

PRAKTIKUM MODUL 08
PEMROGRAMAN FRAMEWORK
LARAVEL AUTHENTICATION



DISUSUN OLEH:

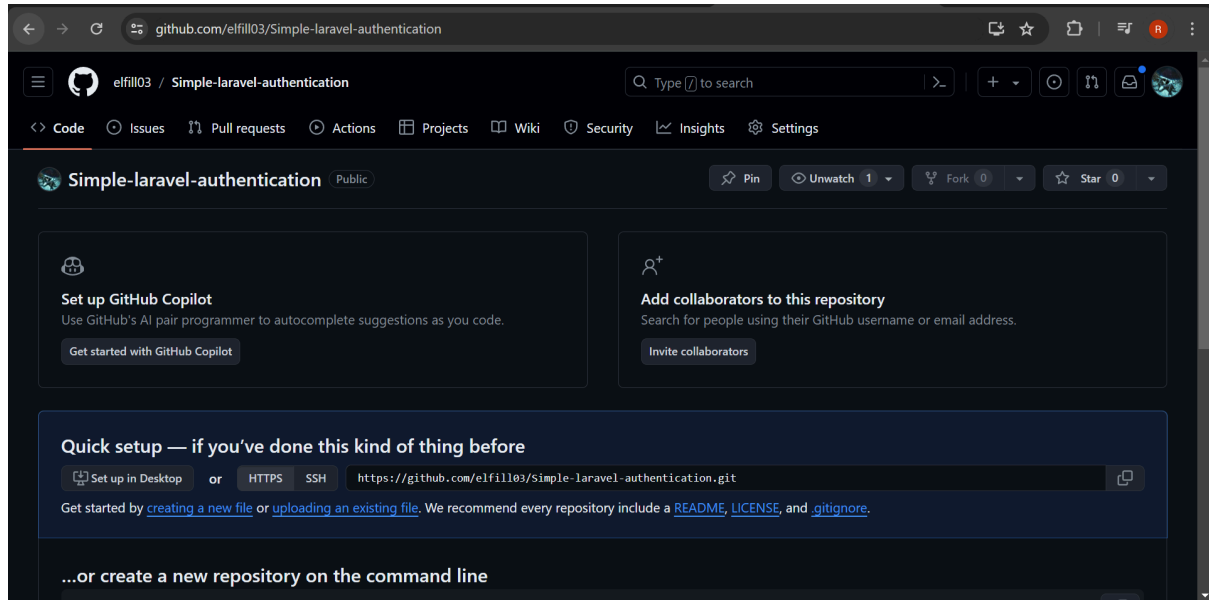
1204200081 RAFAEL FILLAH FIRMANSYAH

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS REKAYASA INDUSTRI
TELKOM UNIVERSITY SURABAYA
2024

Membuat Repository Baru pada Github

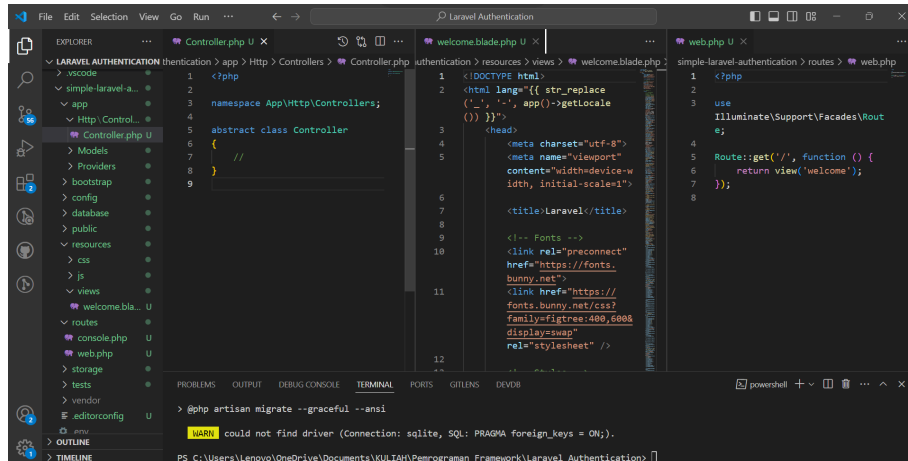
Link Github: <https://github.com/elfill03/Simple-laravel-authentication>

