

Laporan Modul 8: Authentication & Authorization

Mata Kuliah: Workshop Web Lanjut

Nama: Adha Gusti Harmadhan

NIM: 2024573010009

Kelas: 2B TI

Abstrak

Laporan ini menjelaskan hasil praktikum pada Modul 8: *Authentication & Authorization* dalam mata kuliah Workshop Web Lanjut. Fokus praktikum adalah memahami mekanisme autentikasi (verifikasi identitas pengguna) dan otorisasi (kontrol akses berdasarkan peran/izin) di Laravel 12. Praktikum terbagi menjadi dua bagian: (1) implementasi autentikasi menggunakan Laravel Breeze (registrasi, login, logout, reset password, profil) dan (2) implementasi kontrol akses berbasis peran (RBAC) sederhana dengan middleware kustom yang membatasi akses ke view berdasarkan peran (admin, manager, user). Dengan praktikum ini mahasiswa diharapkan mampu menerapkan sistem keamanan dasar pada aplikasi Laravel, termasuk enkripsi password, proteksi rute menggunakan middleware, serta pembuatan middleware kustom dan seeder peran.

1. Dasar Teori

1.1 Autentikasi vs Otorisasi

- **Autentikasi (Authentication):** proses memverifikasi identitas pengguna — contoh: login menggunakan email dan password.
- **Otorisasi (Authorization):** proses menentukan tindakan apa yang boleh dilakukan oleh pengguna yang sudah terautentikasi — contoh: hanya admin yang dapat mengakses panel admin.

Keduanya saling melengkapi: autentikasi memastikan siapa pengguna, sedangkan otorisasi menentukan hak aksesnya.

1.2 Laravel Breeze (Scaffolding Auth)

Laravel Breeze adalah starter kit minimal untuk fitur autentikasi di Laravel. Breeze menyediakan rute, controller, view (Blade atau Inertia), dan utilitas lain untuk registrasi, login, logout, verifikasi email (opsional), dan reset password. Breeze cocok dipakai sebagai dasar implementasi sistem autentikasi yang simpel dan mudah dikustomisasi.

Instalasi singkat:

```
composer require laravel/breeze --dev
php artisan breeze:install
npm install
php artisan migrate
```

1.3 Middleware, Guards, dan Session

- **Middleware** seperti `auth` dapat melindungi rute dan memastikan hanya pengguna terautentikasi yang mengaksesnya.
- **Guards** menentukan mekanisme autentikasi (mis. session, token).
- Saat login, Laravel menyimpan status sesi (session) di cookie yang aman; middleware `auth` memeriksa session ini untuk mengizinkan/menolak akses.

2. Langkah-Langkah Praktikum

2.1 Praktikum 1 — Autentikasi dan Otorisasi dengan Laravel 12 Breeze

Tujuan praktik (implisit):

Mengimplementasikan sistem autentikasi dasar menggunakan Laravel Breeze (registrasi, login, logout, reset password) dan membuat rute yang dilindungi untuk menampilkan profil pengguna.

Langkah-langkah:

1. Buat proyek Laravel baru:

```
adhag@Adhagusti MINGW64 /c/web_lanjut/web-lanjut-2024573010009/projects (main)
$ laravel new auth-lab
```

2. Pastikan ekstensi MySQL aktif pada `php.ini` lalu buat database `authlab_db` di MySQL:

```
CREATE DATABASE authlab_db;
```

Konfigurasi `.env`:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=authlab_db
DB_USERNAME=root
DB_PASSWORD=
```

Clear config cache:

```
php artisan config:clear
```

3. Instal Laravel Breeze:

```
composer require laravel/breeze --dev
```

```
php artisan breeze:install
```

```
adhag@Adhagusti MINGW64 /c/web_lanjut/web-lanjut-2024573010009/projects/auth-lab (main)
$ php artisan breeze:install
public/build/manifest.json 0.33 kB | gzip: 0.17 kB
public/build/assets/app-A9A49wRH.css 38.00 kB | gzip: 6.93 kB
public/build/assets/app-CJy8ASEk.js 80.95 kB | gzip: 30.35 kB
✓ built in 2.07s

