Laporan Modul 8: Authentication & Authorization

Mata Kuliah: Workshop Web Lanjut
Nama: Maila Aziza
NIM: 2024573010024 **Kelas:** TI-2C

Abstrak

Modul ini mengeksplorasi implementasi mekanisme keamanan berupa autentikasi dan otorisasi dalam kerangka kerja Laravel. Topik pembelajaran meliputi pemanfaatan Laravel Breeze sebagai starter kit untuk membangun fitur user registration, login mechanism, profile management, dan pembatasan resource menggunakan sistem middleware.

Dalam konteks praktik, pemahaman akan mencakup: teknik membangun sistem login yang robust dan aman dari berbagai jenis serangan, strategi proteksi routing dengan validasi akses, implementasi sistem role-based (admin, manager, user) untuk pembedaan hak istimewa, dan manajemen kontrol akses yang terpisah untuk setiap level pengguna. Secara keseluruhan, pembelajaran ini memberikan fondasi komprehensif untuk mengimplementasikan security framework berbasis identifikasi dan otorisasi pada aplikasi Laravel modern.

1. Dasar Teori

Konsep Autentikasi

Autentikasi merupakan mekanisme penyesuaian identitas untuk memverifikasi apakah individu yang mengakses adalah benar-benar orang yang mengklaim dirinya. Proses ini dimulai ketika pengguna mencoba masuk ke sistem dengan menyediakan kredensial (username dan password). Sistem kemudian membandingkan data yang diberikan dengan informasi yang tersimpan di database. Jika keduanya cocok, pengguna diizinkan mengakses sistem; jika tidak, akses ditolak.

Konsep Otorisasi

Otorisasi adalah tahap berikutnya setelah autentikasi berhasil, di mana sistem menentukan tindakan apa yang boleh dilakukan oleh pengguna yang telah terverifikasi. Mekanisme ini mengontrol akses ke fitur dan resource berdasarkan peran atau permission yang dimiliki pengguna. Sebagai ilustrasi, pengguna biasa tidak memiliki izin mengakses panel administrator atau mengubah setting sistem, meskipun mereka telah berhasil login. Otorisasi menciptakan lapisan keamanan tambahan untuk melindungi data dan fungsi-fungsi kritis aplikasi.

2. Langkah-Langkah Praktikum

Berikut ini adalah tahapan praktik yang telah dilaksanakan, dilengkapi dengan potongan kode dan hasil tangkapan layar.

2.1 Praktikum 1 – Sistem Autentikasi pada Laravel 12

Inisialisasi Proyek Baru

Buat proyek Laravel baru dengan nama `auth-lab`

Konfigurasi Koneksi Database

Pengaturan database dilakukan dalam file `.env` dengan konfigurasi berikut:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=authlab_db
DB_USERNAME=<username database anda>
DB_PASSWORD=<password database anda jika ada>
```

Setelah konfigurasi selesai, jalankan migrasi awal untuk membuat struktur database:

```
php artisan migrate
```

Pemasangan Laravel Breeze

Download package Laravel Breeze sebagai starter kit autentikasi:

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

Lalu lakukan instalasi ke dalam proyek:

```
php artisan breeze:install
```

Pasang dependency frontend yang diperlukan:

```
npm install
npm run dev
```

Pembuatan Route Profil Terproteksi

Buka file `routes/web.php` untuk mendefinisikan rute-rute dengan proteksi middleware:

```bash
<?php

  use App\Http\Controllers\ProfileController;
```

```
  use Illuminate\Support\Facades\Route;
  use Illuminate\Support\Facades\Auth;

  Route::get('/', function () {
      return view('welcome');
  });

  Route::get('/dashboard', function () {
      return view('dashboard');
  })->middleware(['auth', 'verified'])->name('dashboard');

  Route::middleware('auth')->group(function () {
      Route::get('/profile', [ProfileController::class, 'edit'])-
>name('profile.edit');
      Route::patch('/profile', [ProfileController::class, 'update'])-
>name('profile.update');
      Route::delete('/profile', [ProfileController::class, 'destroy'])-
>name('profile.destroy');

      // Tambahkan rute myprofile baru
      Route::get('/myprofile', function () {
          return Auth::user();
      })->name('myprofile');
  });

  require __DIR__.'/auth.php';
```

Menjalankan Aplikasi

Launch development server Laravel:

```
php artisan serve
```

Verifikasi Hasil

Setelah user berhasil melakukan login, halaman dashboard akan menampilkan informasi seperti pada tangkapan layar berikut:


auth-lab

Apabila pengguna belum melakukan proses login, sistem akan menghalangi akses dan mengalihkan secara otomatis ke halaman autentikasi.

2.2 Praktikum 2 – Implementasi Sistem Kontrol Akses Berbasis Peran

Inisialisasi Proyek Baru

Buat proyek Laravel baru dengan nama `auth-role`

Pengaturan Database

Konfigurasi koneksi database dalam file `.env`:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=authrole_db
DB_USERNAME=<username database anda>
DB_PASSWORD=<password database anda jika ada>
```

Jalankan migrasi untuk membuat struktur database:

```
php artisan migrate
```

Pemasangan Laravel Breeze

Unduh dan pasang Laravel Breeze:

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

Penambahan Kolom Role pada Tabel Users

Untuk memperluas struktur tabel users dengan menambahkan kolom role, jalankan command pembuatan migration:

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

Ubah isi file migration yang baru dibuat:

```
public function up(): void
{
    Schema::table('users', function (Blueprint $table) {
        $table->string('role')->default('user');
    });
}

public function down(): void
```

```
{
    Schema::table('users', function (Blueprint $table) {
        $table->dropColumn('role');
    });
}
```

Setelah selesai, jalankan migration:

```
php artisan migrate
```

Penyisipan Data User dengan Role Berbeda

Untuk menambahkan akun user dengan role yang berbeda, edit file `database/seeders/DatabaseSeeder.php`:

```
User::create([
    'name' => 'Admin User',
    'email' => 'admin@ilmudata.id',
    'password' => Hash::make('password123'),
    'role' => 'admin',
]);

User::create([
    'name' => 'Manager User',
    'email' => 'manager@ilmudata.id',
    'password' => Hash::make('password123'),
    'role' => 'manager',
]);

User::create([
    'name' => 'General User',
    'email' => 'user@ilmudata.id',
    'password' => Hash::make('password123'),
    'role' => 'user',
]);
```

Jalankan seeder untuk memasukkan data ke database:

```
php artisan db:seed
```

Pembuatan Middleware untuk Pengecekan Role

Generate middleware baru:

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

Implementasikan logika pengecekan role di dalam method handle pada file `app/Http/Middleware/RoleMiddleware.php`:

```
if ($request->user() && $request->user()->role === $role) {
    return $next($request);
}

abort(403, 'Unauthorized');
```

Daftarkan middleware pada file `bootstrap/app.php` di dalam method withMiddleware():

```
$middleware->alias([
    'role' => RoleMiddleware::class,
]);
```

Pembuatan View untuk Setiap Role

Buat beberapa file view di direktori `resources/views` untuk menampilkan dashboard yang berbeda berdasarkan role pengguna.

File admin.blade.php:

```
<x-app-layout>
    <x-slot name="header">
        <h2 class="font-semibold text-xl text-gray-800 dark:text-gray-200 leading-
tight">
            {{ __('Admin Dashboard') }}
        </h2>
    </x-slot>

    <div class="py-12">
        <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
            <div class="bg-white dark:bg-gray-800 overflow-hidden shadow-sm
sm:rounded-lg">
                <div class="p-6 text-gray-900 dark:text-gray-100">
                    {{ __("Welcome, Admin! You have full access.") }}
                </div>
            </div>
        </div>
    </div>
</x-app-layout>
```

File manager.blade.php:

```
<x-app-layout>
    <x-slot name="header">
        <h2 class="font-semibold text-xl text-gray-800 dark:text-gray-200 leading-
tight">
            {{ __('Manager Dashboard') }}
        </h2>
    </x-slot>

    <div class="py-12">
        <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
            <div class="bg-white dark:bg-gray-800 overflow-hidden shadow-sm
sm:rounded-lg">
                <div class="p-6 text-gray-900 dark:text-gray-100">
                    {{ __("Welcome, Manager! You can manage and monitor
resources.") }}
                </div>
            </div>
        </div>
    </div>
</x-app-layout>
```

File user.blade.php:

```
<x-app-layout>
    <x-slot name="header">
        <h2 class="font-semibold text-xl text-gray-800 dark:text-gray-200 leading-
tight">
            {{ __('User Dashboard') }}
        </h2>
    </x-slot>

    <div class="py-12">
        <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
            <div class="bg-white dark:bg-gray-800 overflow-hidden shadow-sm
sm:rounded-lg">
                <div class="p-6 text-gray-900 dark:text-gray-100">
                    {{ __("Welcome, User! You have limited access.") }}
                </div>
            </div>
        </div>
    </div>
</x-app-layout>
```

