

Laporan Modul 8: Authentication & Authorization

Mata Kuliah: Workshop Web Lanjut

Nama: [Lediania Berasa]

NIM: [2024573010034]

Kelas: [TI2C]

Abstrak

Laporan praktikum ini membahas penerapan sistem autentikasi dan otorisasi dalam framework Laravel. Tujuan utamanya adalah untuk memahami konsep dasar autentikasi (verifikasi identitas pengguna) dan otorisasi (kontrol akses berdasarkan peran) serta mengimplementasikannya menggunakan Laravel Breeze dan middleware kustom. Metode yang digunakan dalam praktikum ini meliputi instalasi Laravel Breeze untuk sistem autentikasi dasar, pembuatan middleware untuk kontrol akses berdasarkan peran, dan implementasi view yang berbeda untuk setiap peran pengguna. Hasil praktikum menunjukkan pemahaman yang baik tentang sistem keamanan Laravel dan kemampuan dalam mengimplementasikan sistem autentikasi dan otorisasi yang efektif untuk membatasi akses pengguna sesuai dengan perannya.

1. Dasar Teori

Autentikasi dan Otorisasi

Autentikasi dan otorisasi adalah dua komponen kritis dalam keamanan aplikasi web modern.

Autentikasi adalah proses verifikasi identitas pengguna, memastikan mereka adalah siapa yang mereka klaim. Dalam konteks aplikasi web, ini biasanya melibatkan proses login dengan username dan password, serta verifikasi tambahan seperti email atau two-factor authentication.

Otorisasi menentukan apa yang diizinkan untuk dilakukan oleh pengguna yang telah terautentikasi di dalam aplikasi. Ini melibatkan kontrol akses berdasarkan peran (Role-Based Access Control/RBAC) atau izin spesifik yang dimiliki oleh pengguna.

Laravel Breeze

Laravel Breeze adalah implementasi minimal dan sederhana dari fitur autentikasi Laravel. Ini mencakup rute, controller, dan view untuk login, registrasi, reset kata sandi, dan lainnya. Breeze menyediakan perancah (scaffolding) lengkap untuk sistem autentikasi menggunakan Blade atau Inertia.

Middleware dalam Laravel

Middleware adalah lapisan yang menyaring request HTTP sebelum mencapai controller. Dalam konteks autentikasi dan otorisasi, middleware digunakan untuk:

- Memastikan hanya pengguna yang terautentikasi yang dapat mengakses rute tertentu (middleware `auth`)
- Memeriksa peran pengguna untuk mengakses halaman tertentu (middleware kustom)

- Melindungi aplikasi dari serangan CSRF (middleware `web`)

Gates dan Policies

Sistem otorisasi Laravel didasarkan pada Gates dan Policies:

- **Gates** adalah closure yang menentukan apakah seorang pengguna diizinkan untuk melakukan tindakan tertentu
- **Policies** adalah kelas yang mengelompokkan logika otorisasi terkait model tertentu

Role-Based Access Control (RBAC)

RBAC adalah pendekatan otorisasi yang membatasi akses sistem berdasarkan peran pengguna. Dalam RBAC:

- **Peran (Roles)** didefinisikan seperti "admin", "manager", "user"
- **Izin (Permissions)** ditetapkan ke peran-peran tersebut
- Pengguna ditetapkan ke satu atau beberapa peran
- Akses ke sumber daya ditentukan oleh peran pengguna

2. Langkah-Langkah Praktikum

Praktikum 1 - Autentikasi dan Otorisasi dengan Laravel Breeze

Langkah 1: Buat dan Buka Proyek Laravel

1. Membuat proyek Laravel baru dengan perintah:

```
laravel new auth-lab  
cd auth-lab  
code .
```

2. Membuat database baru dengan nama `authlab_db` melalui phpMyAdmin.
3. Mengkonfigurasi koneksi database di file `.env`:

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

4. Membersihkan config cache:

```
php artisan config:clear
```

Langkah 2: Instalasi Laravel Breeze

1. Menginstal Laravel Breeze:

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

2. Menginstal Breeze:

```
php artisan breeze:install
```

3. Selama proses instalasi, memilih opsi:

- Frontend framework: blade
- Dark mode: yes (opsional)