[INFO] Breeze scaffolding installed successfully.
```

```
npm install
```

```
adhag@Adhagusti MINGW64 /c/web_lanjut/web-lanjut-2024573010009/projects/auth-lab (main)
$ npm install
  run `npm fund` for details

3 high severity vulnerabilities

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
```

```
php artisan migrate
```

```
adhag@Adhagusti MINGW64 /c/web_lanjut/web-lanjut-2024573010009/projects/auth-lab (main)
$ php artisan migrate

[INFO] Nothing to migrate.
```

4. Jalankan server development:

```
php artisan serve
```

Akses <http://localhost:8000> → register dan login menggunakan form yang disediakan oleh Breeze.

5. Tambahkan rute profil yang dilindungi di `routes/web.php`:

```
projects > auth-lab > routes > web.php
1  <?php
2
3  use App\Http\Controllers\ProfileController;
4  use Illuminate\Support\Facades\Route;
5  use Illuminate\Support\Facades\Auth;
6
7  Route::get('/', function () {
8      return view('welcome');
9  });
10
11 Route::get('/dashboard', function () {
12     return view('dashboard');
13 })->middleware(['auth', 'verified'])->name('dashboard');
14
15 Route::middleware('auth')->group(function () {
16     Route::get('/profile', [ProfileController::class, 'edit'])->name('profile.edit');
17     Route::patch('/profile', [ProfileController::class, 'update'])->name('profile.update');
18     Route::delete('/profile', [ProfileController::class, 'destroy'])->name('profile.destroy');
19
20     // Tambahkan rute myprofile baru
21     Route::get('/myprofile', function () {
22         return Auth::user();
23     })->name('myprofile');
24 });
25
26 require __DIR__.'/auth.php';
27
```

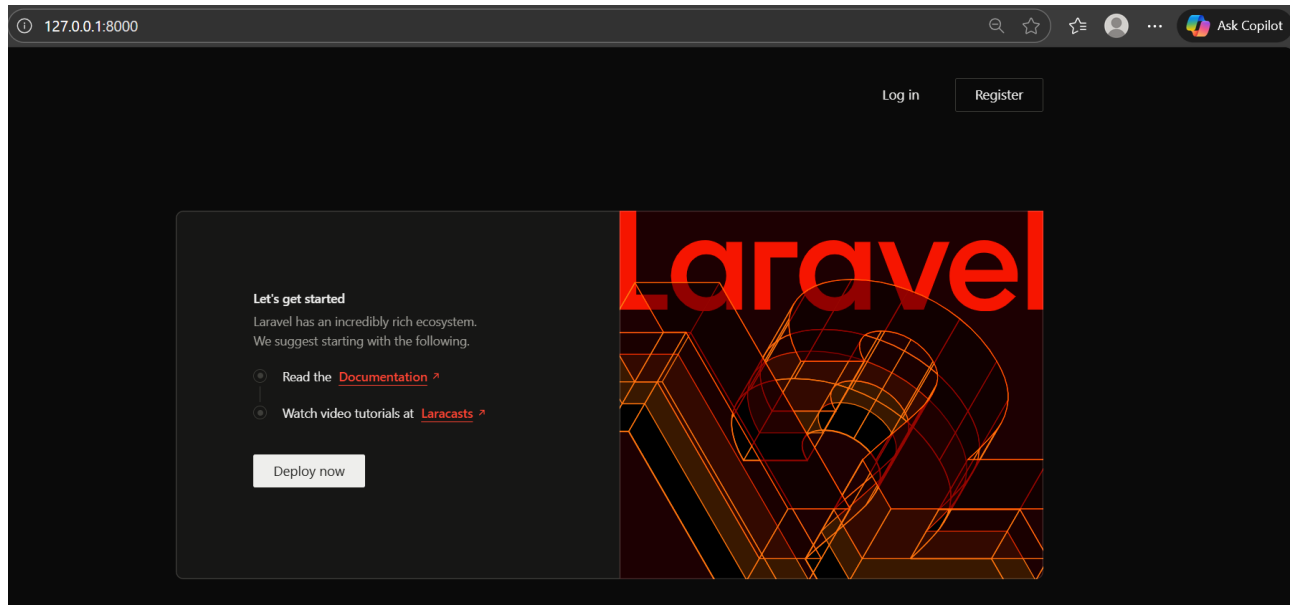
6. Testing rute `/myprofile`:

- Login terlebih dahulu.
- Kunjungi <http://localhost:8000/myprofile>
- Verifikasi JSON user menampilkan `id`, `name`, `email`, `email_verified_at`, `created_at`, `updated_at`.

