

JOB SHEET 09

AUTHENTICATION, MIDDLEWARE

Patricia Sonnya Fridayanti

NIM. 2141762060

SIB-3C / 21

No	Keterangan
<p>Praktikum – Auth dan Middleware</p> <p>Praktikum ini lanjutan dari pertemuan ke-4. Tabel yang digunakan untuk praktikum ini menggunakan m_user. Sebelum melakukan praktikum jalankan perintah di bawah ini</p> <ol style="list-style-type: none">Start Laragon atau XamppJalankan “php artisan serve”Jalankan “npm run dev”	
1.	<p>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</p> <pre>Pertemuan2 > POS > config > auth.php 3 return [62 'providers' => [63 'users' => [64 'driver' => 'eloquent', 65 'model' => App\Models\UserModel::class, 66], </pre>
2.	<p>Membuat middleware dengan perintah pada cmd atau terminal</p> <p>Php artisan make:middleware Cek_login</p> <pre>PS C:\laragon\www\Pemrograman Web Lanjut\Pertemuan2\POS> php artisan make:middleware Cek_login INFO Middleware [C:\laragon\www\Pemrograman Web Lanjut\Pertemuan2\POS\app\Http\Middleware\Cek_login.php] created successfully.</pre> <p>Setelah itu cek didalam folder app/Http/Middleware apakah sudah ada file bernama Cek_login.php</p>

	<pre> Pertemuan2 > POS > app > Http > Middleware > Cek_login.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 Cek_login 10 { 11 /** 12 * Handle an incoming request. 13 * 14 * @param \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response) \$next 15 */ 16 public function handle(Request \$request, Closure \$next): Response 17 { 18 return \$next(\$request); 19 } 20 } 21 </pre>	
3.	Kemudian tambahkan kode dibawah ini, letakkan di function handle()	
4.	<p>Ketika sudah membuat middleware, langkah berikutnya harus registrasikan pada kernel.php. Agar middleware dapat dibaca oleh sistem. Buka folder App/Http/Kernel.php dan tambahkan code pada variabel \$middlewareAliases seperti dibawah</p> <pre> 55 protected \$middlewareAliases = [56 'auth' => \App\Http\Middleware\Authenticate::class, 57 'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class, 58 'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class, 59 'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class, 60 'can' => \Illuminate\Auth\Middleware\Authorize::class, 61 'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class, 62 'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class, 63 'precognitive' => \Illuminate\Foundation\Http\Middleware\HandlePrecognitiveRequests::class, 64 'signed' => \App\Http\Middleware\ValidateSignature::class, 65 'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class, 66 'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class, 67 // tambahkan middleware alias dengan nama cek_login 68 'cek_login' => \App\Http\Middleware\Cek_login::class, 69]; </pre>	
5.	<p>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</p> <p>Php artisan make:controller AuthController</p> <p>Buka file App/Http/Controllers/AuthController.php yang sudah kita buat. Lalu kita akan isi perintah code untuk membuat proses autentikasi , verifikasi, dan logout</p>	
6.	Kemudian buat kode AuthController.php seperti dibawah ini	

Pertemuan2 > POS > app > Http > Controllers > AuthController.php > AuthController

```
16 class AuthController extends Controller
17 {
18     public function index(){
19         // kita ambil data user lalu simpan pada variabel $user
20         $user = Auth::user();
21
22         // kondisi jika usernya ada
23         if ($user) {
24             // jika usernya memiliki level admin
25             if ($user->level_id == '1') {
26                 return redirect()->intended('admin');
27             }
28             else if($user -> level_id == '2'){
29                 return redirect()->intended('manager');
30             }
31         }
32         return view('login');
33     }
34
35     public function proses_login(Request $request){
36         // kita buat validasi pada saat tombol login di klik
37         //validasinya username dan password wajib di isi
38         $request->validate([
39             'username'=>'required',
40             'password'=>'required'
41         ]);
42
43         // ambil data request username dan password
44         $credentials = $request->only('username', 'password');
45         if (Auth::attempt($credentials)) {
46
47             // kalau berhasil simpan data user di variabel $user
48             $user = Auth::user();
49
50             // cek lagi jika level usernya admin maka arahkan ke halaman admin
51             if ($user->level_id == '1') {
52                 return redirect()->intended('admin');
53             }
54
55             // cek lagi jika level usernya biasa maka arahkan ke halaman user
56
57             else if($user -> level_id == '2'){
58                 return redirect()->intended('manager');
59             }
60
61             // jika belum ada role maka ke halaman /
62             return redirect()->intended('/');
63         }
64
65         // jika ga ada data user yang valid maka kembalikan lagi ke halaman login
66         // pastikan kirim pesan error
67         return redirect('login')
68             ->withInput()
69             ->withErrors(['login gagal' => 'Pastikan kembali username dan password yang dimasukkan sudah benar']);
70
71     }
72
73     public function register(){
74         // tampilan view register
75         return view('register');
76     }
77 }
```