4. Melanjutkan dengan perintah:

```
npm install  
php artisan migrate
```

Langkah 3: Akses Register and Login via Web Interface

1. Menjalankan server development:

```
php artisan serve
```

2. Mengakses aplikasi melalui browser di http://localhost:8000

3. Mendaftarkan pengguna baru dengan mengklik tautan Register dan mengisi formulir pendaftaran

4. Verifikasi bahwa setelah mendaftar, pengguna diarahkan ke halaman dashboard

5. Mencoba logout dan login kembali menggunakan kredensial yang baru didaftarkan

Langkah 4: Membuat Rute Profil yang Dilindungi

1. Mengedit file routes/web.php:

```
<?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';

```

Langkah 5: Testing Rute Baru

1. Login ke aplikasi
2. Mengunjungi <http://localhost:8000/myprofile>
3. Memverifikasi bahwa data user dalam format JSON ditampilkan:
 - o id
 - o name
 - o email
 - o email_verified_at
 - o created_at
 - o updated_at
4. Mencoba mengedit profil dengan mengklik tautan Profile dan memperbarui nama, email, atau kata sandi

Praktikum 2 - Membatasi Akses Berdasarkan Peran di Laravel

Langkah 1: Buat dan Buka Proyek Laravel

1. Membuat proyek Laravel baru:

```

laravel new role-lab
cd role-lab
code .

```

2. Membuat database baru dengan nama **authrole_db** melalui phpMyAdmin.
3. Mengkonfigurasi koneksi database di file .env:

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

4. Membersihkan config cache:

```
php artisan config:clear
```

Langkah 2: Instalasi Laravel Breeze

1. Menginstal Laravel Breeze:

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

2. Menginstal Breeze:

```
php artisan breeze:install
```

3. Selama proses instalasi, memilih opsi:

- Frontend framework: blade
- Dark mode: yes (opsional)

4. Melanjutkan dengan perintah:

```
npm install  
php artisan migrate
```

Langkah 3: Menambahkan Field Role ke Tabel Users

1. Membuat migrasi untuk menambahkan field role:

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

2. Mengedit file migrasi:

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    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');
        });
    }
};
```

3. Menjalankan migrasi:

```
php artisan migrate
```

Langkah 4: Seeding Pengguna dengan Peran Berbeda

1. Mengedit file database/seeders/DatabaseSeeder.php:

```
use App\Models\User;
use Illuminate\Support\Facades\Hash;

public function run(): void
{
    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',
    ]);
}
```

2. Menjalankan seeder:

```
php artisan db:seed
```

Langkah 5: Membuat Role Middleware

1. Membuat middleware:

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

2. Mengedit file app/Http/Middleware/RoleMiddleware.php:

```
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Symfony\Component\HttpFoundation\Response;

class RoleMiddleware
{
    public function handle(Request $request, Closure $next, string $role): Response
    {
        if ($request->user() && $request->user()->role === $role) {
            return $next($request);
        }

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

3. Mendaftarkan middleware di bootstrap/app.php:

```
<?php

use Illuminate\Foundation\Application;
```

```

use Illuminate\Foundation\Configuration\Exceptions;
use Illuminate\Foundation\Configuration\Middleware;

use App\Http\Middleware\RoleMiddleware;

return Application::configure(basePath: dirname(__DIR__))
    ->withRouting(
        web: __DIR__.'/../routes/web.php',
        commands: __DIR__.'/../routes/console.php',
        health: '/up',
    )
    ->withMiddleware(function (Middleware $middleware) {
        $middleware->alias([
            'role' => RoleMiddleware::class,
        ]);
    })
    ->withExceptions(function (Exceptions $exceptions) {
        //
    })->create();

```

Langkah 6: Membuat View untuk Setiap Role

1. Membuat file resources/views/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>
    </x-app-layout>

```

2. Membuat file resources/views/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>
```

3. Membuat file resources/views/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>
```

4. Membuat file resources/views/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>
</x-app-layout>

```

Langkah 7: Mendefinisikan Rute untuk View Berbasis Peran

1. Mengedit file routes/web.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');
});


```

Langkah 8: Menjalankan dan Menguji

1. Menjalankan server development:

```
php artisan serve
```

2. Mengunjungi <http://localhost:8000> dan login menggunakan pengguna yang telah disediakan:



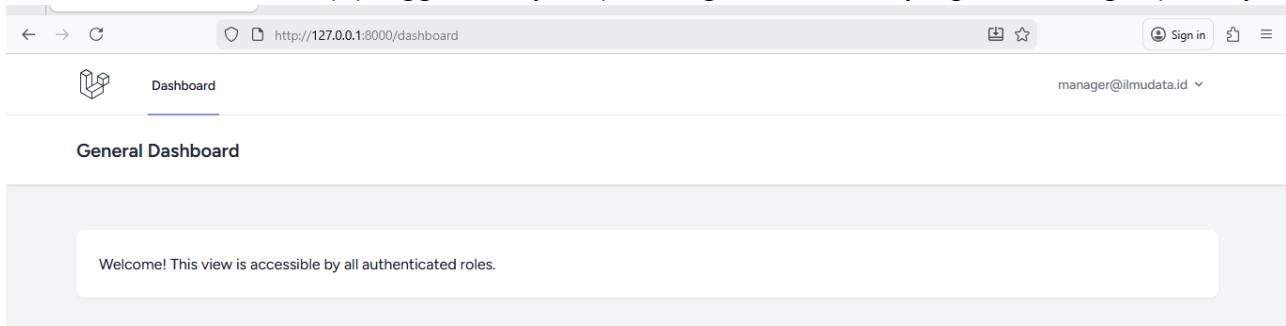
Email

Password

Remember me

[Forgot your password?](#) LOG IN

3. Memverifikasi bahwa setiap pengguna hanya dapat mengakses halaman yang sesuai dengan perannya



The screenshot shows a web browser window with the URL <http://127.0.0.1:8000/dashboard>. The page title is "Dashboard". On the right side, there is a user profile section with the email "manager@ilmudata.id". The main content area is titled "General Dashboard" and contains a message: "Welcome! This view is accessible by all authenticated roles."

3. Hasil dan Pembahasan

Hasil Praktikum 1 - Autentikasi dengan Laravel Breeze

Aplikasi autentikasi berjalan sesuai harapan dengan baik. Sistem registrasi, login, logout, dan manajemen profil berfungsi dengan benar. Beberapa poin penting dari hasil praktikum:

- 1. Registrasi Pengguna:** Formulir registrasi berhasil membuat pengguna baru dalam database dengan kata sandi yang dienkripsi menggunakan bcrypt.

2. **Proses Login:** Pengguna dapat login dengan kredensial yang telah didaftarkan, dan sistem berhasil memverifikasi identitas pengguna.
3. **Rute yang Dilindungi:** Middleware `auth` berhasil membatasi akses ke rute-rute tertentu seperti `/dashboard` dan `/myprofile`, hanya pengguna yang telah login yang dapat mengaksesnya.
4. **Manajemen Profil:** Pengguna dapat memperbarui informasi profil mereka termasuk nama, email, dan kata sandi.
5. **Session Management:** Sistem session berfungsi dengan baik, menjaga status login pengguna selama navigasi di aplikasi.

Hasil Praktikum 2 - Membatasi Akses Berdasarkan Peran

Aplikasi dengan sistem kontrol akses berbasis peran berjalan sesuai harapan. Setiap pengguna hanya dapat mengakses halaman yang sesuai dengan perannya. Beberapa poin penting dari hasil praktikum:

1. **Penambahan Field Role:** Field `role` berhasil ditambahkan ke tabel `users` dengan nilai default 'user'.
2. **Seeding Pengguna:** Tiga pengguna dengan peran berbeda (admin, manager, user) berhasil dibuat melalui seeder.
3. **Middleware Kustom:** RoleMiddleware berhasil dibuat dan didaftarkan, memungkinkan pemeriksaan peran pengguna sebelum mengizinkan akses ke rute tertentu.
4. **Pembatasan Akses:** Sistem berhasil membatasi akses berdasarkan peran:
 - Pengguna dengan peran admin dapat mengakses halaman `/admin`
 - Pengguna dengan peran manager dapat mengakses halaman `/manager`
 - Pengguna dengan peran user dapat mengakses halaman `/user`
 - Semua pengguna yang terautentikasi dapat mengakses halaman `/all`
5. **Penanganan Akses Ditolak:** Pengguna yang mencoba mengakses halaman yang tidak sesuai dengan perannya mendapatkan pesan error 403 (Unauthorized).

Pembahasan

1. **Keamanan Kata Sandi:** Laravel Breeze secara otomatis mengenkripsi kata sandi menggunakan bcrypt, yang merupakan praktik keamanan standar dalam pengembangan aplikasi web.