```
localhost:8000/myprofile
Pretty-print
{"id":2,"name":"Adha Gusti Harmadhan","email":"adhagustiharmadan@gmail.com","email_verified_at":null,"created_at":"2025-11-18T06:52:55.000000Z","updated_at":"2025-11-18T06:52:55.000000Z"}
```

Hasil Pengujian:

- <http://localhost:8000> Pengguna dapat mendaftar (register), login, logout, dan reset password lewat tampilan Breeze.



- Rute </myprofile> hanya bisa diakses oleh pengguna yang sudah login dan menampilkan informasi user.

```
localhost:8000/myprofile
Pretty-print
{"id":2,"name":"Adha Gusti Harmadhan","email":"adhagustiharmadan@gmail.com","email_verified_at":null,"created_at":"2025-11-18T06:52:55.000000Z","updated_at":"2025-11-18T06:52:55.000000Z"}
```

2.2 Praktikum 2 — Membatasi Akses Berdasarkan Peran (RBAC) di Laravel 12

Tujuan praktik (implisit):

Mengimplementasikan kontrol akses berbasis peran sederhana (admin, manager, user) menggunakan field [role](#) di tabel [users](#), middleware kustom [RoleMiddleware](#), dan view terproteksi per peran.

Langkah-langkah:

1. Buat proyek Laravel baru:

```
adhag@Adhagusti MINGW64 /c/web_lanjut/web-lanjut-2024573010009/projects (main)
$ laravel new role-lab
```

2. Buat database [authrole_db](#) dan konfigurasi [.env](#):

```
CREATE DATABASE authrole_db;
```

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=authrole_db
DB_USERNAME=root
DB_PASSWORD=
```

Clear config:

```
php artisan config:clear
```

```
adhag@Adhagusti MINGW64 /c/web_lanjut/web-lanjut-2024573010009/projects/role-lab (main)
$ php artisan config:clear
```

```
INFO Configuration cache cleared successfully.
```

3. Instal Laravel Breeze (opsional untuk auth scaffolding):

```
composer require laravel/breeze --dev
```

```
adhag@Adhagusti MINGW64 /c/web_lanjut/web-lanjut-2024573010009/projects/role-lab (main)
$ composer require laravel/breeze --dev
```

```
INFO No publishable resources for tag [laravel-assets].
```

```
https://repo.packagist.org could not be fully loaded (curl error 6 while downloading https://repo.packagist.org/p2/brianium/paratest.json: Could not resolve host: repo.packagist.org), package information was loaded from the local cache and may be out of date
```

```
Found 1 security vulnerability advisory affecting 1 package.
```

```
Run "composer audit" for a full list of advisories.
```

```
Using version ^2.3 for laravel/breeze
```

```
php artisan breeze:install
```

```
adhag@Adhagusti MINGW64 /c/web_lanjut/web-lanjut-2024573010009/projects/role-lab (main)
$ php artisan breeze:install
public/build/manifest.json      0.33 kB | gzip: 0.17 kB
public/build/assets/app-CglvTTw2.css 50.28 kB | gzip: 8.70 kB
public/build/assets/app-CJy8ASEk.js 80.95 kB | gzip: 30.35 kB
✓ built in 1.89s

INFO Breeze scaffolding installed successfully.
```

```
npm install
```

```
adhag@Adhagusti MINGW64 /c/web_lanjut/web-lanjut-2024573010009/projects/role-lab (main)
$ npm install
  run `npm fund` for details

3 high severity vulnerabilities

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
```

```
php artisan migrate
```

```
adhag@Adhagusti MINGW64 /c/web_lanjut/web-lanjut-2024573010009/projects/role-lab (main)
$ php artisan migrate
INFO Preparing database.

Creating migration table ..... 228.68ms DONE

INFO Running migrations.

0001_01_01_000000_create_users_table ..... 152.26ms DONE
0001_01_01_000001_create_cache_table ..... 52.35ms DONE
0001_01_01_000002_create_jobs_table ..... 149.26ms DONE
```