1. Pembuatan repository dengan nama Simple-laravel-authentication

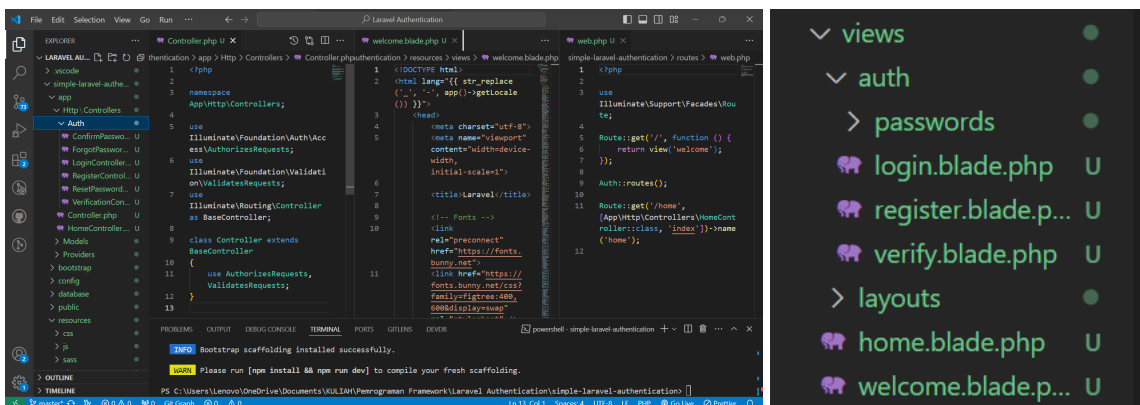


Authentication Via Laravel UI

1. Membuat project laravel baru kemudian meng generate laravel ui
2. Menjalankan fitur ui bootstrap autentikasi dan membandingkan perbedaan antara sebelum dan sesudah menjalankannya pada bagian view, controller, dan routes-nya.



(Sebelum)

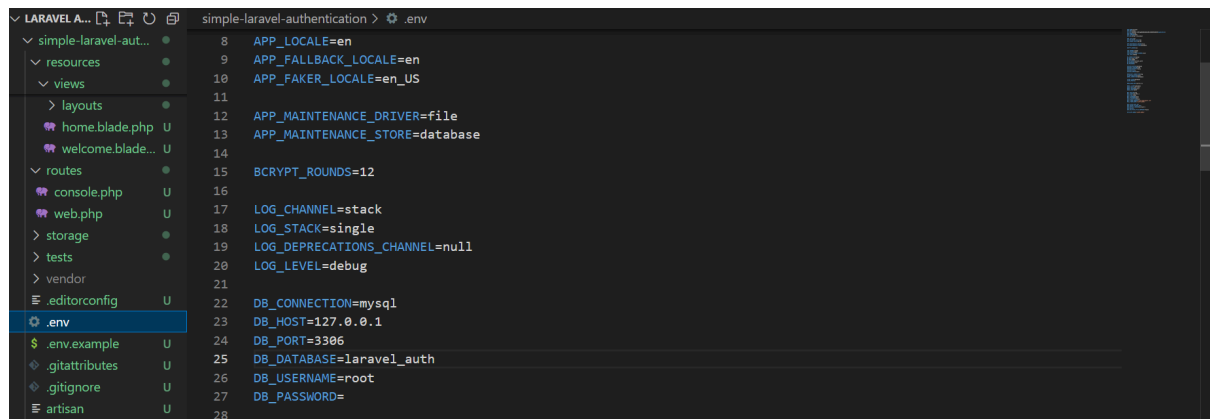


(Sesudah)

Untuk bagian view ketika telah meng generate ui bootstrap auth maka akan muncul folder tambahan dengan nama auth yang berisi beberapa file untuk fitur register, login, dan verifikasi. Sedangkan pada bagian controller terdapat folder tambahan dengan nama Auth yang berisi beberapa controller autentikasi untuk fitur register dan login, selain itu adanya tambahan middleware pada HomeController agar selain user yang telah melakukan login tidak dapat mengakses halaman tersebut. Terakhir pada bagian routes-nya yaitu adanya protected route berdasarkan auth yang dimana pada halaman home atau dashboard tidak dapat diakses apabila user belum login dan akan dikembalikan ke halaman awal.

