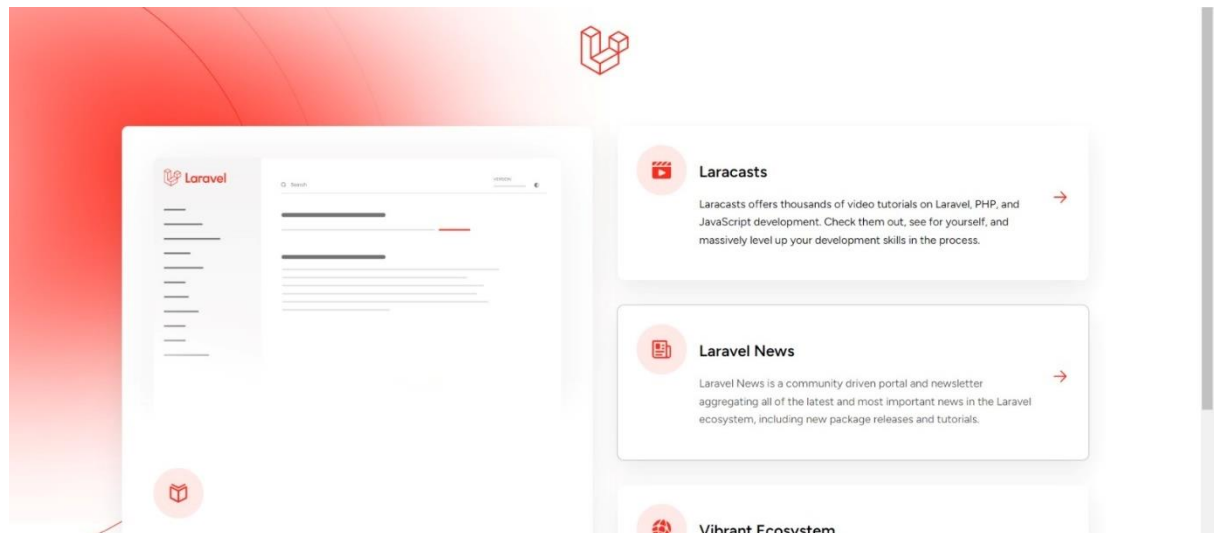
```
C:\Users\Ferdi\Documents\Modul_8\Modul8>php artisan ui bootstrap --auth
```

- Analisis kondisi project Laravel antara sebelum dan sesudah dijalankan dan jelaskan.

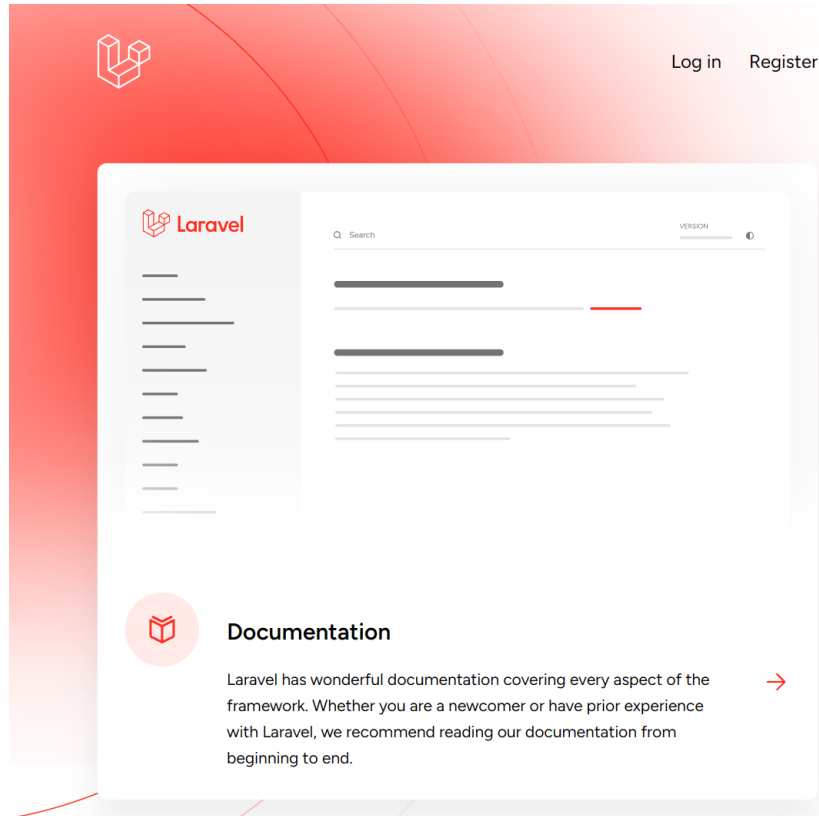
Penjelasan:

Setelah menjalankan fitur Auth, maka pada project mendapatkan beberapa tambahan fitur, seperti login, register dan log out untuk mengelola authentication dan meningkatkan keamanan website.