4. Tambahkan field **role** pada tabel **users**:


Buat migration:

```
php artisan make:migration add_role_to_users_table --table=users
```

```
adhag@Adhagusti MINGW64 /c/web_lanjut/web-lanjut-2024573010009/projects/role-lab (main)
$ php artisan make:migration add_role_to_users_table --table=users

INFO Migration [C:\web_lanjut\web-lanjut-2024573010009\projects\role-lab\database\migrations\2025_11_18_073323_add_role_to_users_table.php] created successfully.
```

Edit migration:

```
projects > role-lab > database > migrations >  2025_11_18_073323_add_role_to_users_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /** * Run the migrations.*/
10     public function up(): void
11     {
12         Schema::table('users', function (Blueprint $table) {
13             $table->string('role')->default('user');
14         });
15     }
16     /** * Reverse the migrations.*/
17     public function down(): void
18     {
19         Schema::table('users', function (Blueprint $table) {
20             $table->dropColumn('role');
21         });
22     }
23 };
24
```

Jalankan migration:

```
adhag@Adhagusti MINGW64 /c/web_lanjut/web-lanjut-2024573010009/projects/role-lab (main)
$ php artisan migrate

INFO Running migrations.

2025_11_18_073323_add_role_to_users_table ..... 15.77ms DONE
```

5. Seed pengguna dengan role berbeda (DatabaseSeeder):

Jalankan seeder:


```
projects > role-lab > database > seeders > DatabaseSeeder.php
1  <?php
2
3  namespace Database\Seeders;
4
5  use App\Models\User;
6  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
7  use Illuminate\Database\Seeder;
8  use Illuminate\Support\Facades\Hash;
9
10 class DatabaseSeeder extends Seeder
11 {
12     use WithoutModelEvents;
13
14     public function run(): void
15     {
16         User::create([
17             'name' => 'Admin User',
18             'email' => 'admin@ilmudata.id',
19             'password' => Hash::make('password123'),
20             'role' => 'admin',
21         ]);
22
23         User::create([
24             'name' => 'Manager User',
25             'email' => 'manager@ilmudata.id',
26             'password' => Hash::make('password123'),
27             'role' => 'manager',
28         ]);
29
30         User::create([
31             'name' => 'General User',
32             'email' => 'user@ilmudata.id',
33             'password' => Hash::make('password123'),
34             'role' => 'user',
35         ]);
36
37         User::factory()->create([
38             'name' => 'Test User',
39             'email' => 'test@example.com',
40         ]);
41     }
42 }
```

```
php artisan db:seed
```

```
adhag@Adhagusti MINGW64 /c/web_lanjut/web-lanjut-2024573010009/projects/role-lab (main)
$ php artisan db:seed

INFO Seeding database.
```

6. Buat middleware `RoleMiddleware`:

```
php artisan make:middleware RoleMiddleware
```

```
adhag@Adhagusti MINGW64 /c/web_lanjut/web-lanjut-2024573010009/projects/role-lab (main)
$ php artisan make:middleware RoleMiddleware


INFO Middleware [C:\web_lanjut\web-lanjut-2024573010009\projects\role-lab\app\Http\Middleware\RoleMiddleware.php] created successfully.
```

Edit `app/Http/Middleware/RoleMiddleware.php`:

```
projects > role-lab > app > Http > Middleware > RoleMiddleware.php
1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Closure;
6  use Illuminate\Http\Request;
7  use Symfony\Component\HttpFoundation\Response;
8
9  class RoleMiddleware
10 {
11     public function handle(Request $request, Closure $next, string $role): Response
12     {
13         if ($request->user() && $request->user()->role === $role) {
14             return $next($request);
15         }
16
17         abort(403, 'Unauthorized');
18     }
19 }
20
```