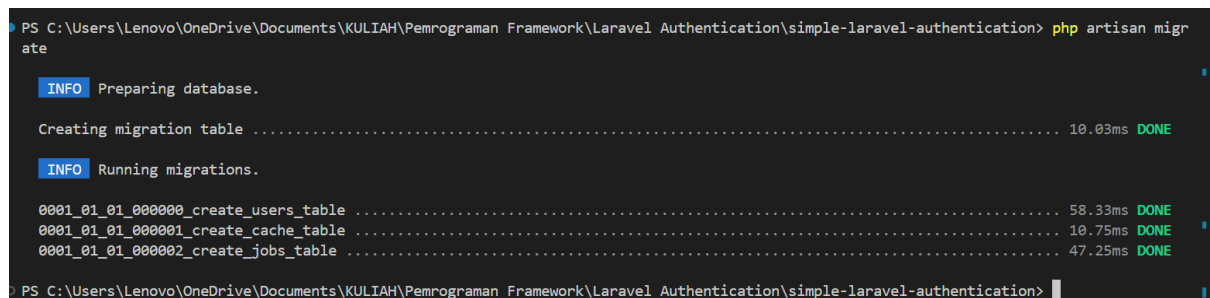
Setup Database

1. Untuk database telah di set up menggunakan mysql dan dengan nama database yaitu laravel_auth



```
simple-laravel-authentication > .env
8 APP_LOCALE=en
9 APP_FALLBACK_LOCALE=en
10 APP_FAKER_LOCALE=en_US
11
12 APP_MAINTENANCE_DRIVER=file
13 APP_MAINTENANCE_STORE=database
14
15 BCRYPT_ROUNDS=12
16
17 LOG_CHANNEL=stack
18 LOG_STACK=single
19 LOG_DEPRECATIONS_CHANNEL=null
20 LOG_LEVEL=debug
21
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=
28
```