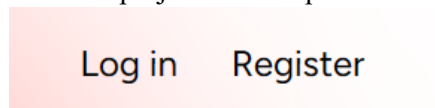
Sebelum:



Sesudah:

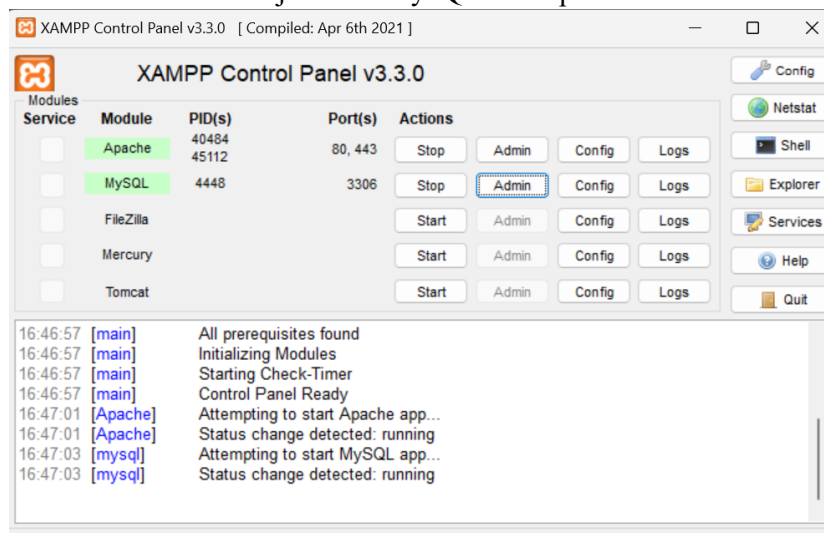


- Jalankan project Laravel pada browser akan muncul fitur Login dan Register.

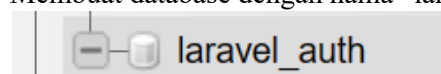


2. Setup Database

- Jalankan XAMPP dan jalankan MySQL dan Apache.



- Membuat database dengan nama "laravel_auth" di PhpMyAdmin.



- Pada file .env sesuaikan value pada variable DB_DATABASE, DB_USERNAME, DB_PASSWORD.

```
22 DB_CONNECTION=mysql
23 DB_HOST=127.0.0.1
24 DB_PORT=3306
25 DB_DATABASE=laravel_auth
26 DB_USERNAME=root
27 DB_PASSWORD=
```

- Melakukan Laravel migration.

```
PS C:\Users\Ferdi\Documents\Modul_8\Modul8>
php artisan migrate
INFO Nothing to migrate.
```

3. Menjalankan Fitur Register dan Login

- Pada browser dan coba fitur Register untuk membuat akun baru.

Laravel

[Login](#) [Register](#)

The screenshot shows a web browser displaying the 'Register' form of a Laravel application. The form is titled 'Register' and contains four input fields: 'Name' with the value 'Ferdi', 'Email Address' with the value 'ya@gmail.com', 'Password' with masked characters '.....', and 'Confirm Password' with masked characters '.....'. Below the fields is a blue 'Register' button. The form is set against a light blue background.

- Lalu coba fitur Login yang baru saja dibuat.

Laravel

[Login](#) [Register](#)

The screenshot shows a web browser displaying the 'Login' form of a Laravel application. The form is titled 'Login' and contains two input fields: 'Email Address' with the value 'ya@gmail.com' and 'Password' with masked characters '.....'. Below the fields is a checkbox labeled 'Remember Me' which is unchecked. At the bottom of the form are a blue 'Login' button and a blue link labeled 'Forgot Your Password?'. The form is set against a light blue background.



- Analisis alur program ketika menjalankan fitur Login dan Register.