7. Daftarkan alias middleware `role` (contoh setup di `App\Http\Kernel.php` atau sesuai struktur proyek):

```


projects > role-lab > bootstrap >  app.php
1  <?php
2
3  use Illuminate\Foundation\Application;
4  use Illuminate\Foundation\Configuration\Exceptions;
5  use Illuminate\Foundation\Configuration\Middleware;
6
7  use App\Http\Middleware\RoleMiddleware;
8
9  return Application::configure(basePath: dirname(__DIR__))
10     ->withRouting(
11         web: __DIR__.'/../routes/web.php',
12         commands: __DIR__.'/../routes/console.php',
13         health: '/up',
14     )
15     ->withMiddleware(function (Middleware $middleware) {
16         //
17         $middleware->alias([
18             'role' => RoleMiddleware::class,
19         ]);
20     })
21     ->withExceptions(function (Exceptions $exceptions) {
22         //
23     })->create();
24

```

8. Buat view untuk tiap role di `resources/views`:

- `admin.blade.php`

```

projects > role-lab > resources > views >  admin.blade.php
1  <x-app-layout>
2      <x-slot name="header">
3          <h2 class="font-semibold text-xl text-gray-800 dark:text-gray-200 leading-tight">
4              {{ __('Admin Dashboard') }}
5          </h2>
6      </x-slot>
7
8      <div class="py-12">
9          <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
10             <div class="bg-white dark:bg-gray-800 overflow-hidden shadow-sm sm:rounded-lg">
11                 <div class="p-6 text-gray-900 dark:text-gray-100">
12                     {{ __("Welcome, Admin! You have full access.") }}
13                 </div>
14             </div>
15         </div>
16     </div>
17 </x-app-layout>
18

```

- `manager.blade.php`

```

projects > role-lab > resources > views > manager.blade.php
1  <x-app-layout>
2    <x-slot name="header">
3      <h2 class="font-semibold text-xl text-gray-800 dark:text-gray-200 leading-tight">
4        {{ __('Manager Dashboard') }}
5      </h2>
6    </x-slot>
7
8    <div class="py-12">
9      <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
10       <div class="bg-white dark:bg-gray-800 overflow-hidden shadow-sm sm:rounded-lg">
11         <div class="p-6 text-gray-900 dark:text-gray-100">
12           {{ __('Welcome, Manager! You can manage and monitor resources.') }}
13         </div>
14       </div>
15     </div>
16   </div>
17 </x-app-layout>

```

- user.blade.php

```

projects > role-lab > resources > views > user.blade.php
1  <x-app-layout>
2    <x-slot name="header">
3      <h2 class="font-semibold text-xl text-gray-800 dark:text-gray-200 leading-tight">
4        {{ __('User Dashboard') }}
5      </h2>
6    </x-slot>
7
8    <div class="py-12">
9      <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
10       <div class="bg-white dark:bg-gray-800 overflow-hidden shadow-sm sm:rounded-lg">
11         <div class="p-6 text-gray-900 dark:text-gray-100">
12           {{ __('Welcome, User! You have limited access.') }}
13         </div>
14       </div>
15     </div>
16   </div>
17 </x-app-layout>

```

- all.blade.php

```

projects > role-lab > resources > views > all.blade.php
1  <x-app-layout>
2    <x-slot name="header">
3      <h2 class="font-semibold text-xl text-gray-800 dark:text-gray-200 leading-tight">
4        {{ __('General Dashboard') }}
5      </h2>
6    </x-slot>
7
8    <div class="py-12">
9      <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
10       <div class="bg-white dark:bg-gray-800 overflow-hidden shadow-sm sm:rounded-lg">
11         <div class="p-6 text-gray-900 dark:text-gray-100">
12           {{ __('Welcome! This view is accessible by all authenticated roles.') }}
13         </div>
14       </div>
15     </div>
16   </div>
17 </x-app-layout>

```

9. Definisikan rute berbasis peran di `routes/web.php`:

```
projects > role-lab > routes > web.php
14 Route::middleware('auth')->group(function () {
15     Route::patch('/profile', [ProfileController::class, 'update'])->name('profile.update');
16     Route::delete('/profile', [ProfileController::class, 'destroy'])->name('profile.destroy');
17
18     // Rute yang dapat diakses oleh semua pengguna terautentikasi
19     Route::get('/all', function () {
20         return view('all');
21     });
22
23     // Rute khusus admin dengan middleware role
24     Route::get('/admin', function () {
25         return view('admin');
26     })->middleware('role:admin');
27
28     // Rute khusus manager dengan middleware role
29     Route::get('/manager', function () {
30         return view('manager');
31     })->middleware('role:manager');
32
33     // Rute khusus user dengan middleware role
34     Route::get('/user', function () {
35         return view('user');
36     })->middleware('role:user');
37 });
38
39
```