2. Kemudian menjalankan migrate untuk membuat schema database berdasarkan default dari ui bootstrap auth



```
PS C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Laravel Authentication\simple-laravel-authentication> php artisan migrate

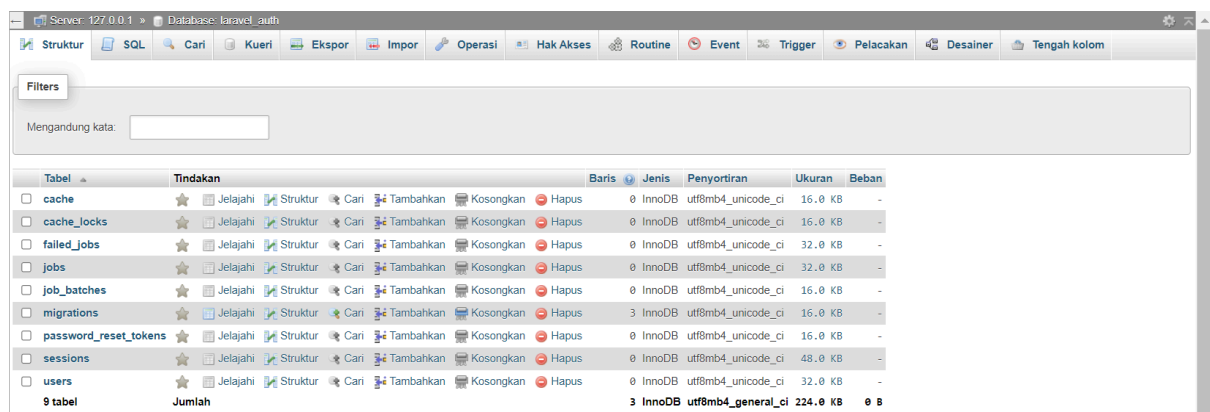
INFO Preparing database.

Creating migration table ..... 10.03ms DONE

INFO Running migrations.

0001_01_01_000000_create_users_table ..... 58.33ms DONE
0001_01_01_000001_create_cache_table ..... 10.75ms DONE
0001_01_01_000002_create_jobs_table ..... 47.25ms DONE

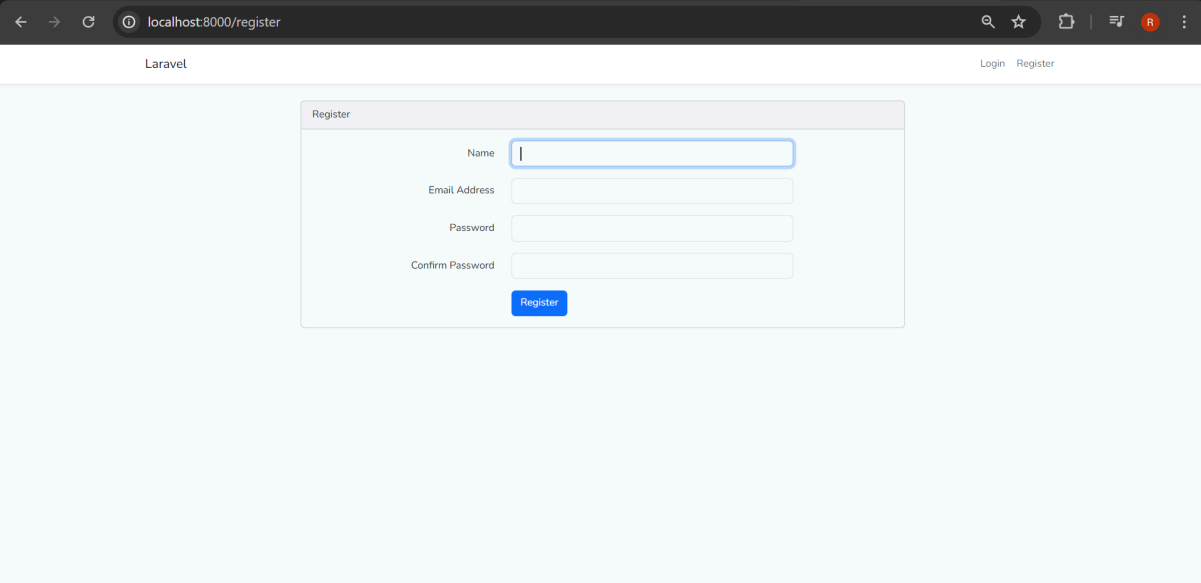
PS C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Laravel Authentication\simple-laravel-authentication>
```



Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
cache	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
cache_locks	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
failed_jobs	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
jobs	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
job_batches	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
migrations	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	3	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
password_reset_tokens	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	16.0 KB	-
sessions	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	48.0 KB	-
users	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_unicode_ci	32.0 KB	-
9 tabel	Jumlah	3	InnoDB	utf8mb4_general_ci	224.0 KB	0 8

Menjalankan Fitur Register dan Login

1. Untuk alur fitur register, saat pengguna menekan fitur untuk register akun maka route akan mengakses RegisterController yang berisi beberapa function public dan protected function, penggunaan middleware pada RegisterController mengarah pada akun yg awalnya masih guest sehingga perlu melakukan pembuatan akun dan melakukan login agar dideteksi sebagai akun yang terdaftar. Pada RegisterController mengarahkan ke file views dengan nama register, dari file views ini berisi form untuk pembuatan akun. Form yang ada pada halaman register ada beberapa yang mengarah ke Controller tertentu, diantaranya yaitu form untuk email dan form confirm password. Untuk form email terdapat controller untuk melakukan verifikasi dan untuk form confirm password adanya controller untuk konfirmasi password. Apabila pengguna telah mengisi semua form dengan data yang valid maka pengguna akan diarahkan menuju halaman home dengan status login dan bukan guest. Untuk data user telah masuk kedalam database.