File all.blade.php:

```
<x-app-layout>
    <x-slot name="header">
        <h2 class="font-semibold text-xl text-gray-800 dark:text-gray-200 leading-
```

```
tight">
            {{ __('General Dashboard') }}
        </h2>
    </x-slot>

    <div class="py-12">
        <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
            <div class="bg-white dark:bg-gray-800 overflow-hidden shadow-sm
sm:rounded-lg">
                <div class="p-6 text-gray-900 dark:text-gray-100">
                    {{ __("Welcome! This view is accessible by all authenticated
roles.") }}
                </div>
            </div>
        </div>
    </div>
</x-app-layout>
```

Definisi Routing Berbasis Role

Buka file `routes/web.php` dan tambahkan rute-rute untuk setiap role:

```php
<?php

use Illuminate\Support\Facades\Route;

Route::middleware('auth')->group(function () {
    // Rute yang dapat diakses oleh semua pengguna terautentikasi
    Route::get('/all', function () {
        return view('all');
    });

    // Rute khusus admin dengan middleware role
    Route::get('/admin', function () {
        return view('admin');
    })->middleware('role:admin');

    // Rute khusus manager dengan middleware role
    Route::get('/manager', function () {
        return view('manager');
    })->middleware('role:manager');

    // Rute khusus user dengan middleware role
    Route::get('/user', function () {
        return view('user');
    })->middleware('role:user');
});
```

Penjelasan Struktur Routing:

- `/all` - Dapat diakses oleh semua pengguna yang sudah melakukan autentikasi
- `/admin` - Hanya dapat diakses oleh pengguna dengan role admin
- `/manager` - Hanya dapat diakses oleh pengguna dengan role manager
- `/user` - Hanya dapat diakses oleh pengguna dengan role user

Middleware yang Digunakan:

- `auth` - Memastikan pengguna telah berhasil login
- `role:admin` - Memastikan pengguna memiliki peran admin
- `role:manager` - Memastikan pengguna memiliki peran manager
- `role:user` - Memastikan pengguna memiliki peran user

Menjalankan Aplikasi dan Melakukan Pengujian

Jalankan development server:

```
php artisan serve
```

Buka browser dan kunjungi `http://localhost:8000`, kemudian login menggunakan salah satu akun yang telah dibuat dalam seeding. Setelah berhasil login, coba akses rute-rute berikut untuk menguji sistem kontrol akses:

- `/all` - Halaman umum yang dapat diakses semua role
- `/admin` - Halaman khusus admin
- `/manager` - Halaman khusus manager
- `/user` - Halaman khusus user

Hasil Pengujian

Berikut adalah beberapa skenario pengujian yang dilakukan:

Skenario 1: Admin mengakses halaman admin

 role-lab

Skenario 2: Admin mengakses halaman all (dapat diakses)

 role-lab

Skenario 3: Admin mencoba mengakses halaman manager (ditolak)

 role-lab

---

3. Kesimpulan

Melalui pembelajaran modul ini, mekanisme autentikasi dan otorisasi dalam ekosistem Laravel telah dieksplorasi dan diterapkan secara praktis. Alur pembelajaran mencakup tahapan login/registrasi pengguna, pengelolaan profil, hingga implementasi kontrol akses yang berbasis peran dan sistem middleware.

Laravel Breeze memberikan kemudahan signifikan dalam membangun fondasi autentikasi yang sudah siap pakai, mengurangi kompleksitas setup awal. Sebaliknya, implementasi middleware khusus dan sistem role-based memungkinkan developer untuk membangun aturan akses yang lebih sophisticated dan sesuai dengan kebutuhan bisnis.

Hasil praktikum membuktikan bahwa Laravel menyediakan tools dan infrastruktur yang mature untuk membangun sistem keamanan yang solid. Dengan arsitektur yang well-structured, developer dapat dengan percaya diri mengimplementasikan security layer yang melindungi data sensitif dan mengontrol alur akses di dalam aplikasi.

---

4. Referensi

Cantumkan sumber yang Anda baca (buku, artikel, dokumentasi) — minimal 2 sumber. Gunakan format sederhana (judul — URL).

Laravel Blade Templating Engine — https://hackmd.io/@mohdrzu/r1AIUzWpll Panduan Lengkap Authorization di Laravel 12 — https://qadrlabs.com/post/laravel-gate-panduan-lengkap-authorization-di-laravel-12 Membuat Autentikasi (Multi Role) Web Laravel Anti Ribet dengan Laravel Breeze — https://wahyuivan.medium.com/membuat-autentikasi-multi-role-web-laravel-anti-ribet-dengan-laravel-breeze-269ad99f1197