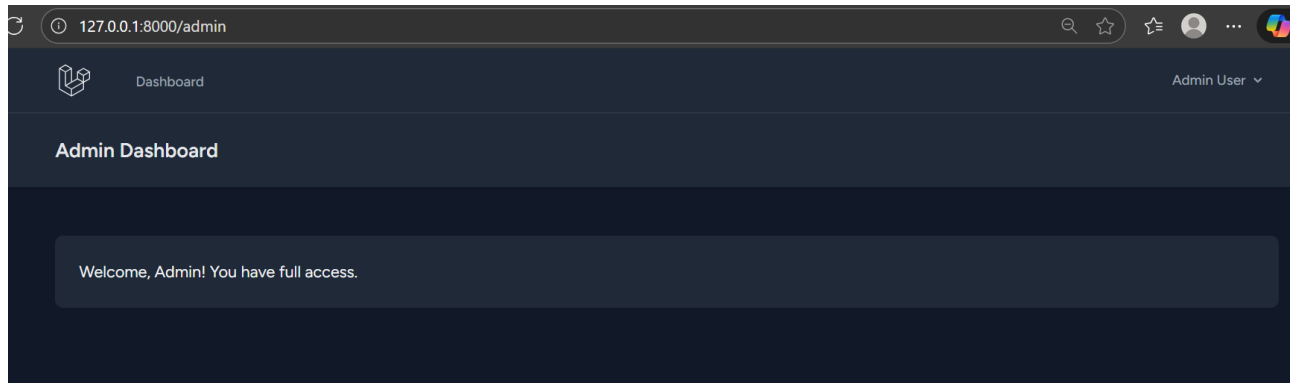
10. Jalankan server dan uji akses:

```
php artisan serve
```

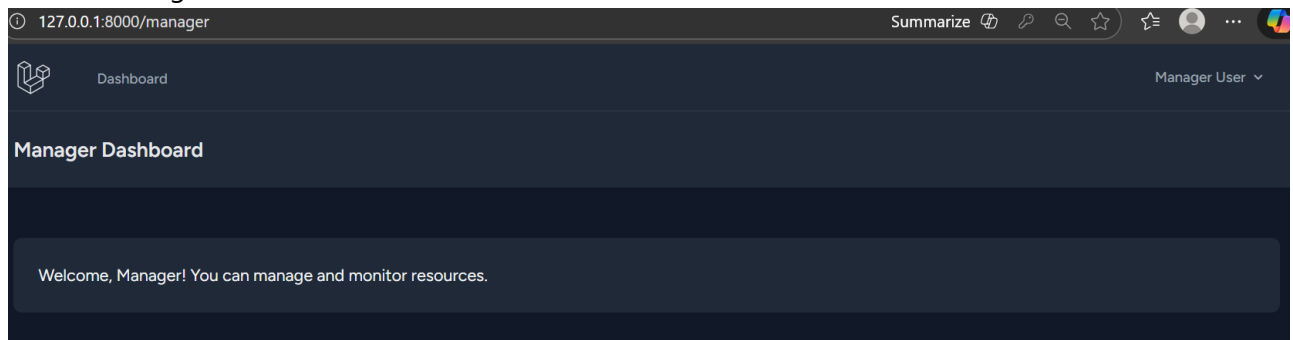
- Login dengan akun yang telah di-seed.
 - Admin: admin@ilmudata.id / password123
 - Manager: manager@ilmudata.id / password123
 - User: user@ilmudata.id / password123
- Akses:
 - `/admin` → hanya admin yang bisa masuk
 - `/manager` → hanya manager yang bisa masuk
 - `/user` → hanya user yang bisa masuk
 - `/all` → semua pengguna terautentikasi bisa mengakses