Penjelasan:

Pada alur program ketika user mengakses halaman register dan login maka Laravel akan mengarahkan permintaan ke controller yang sesuai dan akan memproses data yang sesuai dengan views yang telah ditentukan sebelumnya.

4. Proses Authentication Secara Manual

- Pada file LoginController tambah function dengan nama authenticate().

```
<?php

namespace App\Http\Controllers\Auth;
use App\Http\Controllers\Controller;
use App\Providers\RouteServiceProvider;
use Illuminate\Foundation\Auth\AuthenticatesUsers;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class LoginController extends Controller
{
    /**
     |-----
     | Login Controller
     |-----
     |
     | This controller handles authenticating users for the application and
     | redirecting them to your home screen. The controller uses a trait
     | to conveniently provide its functionality to your applications.
     |
     */

    use AuthenticatesUsers;

    /**
     * Where to redirect users after login.
     *
     * @var string
     */
    protected $redirectTo = RouteServiceProvider::HOME;

    /**
     * Create a new controller instance.
     */
}
```

```

*
* @return void
*/
public function construct()
{
    $this->middleware('guest')->except('logout');
}

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

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

    return back()->withErrors([
        'email' => 'The provided credentials do not match our records.',
    ]->onlyInput('email'));
}
}

```

Penjelasan:

Pada method authentication(Request \$request)

- Pada method authentication() digunakan untuk melakukan validasi terhadap kredensial yang diterima di form login.
 - Pada method validate() digunakan untuk memastikan input yang diisi oleh user sesuai dengan aturan validasi yang telah ditentukan.
 - Pada Auth::attempt(\$credentials) digunakan untuk mencoba melakukan autentikasi menggunakan kredensial yang telah diberikan.
 - Jika pada auth::attempt() berhasil, maka user akan diarahkan ke halaman yang dituju dan jika tidak ada yang dituju akan diarahkan ke home.
 - Jika gagal maka user akan diarahkan ke halaman login dengan pesan yang telah ditentukan.
- Pada file web.php buat route baru dan diarahkan pada authenticate() di LoginController.

```
Route::post('/login', [LoginController::class, 'authenticate']);
```

5. Latihan

Berikut ini terdapat perubahan kode program yaitu:

- LoginController

```
<?php
namespace App\Http\Controllers\Auth;
use App\Http\Controllers\Controller;

use App\Providers\RouteServiceProvider;
use Illuminate\Foundation\Auth\AuthenticatesUsers;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class LoginController extends Controller
{
    /**
     |-----
     | Login Controller
     |-----
     |
     | This controller handles authenticating users for the application and
     | redirecting them to your home screen. The controller uses a trait
     | to conveniently provide its functionality to your applications.
     |
     */

    use AuthenticatesUsers;

    /**
     * Where to redirect users after login.
     *
     * @var string
     */
    protected $redirectTo = '/home';

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function construct()
    {
        $this->middleware('guest')->except('logout');
    }
}
```

```

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

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

    return back()->withErrors([
        'email' => 'The provided credentials do not match our records.',
    ]->onlyInput('email'));
}
}

```

- Login.blade.php

```

<?php
namespace App\Http\Controllers\Auth;
use App\Http\Controllers\Controller;

use App\Providers\RouteServiceProvider;
use Illuminate\Foundation\Auth\AuthenticatesUsers;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class LoginController extends Controller
{
    /**
     |-----
     | Login Controller
     |-----
     |
     | This controller handles authenticating users for the application and
     | redirecting them to your home screen. The controller uses a trait
     | to conveniently provide its functionality to your applications.
     |
     */

    use AuthenticatesUsers;

    /**
     * Where to redirect users after login.
     *
     * @var string
     */
}

```



```

    */
    protected $redirectTo = '/home';

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function construct()
    {
        $this->middleware('guest')->except('logout');
    }

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

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

        return back()->withErrors([
            'email' => 'The provided credentials do not match our records.',
        ]->onlyInput('email'));
    }
}

```

- app.blade.php

```

<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- CSRF Token -->
    <meta name="csrf-token" content="{{ csrf_token() }}">

    <title>{{ config('app.name', 'Laravel') }}</title>

    <!-- Fonts -->
    <link rel="dns-prefetch" href="//fonts.bunny.net">
    <link href="https://fonts.bunny.net/css?family=Nunito" rel="stylesheet">

```

```

    <!-- Scripts -->
    @vite(['resources/sass/app.scss', 'resources/js/app.js'])
</head>
<body>
    @include('layouts.nav')
    @yield('content')
    @vite('resources/js/app.js')
</body>
</html>