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/register'. The page title is 'Laravel'. In the top right corner, there are links for 'Login' and 'Register'. The main content area features a 'Register' form with the following fields: 'Name' (with a cursor), 'Email Address', 'Password', and 'Confirm Password'. A blue 'Register' button is located at the bottom of the form.

2. Untuk alur fitur login, saat pengguna menekan fitur untuk login maka route akan mengakses LoginController yang berisi beberapa function public dan protected function, serta adanya autentikasi pada saat user telah mengisi form email dan password. Pada LoginController mengarahkan ke file views dengan nama login, dari file views ini berisi form untuk login akun. Adapun beberapa fitur tambahan yang ada pada halaman login yang mengarah ke Controller tertentu, diantaranya yaitu fitur untuk forgot password dan remember me. Selanjutnya saat user telah melakukan login maka akan membuka halaman home.

Login

Email Address

D

Password

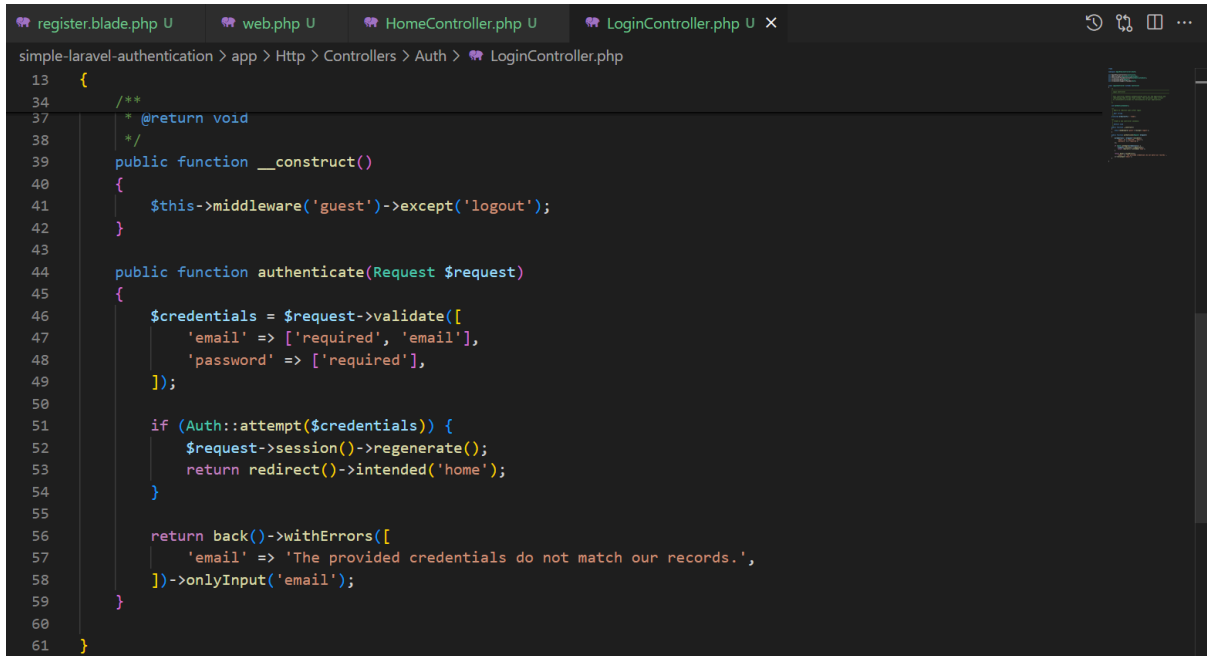
☐ Remember Me

Login

[Forgot Your Password?](#)