2. **Middleware dalam Autentikasi:** Middleware memainkan peran krusial dalam sistem autentikasi dan otorisasi Laravel. Middleware `auth` memastikan hanya pengguna yang terautentikasi yang dapat mengakses rute tertentu, sementara middleware kustom `role` memungkinkan kontrol akses yang lebih granular berdasarkan peran pengguna.
3. **Pentingnya Role-Based Access Control:** RBAC sangat penting untuk aplikasi dengan kompleksitas tinggi di mana pengguna dengan peran berbeda memerlukan tingkat akses yang berbeda. Ini membantu menjaga keamanan dan integritas data dengan membatasi akses hanya ke fungsi-fungsi yang diperlukan untuk setiap peran.
4. **Penggunaan Seeder:** Seeder sangat berguna untuk mengisi database dengan data awal, terutama untuk pengguna dengan peran berbeda. Ini memudahkan pengujian dan pengembangan aplikasi.

5. Tantangan yang Dihadapi:

- Pada praktikum pertama, tantangan adalah memahami alur kerja sistem autentikasi Laravel dan bagaimana middleware berinteraksi dengan rute.
 - Pada praktikum kedua, tantangan adalah membuat middleware kustom yang benar-benar memeriksa peran pengguna dan mendaftarkannya dengan benar di aplikasi.
-

4. Kesimpulan

Dari praktikum ini, saya mendapatkan pemahaman yang lebih baik tentang sistem autentikasi dan otorisasi dalam Laravel.

Praktikum pertama berhasil mengimplementasikan sistem autentikasi dasar menggunakan Laravel Breeze yang menyediakan scaffolding lengkap untuk fitur registrasi, login, logout, dan reset password. Saya mempelajari cara mengamankan kata sandi pengguna dengan enkripsi bcrypt secara otomatis, membuat rute yang dilindungi middleware auth, serta konfigurasi MySQL sebagai database. Hasilnya terbukti bahwa middleware auth efektif dalam membatasi akses hanya untuk pengguna yang telah terautentikasi, sementara fitur verifikasi email dan manajemen session berjalan dengan baik.

Praktikum kedua berfokus pada implementasi sistem kontrol akses berbasis peran (RBAC) yang lebih advanced dengan mendefinisikan tiga level pengguna yaitu admin, manager, dan user. Saya berhasil membuat middleware kustom untuk pengecekan peran, menambahkan field role pada tabel users, melakukan seeding data pengguna dengan peran berbeda, serta membuat view khusus untuk setiap peran. Sistem yang dibangun terbukti efektif dalam membatasi akses dimana setiap pengguna hanya dapat mengakses halaman yang sesuai dengan perannya saja, sementara halaman umum dapat diakses oleh semua pengguna terautentikasi.

Kedua praktikum ini memberikan fondasi kuat untuk pengembangan aplikasi web yang membutuhkan sistem keamanan standar dalam mengelola akses pengguna. Saya sekarang lebih siap untuk menghadapi tantangan dalam membangun aplikasi dengan sistem autentikasi dan otorisasi yang kompleks.

5. Referensi

1. Laravel Documentation - Authentication <https://laravel.com/docs/authentication>
2. Laravel Documentation - Authorization <https://laravel.com/docs/authorization>
3. Laravel Breeze Documentation <https://laravel.com/docs/starter-kits#laravel-breeze>
4. Laravel Middleware Documentation <https://laravel.com/docs/middleware>
5. Materi Workshop Web Lanjut (Google Share) <https://share.google/FcFzL3VQU36QE8MrM\>