```

- nav.blade.php

```

@php
    $currentRouteName = Route::currentRouteName();
@endphp

{{-- Navbar Component --}}
<nav class="navbar navbar-expand-md navbar-dark bg-primary">
    <div class="container">
        <a href="{{ route('home') }}" class="navbar-brand mb-0 h1"><i class="bi-hexagon-fill me-2"></i> Data Master</a>
        <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <hr class="d-lg-none text-white-50">
            <ul class="navbar-nav flex-row flex-wrap">
                <li class="nav-item col-2 col-md-auto"><a href="{{ route('home') }}"
                    class="nav-link {{ $currentRouteName == 'home' ? 'active' : '' }}">Home</a></li>
                <li class="nav-item col-2 col-md-auto"><a href="{{ route('employees.index') }}"
                    class="nav-link {{ $currentRouteName == 'employees.index' ? 'active' : '' }}">Employee List</a>
                </li>
            </ul>
            <hr class="d-lg-none text-white-50">
            <div class="dropdown ms-md-auto">
                <button class="btn btn-outline-light dropdown-toggle" type="button"
                    id="dropdownMenuButton"
                    data-bs-toggle="dropdown" aria-expanded="false">
                    <i class="bi-person-circle me-1"></i> Administrator
                </button>
                <ul class="dropdown-menu" aria-labelledby="dropdownMenuButton">
                    <li><a class="dropdown-item" href="{{ route('profile') }}"><i class="bi-person me-1"></i> My

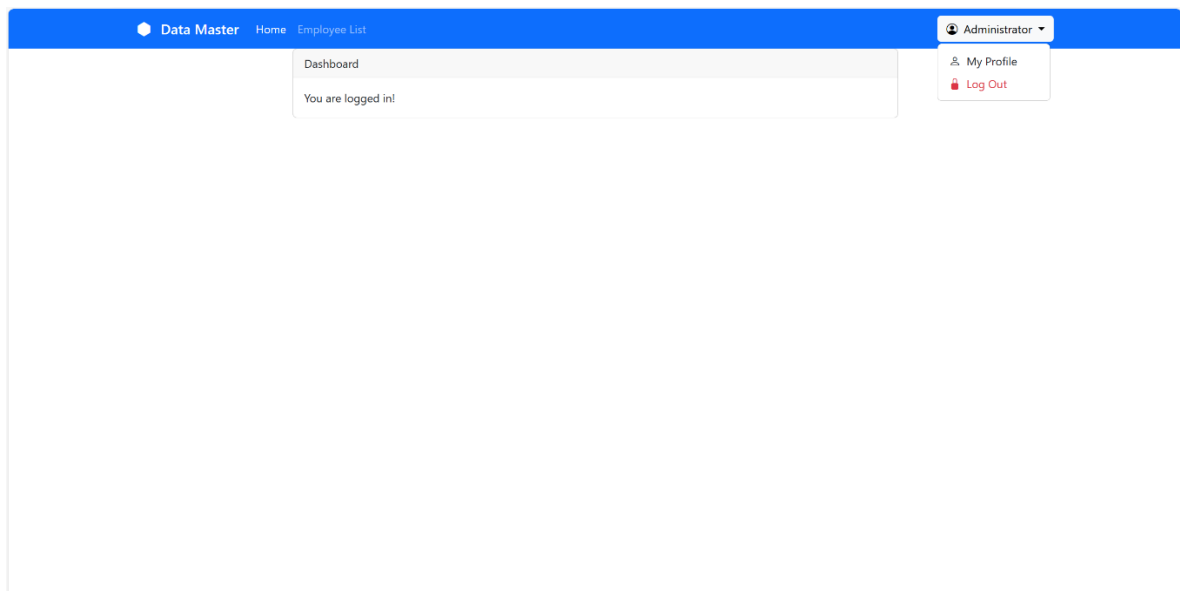
```