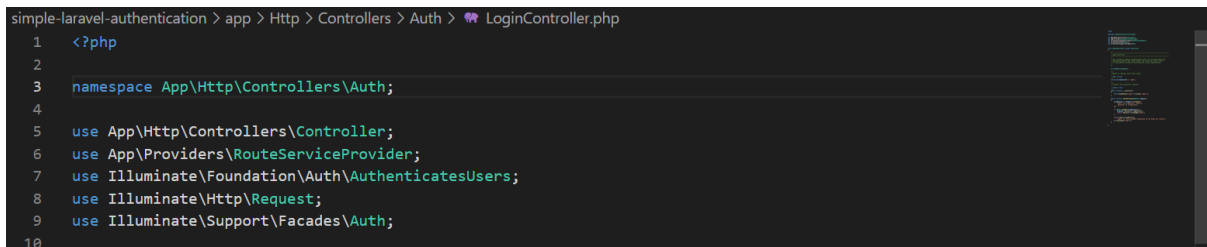
Proses Authentication Secara Manual

1. Untuk melakukan autentikasi secara manual dapat dilakukan dengan cara menambahkan function baru pada LoginController contohnya dengan nama authenticate yang berisi validasi form dan auth untuk autentikasi



```
13 {
14     /**
15      * @return void
16      */
17     public function __construct()
18     {
19         $this->middleware('guest')->except('logout');
20     }
21
22     public function authenticate(Request $request)
23     {
24         $credentials = $request->validate([
25             'email' => ['required', 'email'],
26             'password' => ['required'],
27         ]);
28
29         if (Auth::attempt($credentials)) {
30             $request->session()->regenerate();
31             return redirect()->intended('home');
32         }
33
34         return back()->withErrors([
35             'email' => 'The provided credentials do not match our records.',
36         ])->onlyInput('email');
37     }
38 }
```

2. Kemudian tambahkan juga import untuk auth, controller, dan validasinya di bagian atas



```
1 <?php
2
3 namespace App\Http\Controllers\Auth;
4
5 use App\Http\Controllers\Controller;
6 use App\Providers\RouteServiceProvider;
7 use Illuminate\Foundation\Auth\AuthenticatesUsers;
8 use Illuminate\Http\Request;
9 use Illuminate\Support\Facades\Auth;
10
```

3. Terakhir pada bagian route (file web.php) tambahkan juga route yang mengarah ke halaman login dan penggunaan function authenticate



```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5 Route::get('/', function () {
6     return view('welcome');
7 });
8
9 Auth::routes();
10
11 Route::get('/home', [App\Http\Controllers\HomeController::class, 'index']->name('home'));
12
13 Route::post('/login', [App\Http\Controllers\Auth\LoginController::class, 'authenticate']);
14
```