Hasil Pengujian:

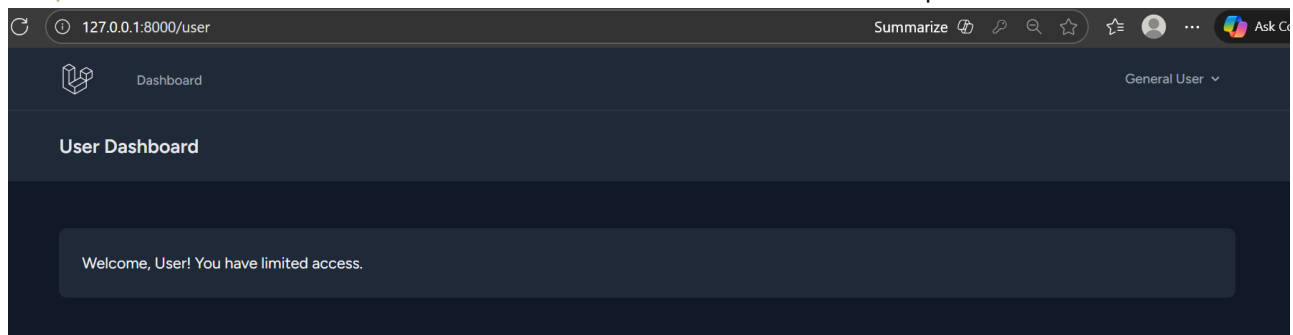
- `http://127.0.0.1:8000/admin` → berhasil diakses oleh role admin, menampilkan halaman khusus admin.



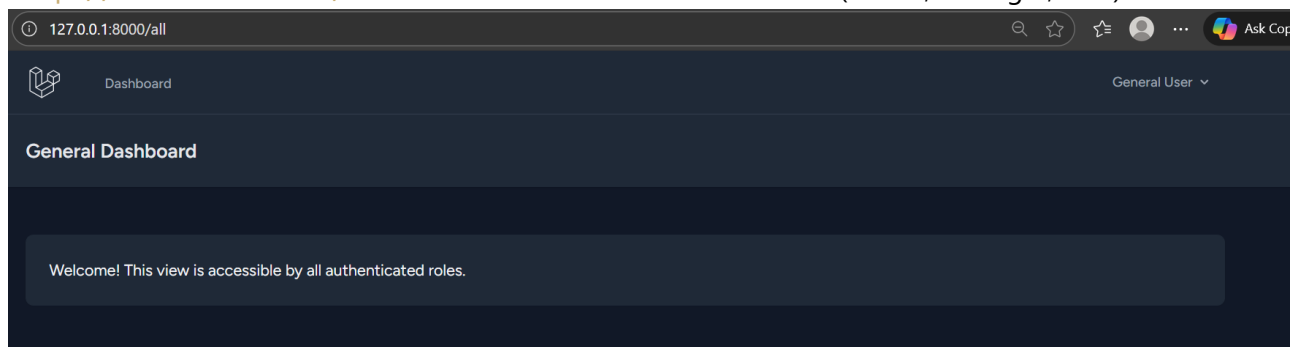
- <http://127.0.0.1:8000/manager> → berhasil diakses oleh role manager, menampilkan halaman khusus manager.



