

JOBSHEET 09

AUTHENTICATION, MIDDLEWARE

Nama : Raffy Jamil Octavialdy

Kelas : TI - 2H

NIM : 2241720082

Praktikum – Auth dan Middleware

1. Sebelum melangkah lebih lanjut setting file pada config/auth.php ubah pada bagian model menjadi model yang telah kita buat sebelumnya untuk menyimpan username dan password

```
'providers' => [  
    'users' => [  
        'driver' => 'eloquent',  
        'model' => App\Models\UserModel::class,  
    ],  
],
```

2. Membuat middleware dengan perintah pada cmd atau terminal

```
PS D:\Semester 4\Pemrograman Web Lanjut\PWL_POS> php artisan make:middleware Cek_login
```

```
INFO Middleware [D:\Semester 4\Pemrograman Web Lanjut\PWL_POS\app\Http\Middleware\Cek_login.php] created successfully.
```

Setelah itu cek didalam folder app/Http/Middleware apakah sudah ada file bernama Cek_login.php

```
class Cek_login  
{  
    /**  
     * Handle an incoming request.  
     *  
     * @param \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response) $next  
     */  
    public function handle(Request $request, Closure $next): Response  
    {  
        return $next($request);  
    }  
}
```

3. Kemudian tambahkan kode di bawah ini, letakkan di function handle()

```
class Cek_login
{
    /**
     * Handle an incoming request.
     *
     * @param \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response) $next
     */
    public function handle(Request $request, Closure $next, $roles): Response
    {
        // cek sudah login atau belum. Jika belum kembali ke halaman login
        if (!Auth::check()) {
            return redirect('login');
        }

        // simpan data user pada variabel $user
        $user = Auth::user();

        // jika user memiliki level sesuai pada kolom, lanjutkan request
        if ($user->level_id == $roles) {
            return $next($request);
        }

        // jika tidak memiliki akses maka kembalikan ke halaman login
        return redirect('login')->with('error', 'maaf anda tidak memiliki akses');
    }
}
```

4. Ketika sudah membuat middleware, langkah berikutnya harus diregistrasikan pada kernel.php. Agar middleware dapat dibaca oleh sistem. Buka folder App/Http/Kernel.php dan tambahkan code pada variabel \$middlewareAliases seperti dibawah

```
protected $middlewareAliases = [
    'auth' => \App\Http\Middleware\Authenticate::class,
    'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
    'precognitive' => \Illuminate\Foundation\Http\Middleware\HandlePrecognitiveRequests::class,
    'signed' => \App\Http\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
    // tambahkan middleware alias dengan nama cek_login
    'cek_login' => \App\Http\Middleware\Cek_login::class,
];
```

5. Membuat Controller Auth. Pada langkah ini kita akan membuat controller auth, yang dimana di dalamnya terdapat proses autentikasi dari request login. Jalankan perintah make:controller

```
PS D:\Semester 4\Pemrograman Web Lanjut\PWL_POS> php artisan make:controller AuthController
```

```
INFO Controller [D:\Semester 4\Pemrograman Web Lanjut\PWL_POS\app\Http\Controllers\AuthController.php] created successfully.
```

Buka file App/Http/Controllers/AuthController.php yang sudah kita buat. Lalu kita akan isi perintah code untuk membuat proses autentikasi , verifikasi, dan logout

6. Kemudian buat kode AuthController.php seperti dibawah ini

```
class AuthController extends Controller
{
    public function index()
    {
        // kita ambil data user lalu simpan pada variabel $user
        $user = Auth::user();

        // kondisi jika usernya ada
        if ($user) {
            // jika usernya memiliki level admin
            if ($user->level_id == '1') {
                return redirect()->intended('admin');
            }
            // jika usernya memiliki level manager
            else if ($user->level_id == '2') {
                return redirect()->intended('manager');
            }
        }
        return view('login');
    }

    public function proses_login(Request $request)
    {
        // kita buat validasi pada saat tombol login diklik
        // validasinya username & password wajib diisi
        $request->validate([
            'username' => 'required',
        ]);
    }
}
```

7. Kemudian tambahkan kode pada route pada file routes/web.php seperti kode program dibawah

```
Route::get('login', [AuthController::class, 'index'])->name('login');
Route::get('register', [AuthController::class, 'register'])->name('register');
Route::get('proses_login', [AuthController::class, 'proses_login'])->name('proses_login');
Route::get('logout', [AuthController::class, 'logout'])->name('logout');
Route::get('proses_register', [AuthController::class, 'proses_register'])->name('proses_register');

// kita atur juga untuk middleware menggunakan group pada routing
// di dalamnya terdapat group untuk mengecek kondisi login
// jika user yang login merupakan admin maka akan diarahkan ke AdminController
// jika user yang login merupakan manager maka akan diarahkan ke UserController

Route::group(['middleware' => ['auth']], function () {
    Route::group(['middleware' => ['cek_login:1']], function () {
        Route::resource('admin', AdminController::class);
    });
    Route::group(['middleware' => ['cek_login:2']], function () {
        Route::resource('manager', ManagerController::class);
    });
});
```

8. Kemudian buat file controller dengan nama AdminController sebagai halaman yang boleh diakses oleh admin saja. Jalankan perintah dibawah

```
INFO Controller [D:\Semester 4\Pemrograman Web Lanjut\PWL_POS\app\Http\Controllers\AdminController.php] created successfully.
```

9. Buka folder App/Http/Controllers dan isi file AdminController seperti dibawah

```
class AdminController extends Controller
{
    public function index()
    {
        return view('admin');
    }
}
```