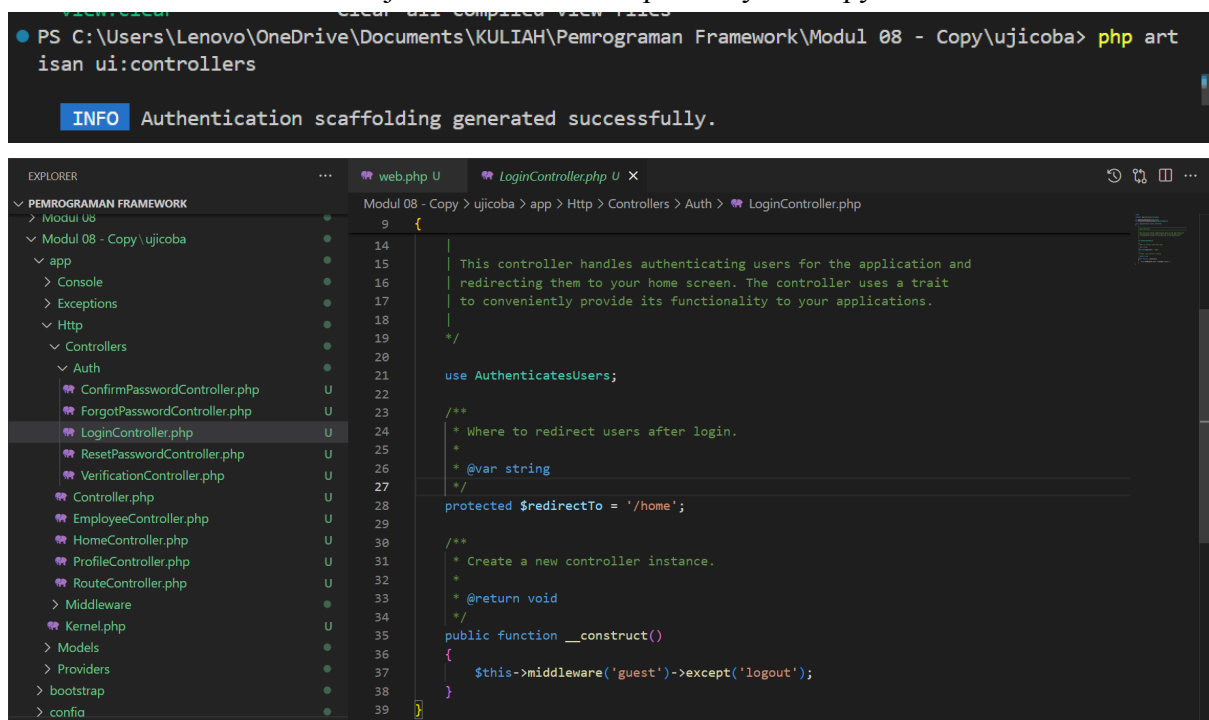
LATIHAN

(Menambahkan Fitur Login untuk Employee List)

Link Github: <https://github.com/elFill03/Pemrograman-Framework>

1. Ketika hendak membuat fitur login perlu menambahkan Scaffold the authentication controllers dengan cara “php artisan ui:controllers”, kenapa tidak menggunakan ui:auth karena penggunaan ui:auth hanya mengenerate basic login sedangkan yang dibutuhkan untuk Employee List yaitu controller serta set up autentikasi yang dapat dikustomisasi dan hanya membutuhkan controller dan middlewarynya.

Disclaimer: nama folder yang saya gunakan terdapat “- copy” untuk berjaga-jaga jika terjadi crash maka folder utama Modul 8 tidak terpengaruh (backup), Nantinya ketika latihan telah selesai akan diubah menjadi “Modul 08” tanpa adanya “- copy”.



The image shows a terminal window and a VS Code editor. The terminal window displays the command `php artisan ui:controllers` and the output `INFO Authentication scaffolding generated successfully.`. The VS Code editor shows the file explorer on the left with the project structure, and the main editor window displaying the code for `LoginController.php`. The code includes a comment describing the controller's purpose, a trait `AuthenticatesUsers`, a protected property `$redirectTo` set to `'/home'`, and a `__construct()` method that sets up a middleware exception handler.

```
PS C:\Users\Lenovo\OneDrive\Documents\KULIAH\Pemrograman Framework\Modul 08 - Copy\ujicoba> php artisan ui:controllers

INFO Authentication scaffolding generated successfully.
```

```
Modul 08 - Copy > app > Http > Controllers > Auth > LoginController.php
9  {
14
15      | This controller handles authenticating users for the application and
16      | redirecting them to your home screen. The controller uses a trait
17      | to conveniently provide its functionality to your applications.
18      |
19      | */
20
21      use AuthenticatesUsers;
22
23      /**
24       * Where to redirect users after login.
25       *
26       * @var string
27       */
28      protected $redirectTo = '/home';
29
30      /**
31       * Create a new controller instance.
32       *
33       * @return void
34       */
35      public function __construct()
36      {
37          $this->middleware('guest')->except('logout');
38      }
39  }
```

