

Nama : Ereen Lourenza Natalia Mamahi

Kelas / no : SIB3C / 09

NIM : 2141762044

Jobsheet 9

Authentication, Middleware

1	<p>Menyeting pada config/auth.php</p> <pre>'providers' => ['users' => ['driver' => 'eloquent', 'model' => App\Models\UserModel::class,], // 'users' => [// 'driver' => 'database', // 'table' => 'users', //],],</pre>
2	<p>Membuat middleware (<i>php artisan make:middleware Cek_Login</i>)</p> <pre>C:\laragon\www\Pemrograman_Web_Lanjut\Pertemuan 3\PWL_POS>php artis an make:middleware Cek_Login INFO Middleware [C:\laragon\www\Pemrograman_Web_Lanjut\Pertemua n 3\PWL_POS\app\Http\Middleware\Cek_Login.php] created successfully .</pre>
3	<p>Menambahkan kode di function handle()</p> <pre>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 pada 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'); } }</pre>
4	<p>Menambahkan kode cek_login sebagai registrasi agar middleware dapat dibaca sistem</p>

```
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 AuthController

```
public function index(){
    // kita ambil data user lalu simpan pada variable $user
    $user = Auth::user();

    // kondisi jika user nya ada
    if ($user){
        // jika user nya memiliki level admin
        if ($user->level_id == '1'){
            return redirect()->intended('admin');
        }
        // jika user nya 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 di klik
    // validasi nya username & password wajib di isi
    $request->validate([
        'username' => 'required',
        'password' => 'required'
    ]);

    // ambil data request username & password saja
    $credential = $request->only('username', 'password');
    // cek jika data username dan password valid (sesuai) dengan data
    if (Auth::attempt($credential)){
        // kalau berhasil simpan data user nya di variabel $user
        $user = Auth::user();

        // cek lagi jika level user admin maka arahkan ke halaman admin
        if ($user->level_id == '1'){
            //dd($user->level_id);
            return redirect()->intended('admin');
        }
        //tapi jika level user nya user biasa maka arahkan ke halaman user
        else if ($user->level_id == '2'){
            return redirect()->intended('manager');
        }
    }
    //tapi jika level user nya user biasa maka arahkan ke halaman user
    else if ($user->level_id == '2'){
        return redirect()->intended('manager');
    }
    // jika belum ada role maka ke halaman /
    return redirect()->intended('/');
}

// jika ga ada data user yang valid maka kembalikan lagi ke halaman login
// pastikan kirim pesan error juga kalau login gagal ya
return redirect('login')
->withInput()
->withErrors(['login_gagal' => 'Pastikan kembali username dan password yang dimasukkan sudah benar']);
}

public function register(){
    // tampilkan view register
    return view('register');
}
```

	<pre>// aksi form register public function proses_register(Request \$request){ // validasi semua field wajib di isi // validasi username unique \$validator = FacadesValidator::make(\$request->all(),['nama'=> 'required', 'username'=>'required unique:m_user', 'password'=>'required']); // kalau gagal kembali ke halaman register dan muncul pesna error if (\$validator ->fails()) { return redirect('/regsiter') ->withErrors(\$validator) ->withInput(); } // kalau berhasil isi level dan hash passwordnya biar secure \$request['level_id']='2'; \$request['password'] = Hash::make(\$request -> password); // masukkan semua data pada request ke tabel m_user UserModel::create(\$request->all()); // kalo berhasil arahkan ke halaman login return redirect()->route('login'); } public function logout(Request \$request){ // logout itu harus menghapus sessionnya \$request->session()->flush(); // jalankan fungsi logout pada auth Auth::logout(); // kembali ke halaman login return redirect('login'); }</pre>
6	<p>Menambahkan route</p> <pre>Route::get('login', [AuthController::class, 'index'])->name('login'); Route::get('register', [AuthController::class, 'register'])->name('register'); Route::post('proses_login', [AuthController::class, 'proses'])->name('proses_login'); Route::get('logout', [AuthController::class, 'logout'])->name('logout'); Route::post('proses_register', [AuthController::class, 'proses_register'])->name('proses_register'); 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('admin', ManagerController::class); }); });</pre>
7	<p>Membuat AdminController</p> <pre>C:\laragon\www\Pemrograman_Web_Lanjut\Pertemuan 3\PMW_POS>php artisan make:controller AdminController</pre> <p>INFO Controller [C:\laragon\www\Pemrograman_Web_Lanjut\Pertemuan 3\PMW_POS\app\Http\Controllers\AdminController.php] created successfully.</p> <p>Function index()</p> <pre>class AdminController extends Controller { public function index(){ return view('admin'); } }</pre>

8

Membuat ManagerController

```
C:\laragon\www\Pemrograman_Web_Lanjut\Pertemuan 3\PWL_POS>php artisan make:controller ManagerController
```

```
INFO Controller [C:\laragon\www\Pemrograman_Web_Lanjut\Pertemuan 3\PWL_POS\app\Http\Controllers\ManagerController.php] created successfully.
```

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

9

Hasil