10. Kemudian buat file controller dengan nama ManagerController sebagai halaman yang boleh diakses oleh admin saja. Jalankan perintah dibawah

```
PS D:\Semester 4\Pemrograman Web Lanjut\PWL_POS> php artisan make:controller ManagerController
```

```
INFO Controller [D:\Semester 4\Pemrograman Web Lanjut\PWL_POS\app\Http\Controllers\ManagerController.php] created successfully.
```

11. Buka folder App/Http/Controllers dan isi file ManagerController seperti dibawah

```
class ManagerController extends Controller
{
    public function index()
    {
        return view('manager');
    }
}
```

12. Pada langkah terakhir yaitu membuat layout sederhana

13. Buatlah halaman login dengan membuka folder resources/views dengan nama file login.blade.php. Lalu kita isi filenya seperti contoh dibawah

```
@extends('adminlte::auth.auth-page', ['auth_type' => 'login'])

@section('adminlte_css_pre')
    <link rel="stylesheet" href="{{ asset('vendor/icheck-bootstrap/icheck-bootstrap.min.css') }}">
@stop

@php( $login_url = View::getSection('login_url') ?? config('adminlte.login_url', 'login') )
@php( $register_url = View::getSection('register_url') ?? config('adminlte.register_url', 'register') )
@php( $password_reset_url = View::getSection('password_reset_url') ?? config('adminlte.password_reset_url', 'password/reset') )

@if (config('adminlte.use_route_url', false))
    @php( $login_url = $login_url ? route($login_url) : '' )
    @php( $register_url = $register_url ? route($register_url) : '' )
    @php( $password_reset_url = $password_reset_url ? route($password_reset_url) : '' )
@else
    @php( $login_url = $login_url ? url($login_url) : '' )
    @php( $register_url = $register_url ? url($register_url) : '' )
    @php( $password_reset_url = $password_reset_url ? url($password_reset_url) : '' )
@endif

@section('auth-header', __('adminlte::adminlte.login_message'))

@section('auth_body')
    @error('login_gagal')
        <div class="alert alert-warning alert-dismissible fade show" role="alert">
            <span class="alert-inner--text"><strong>Warning!</strong> {{ $message }} </span>
            <button type="button" class="close" data-dismiss="alert" aria-label="Close">

```

14. Buatlah halaman register dengan membuka folder resources/views dengan nama file register.blade.php. Lalu kita isi filenya seperti contoh dibawah

```
@extends('adminlte::auth.auth-page', ['auth_type' => 'register'])

@section('adminlte_css_pre')
    <link rel="stylesheet" href="{{ asset('vendor/icheck-bootstrap/icheck-bootstrap.min.css') }}">
@stop

@php( $login_url = View::getSection('login_url') ?? config('adminlte.login_url', 'login') )
@php( $register_url = View::getSection('register_url') ?? config('adminlte.register_url', 'register') )

@if (config('adminlte.use_route_url', false))
    @php( $login_url = $login_url ? route($login_url) : '' )
    @php( $register_url = $register_url ? route($register_url) : '' )
#else
    @php( $login_url = $login_url ? url($login_url) : '' )
    @php( $register_url = $register_url ? url($register_url) : '' )
@endif

@section('auth-header', __('adminlte::adminlte.register_message'))

@section('auth_body')
    <form action="{{ url('proses_register') }}" method="post">
        @csrf

        {{-- Nama field --}}
        <div class="input-group mb-3">
            <input type="text" name="nama" class="form-control @error('nama') is-invalid @enderror"
                value="{{ old('nama') }}" placeholder="Nama" autofocus>
        </div>
    </form>

```

15. Buatlah halaman register dengan membuka folder resources/views dengan nama file admin.blade.php. Lalu kita isi filenya seperti contoh dibawah

```
@extends('layouts.app')

{{-- Customize layout sections --}}

@section('subtitle', 'Admin')
@section('content_header_title', 'Home')
@section('content_header_subtitle', 'Admin')

@section('content')
<div class="container">
    <div class="card">
        <div class="card-header">Tampilan <?php echo (Auth::user()->level_id == 1) ? 'Admin' : 'Manager'; ?></div>
        <div class="card-body">
            <h1>Login Sebagai:
                <?php echo (Auth::user()->level_id == 1) ? 'Admin' : 'Manager'; ?>
            </h1>
            <a href="{{ route('logout') }}">Logout</a>
        </div>
    </div>
</div>
@endsection

@push('js')
@endpush

```

16. Buatlah halaman register dengan membuka folder resources/views dengan nama file manager.blade.php. Lalu kita isi filenya seperti contoh dibawah

```
@extends('layouts.app')

{{-- Customize layout sections --}}

@section('subtitle', 'Manager')
@section('content_header_title', 'Home')
@section('content_header_subtitle', 'Manager')

@section('content')
<div class="container">
    <div class="card">
        <div class="card-header">Tampilan <?php echo (Auth::user()->level_id == 1) ? 'Admin' : 'Manager'; ?></div>
        <div class="card-body">
            <h1>Login Sebagai:
                <?php echo (Auth::user()->level_id == 1) ? 'Admin' : 'Manager'; ?>
            </h1>
            <a href="{{ route('logout') }}">Logout</a>
        </div>
    </div>
</div>
</div>
@endsection

@push('js')
@endpush
```

17. Jalankan kode diatas dan buat laporan dengan isi screenshoot hasil dan beri penjelasan pada setiap screenshoot hasil dan commit perubahan pada git.