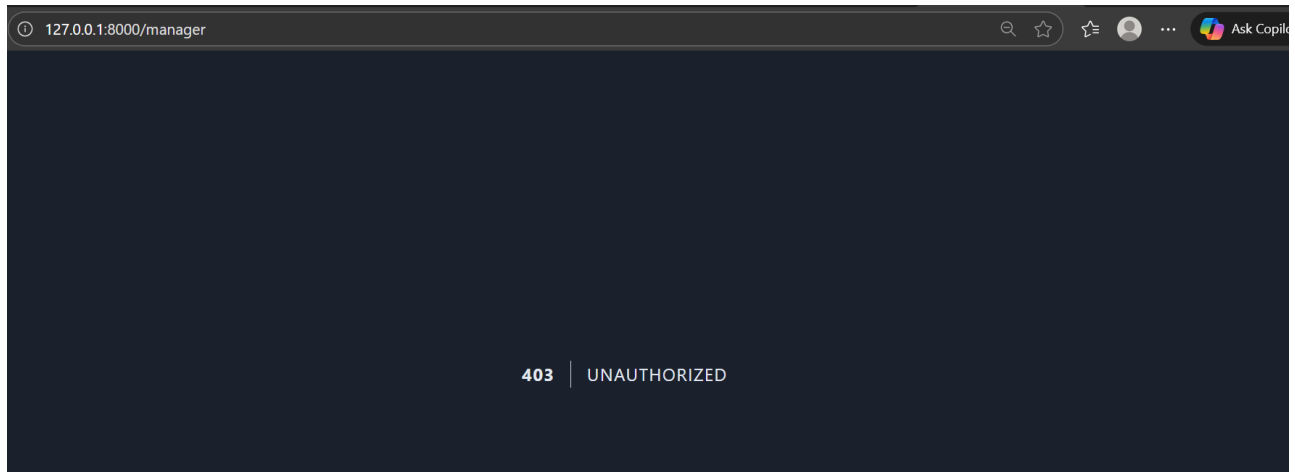
- <http://127.0.0.1:8000/user> → berhasil diakses oleh role user, menampilkan halaman khusus user.



- <http://127.0.0.1:8000/all> → berhasil diakses oleh semua role (admin, manager, user).



- Jika pengguna dengan role tidak sesuai mencoba membuka salah satu halaman di atas, sistem menampilkan HTTP 403 Forbidden



3. Hasil dan Pembahasan

1. **Autentikasi (Breeze):** Laravel Breeze menyediakan scaffolding autentikasi lengkap dan siap pakai sehingga implementasi registrasi, login, logout, dan reset password dapat dilakukan dengan cepat. Enkripsi password dilakukan otomatis menggunakan bcrypt saat menyimpan user.
2. **Rute Terproteksi:** Middleware `auth` efektif dalam membatasi akses rute hanya kepada pengguna yang sudah login. Rute `/myprofile` adalah contoh sederhana untuk memverifikasi status autentikasi dan menampilkan data user.
3. **RBAC Sederhana:** Menambahkan kolom `role` ke tabel `users` dan membuat middleware `RoleMiddleware` memungkinkan penerapan pembatasan akses berbasis peran tanpa paket tambahan. Pendekatan ini cocok untuk kebutuhan sederhana.
4. **Kelemahan Pendekatan Sederhana:** Mengelola role sebagai string di kolom `role` memadai untuk skenario kecil, tetapi untuk aplikasi lebih besar sebaiknya menggunakan pendekatan yang lebih fleksibel seperti tabel roles & permissions (mis. paket Spatie) untuk granularitas dan kemudahan manajemen.
5. **Keamanan:** Pastikan selalu memvalidasi dan men-escape input, gunakan CSRF protection (sudah otomatis pada group `web`), dan gunakan HTTPS pada deployment produksi untuk menjaga keamanan sesi.

4. Kesimpulan

Dari praktikum Modul 8 ini dapat disimpulkan bahwa:

1. Laravel menyediakan alat yang lengkap untuk autentikasi (Breeze) dan otorisasi (Gates, Policies, middleware). Breeze mempercepat pembuatan fitur auth dasar, sedangkan middleware/policy/gate digunakan untuk otorisasi.
2. Middleware `auth` dan middleware kustom (`RoleMiddleware`) dapat digunakan untuk menerapkan proteksi rute dan kontrol akses berbasis peran.
3. Untuk kebutuhan skala kecil, menambah kolom `role` pada tabel `users` sudah cukup; untuk kebutuhan yang lebih kompleks, paket seperti Spatie Laravel Permission lebih direkomendasikan.

4. Praktikum ini memberikan pemahaman praktis tentang bagaimana mengamankan aplikasi Laravel dari sisi autentikasi dan otorisasi, serta fondasi untuk membangun sistem RBAC yang lebih matang.
-

5. Referensi

- Modul 8 - Authentication & Authorization — ((<https://hackmd.io/@mohdrzu/BypBawklWg>))
- Dokumentasi Laravel Breeze — (<https://laravel.com/docs/12.x/starter-kits#laravel-breeze>)