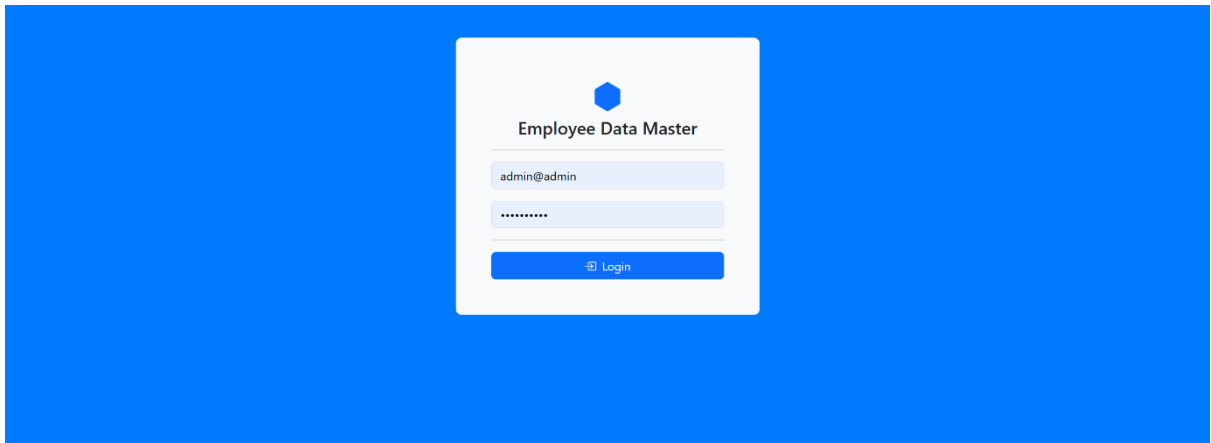
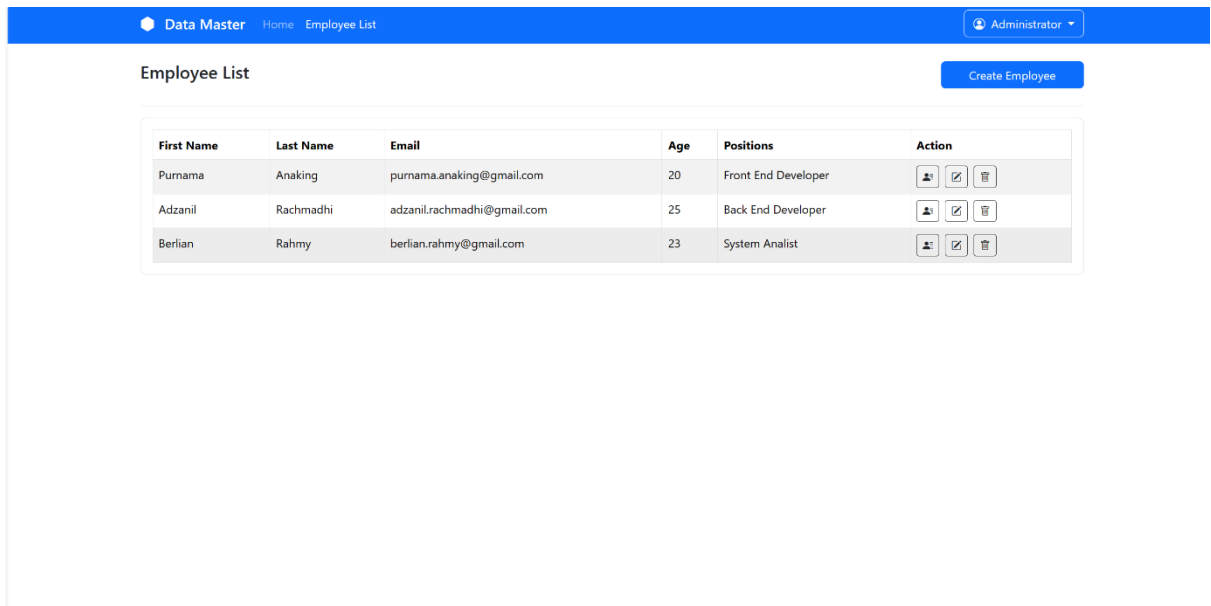
```

        Profile</a></li>
      <li>
        <form action="{{ route('logout') }}" method="POST" class="d-
inline">
          @csrf
          <button type="submit" class="dropdown-item text-danger"><i
class="bi bi-lock-fill text-danger me-1"></i> Log
Out</button>
        </form>
      </li>
    </ul>
  </div>
</div>
</div>
</nav>

```

Output:





Link Github:

Repository 1: https://github.com/FerdiZ12/Modul_8_Part-1.git

Repository 2: https://github.com/FerdiZ12/Modul_8_Part-2.git