2. Selanjutnya membuat route yang telah di protect menggunakan middleware, hal ini diperlukan agar ketika pengguna hendak menuju halaman home atau employee list atau profil tidak langsung menuju halaman tersebut tapi di redirect ke halaman login karena memerlukan status user sebagai akun yang telah melakukan login


```
web.php U x LoginController.php U
Modul 08 - Copy > ujicoba > routes > web.php
82 Route::resource('employees', EmployeeController::class);
83
84 // Modul 05
85 Route::get('/employees/{id}/edit', [EmployeeController::class, 'edit'])->name('employees.
edit');
86
87 // Modul 08
88 Auth::routes();
89
90 Route::get('/', function() {
91     return redirect('/login');
92 });
93
94 Route::get('/home', [HomeController::class, 'index'])->name('home')->middleware('auth');
95
96 Route::get('/profile', ProfileController::class)->name('profile')->middleware('auth');
97
98 Route::resource('/employees', EmployeeController::class)->middleware('auth');
99
```

3. Kemudian membuat sebuah seeder dengan nama user yang nantinya akan dijadikan sebagai akun untuk login, karena tidak adanya fitur register maka akun di generate dari seeder. Untuk jenis password perlu ditambahkan bcrypt() agar sandi yang dimasukkan akan di enkripsi dan berubah menjadi kumpulan huruf yang tidak bisa dibaca oleh manusia

```
web.php U UserSeeder.php U x
Modul 08 - Copy > ujicoba > database > seeders > UserSeeder.php
1 <?php
2
3 namespace Database\Seeders;
4
5 use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6 use Illuminate\Database\Seeder;
7 use Illuminate\Support\Facades\DB;
8
9 class UserSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      */
14     public function run(): void
15     {
16         DB::table('users')->insert([
17             [
18                 'name' => 'admin',
19                 'email' => 'admin@admin',
20                 'password' => bcrypt('adminadmin')
21             ],
22         ]);
23     }
24 }
25 }
```

4. Selanjutnya membuat sebuah halaman login pada folder views, jika melakukan generate menggunakan ui:auth maka akan otomatis muncul folder auth yang berisi halaman register, login, forgot password, dst. Namun karena yang di generate adalah ui:controllers maka kita perlu membuat sendiri folder auth serta halaman loginnya.

```

web.php U login.blade.php X
Modul 08 - Copy > ujicoba > resources > views > auth > login.blade.php > html > body.bg-primary > div.container-sm.mt-5.p-5 > form > div.row.justify-c
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <meta http-equiv="X-UA-Compatible" content="ie=edge">
8     <title>Login</title>
9     @vite('resources/sass/app.scss')
10 </head>
11
12 <body class="bg-primary">
13     <div class="container-sm mt-5 p-5">
14         <form method="POST" action="{{ route('login') }}">
15             @csrf
16             <div class="row justify-content-center">
17                 <div class="p-5 bg-light rounded-3 col-xl-4 shadow-lg">
18                     <div class="mb-5 text-center">
19                         <i class="bi-hexagon-fill fs-1 text-primary"></i>
20                         <h4>Employee Data Master</h4>
21                     </div>
22                     <hr>
23                     <div class="col my-4">
24                         <input id="email" type="email" class="form-control @error('email') is-invalid
25                             name="email" value="{{ old('email') }}" required autocomplete="email" autofocus
26                             placeholder="Enter Your Email">
27                         @error('email')
28                             <span class="invalid-feedback" role="alert">
29                                 <small>{{ $message }}</small>
30                             </span>
31                         @enderror
32                     </div>
33
34                     <div class="col my-4">
35                         <input id="password" type="password"
36                             class="form-control @error('password') is-invalid @enderror" name="password"
37                             required
38                             autocomplete="current-password" placeholder="Enter Your Password">
39                         @error('password')
40                             <span class="invalid-feedback" role="alert">
41                                 <small>{{ $message }}</small>
42                             </span>
43                         @enderror
44                     </div>
45
46                     <hr>
47
48                     <div class="row">
49                         <div class="col d-grid">
50                             <button type="submit" class="btn btn-primary btn-lg mt-3"><i
51                                 class=" " "></i> Log In</button>
52                         </div>
53                     </div>
54                 </div>
55             </form>
56         </div>
57         @vite('resources/js/app.js')
58     </body>
59
60 </html>
61

```

5. Berikut tampilan dari fitur login yang telah berhasil dibuat. Untuk simulasi login pada halaman employee list akan saya buat dalam bentuk video dan tambahkan pada github