	<pre> 78 public function proses_register(Request \$request){ 79 // validasi semua field wajib di isi 80 // validasi username unique 81 \$validator = FacadesValidator::make(\$request->all()),[82 'nama'=> 'required', 83 'username'=>'required unique:m_user', 84 'password'=>'required' 85]); 86 87 // kalau gagal kembali ke halaman register dan muncul pesna error 88 if (\$validator ->fails()) { 89 return redirect('/regsiter') 90 ->withErrors(\$validator) 91 ->withInput(); 92 } 93 94 // kalau berhasil isi level dan hash passwordnya biar secure 95 \$request['level_id']='2'; 96 \$request['password'] = Hash::make(\$request -> password); 97 98 // masukkan semua data pada request ke tabel m_user 99 UserModel::create(\$request->all()); 100 101 // kalo berhasil arahkan ke halaman login 102 return redirect()->route('login'); 103 } 104 105 public function logout(Request \$request){ 106 //logout itu harus menghapus sessionnya 107 \$request->session()->flush(); 108 </pre>
7.	<p>Kemudian tambahkan kode pada route pada file routes/web.php seperti kode program dibawah</p> <pre> 35 Route::get('login', [AuthController::class, 'index'])->name('login'); //route untuk view login atau tampilan 36 Route::post('register', [AuthController::class, 'register'])->name('register'); //route untuk operasi login 37 Route::post('proses_login', [AuthController::class, 'proses_login'])->name('proses_login'); // route operasi register 38 Route::get('logout',[AuthController::class, 'logout'])->name('logout'); //route tampilan register 39 Route::get('proses_register', [AuthController::class, 'proses_register'])->name('proses_register'); 40 41 Route::group(['middleware'=> ['auth']], function(){ 42 Route::group(['middleware'=>['cek_login:1']], function(){ 43 Route::resource('admin', AdminController::class); 44 }); 45 Route::group(['middleware'=>['cek_login:2']], function(){ 46 Route::resource('admin', ManagerController::class); 47 }); 48 49 }); </pre>
8.	<p>Kemudian buat file controller dengan nama AdminController sebagai halaman yang boleh diakses oleh admin saja. Jalankan perintah dibawah</p> <p>Php artisan make:controller AdminController</p> <pre> PS C:\laragon\www\Pemrogramman Web Lanjut\Pertemuan2\POS> php artis an make:controller AdminController </pre> <pre> INFO Controller [C:\laragon\www\Pemrogramman Web Lanjut\Pertemu an2\POS\app\Http\Controllers\AdminController.php] created successfu lly. </pre>
9.	<p>Buka folder App/Http/Controllers dan isi file AdminController seperti dibawah</p>

	<pre> Pertemuan2 > POS > app > Http > Controllers > AdminController.php > ... 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 class AdminController extends Controller 8 { 9 public function index(){ 10 return view('admin'); 11 } 12 } </pre>
10.	<p>Kemudian buat file controller dengan nama ManagerController sebagai halaman yang boleh diakses oleh admin saja. Jalankan perintah dibawah</p> <p>php artisan make:controller ManagerController</p> <pre> PS C:\laragon\www\Pemrogramman Web Lanjut\Pertemuan2\POS> php artis an make:controller ManagerController INFO Controller [C:\laragon\www\Pemrogramman Web Lanjut\Pertemu an2\POS\app\Http\Controllers\ManagerController.php] created success fully. </pre>
11.	<p>Buka folder App/Http/Controllers dan isi file ManagerController seperti dibawah</p> <pre> Pertemuan2 > POS > app > Http > Controllers > ManagerController.php > 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 class ManagerController extends Controller 8 { 9 public function index(){ 10 return view('manager'); 11 } 12 } </pre>
12.	<p>Pada langkah terakhir yaitu membuat layout sederhana</p>
13.	<p>Buatlah halaman login dengan membuka folder resources/views dengan nama file login.blade.php. Lalu kita isi filenya seperti contoh dibawah</p>

Pertemuan2 > POS > resources > views > login.blade.php > ...

```
1 | @extends('adminlte::auth.auth-page', ['auth_type' => 'login'])
2 |
3 | @section('adminlte_css_pre')
4 | <link rel="stylesheet" href="{{ asset('vendor/icheck-bootstrap/icheck-bootstrap.min.css')}}">
5 | @stop
6 |
7 | @php( $login_url = View::getSection('login_url') ?? config('adminlte.login_url', 'login'))
8 | @php( $register_url = View::getSection('register_url') ?? config('adminlte.register_url', 'register'))
9 | @php( $password_reset_url = View::getSection('password_reset_url') ?? config('adminlte.password_reset_url', 'password/reset'))
10 |
11 | @if (config('adminlte.use_route_url', false))
12 |     @php( $login_url = $login_url ? route($login_url): '' )
13 |     @php( $register_url = $register_url ? route($register_url): '' )
14 |     @php( $password_reset_url = $password_reset_url ? route($password_reset_url): '' )
15 | @else
16 |     @php( $login_url = $login_url ? url($login_url): '' )
17 |     @php( $register_url = $register_url ? url($register_url): '' )
18 |     @php( $password_reset_url = $password_reset_url ? url($password_reset_url): '' )
19 | @endif
20 |
21 | @section('auth_header', __('adminlte::adminlte.login_message'))
22 |
23 | @section('auth_body')
24 | @error('login_gagal')
25 | <div class="alert alert-warning alert-dismissible dafé show" role="alert">
26 |     <span class="alert-inner--text"><strong>Warning</strong> {{ $message }}</span>
27 |     <button type="button" class="close" data-dismiss="alert" aria-label="Close">
28 |         <span aria-hidden="true">&times;</span></button>
29 | </div>
30 | @enderror
31 | <form action="{{url('proses login')}}" method="POST">
32 |
33 |     @csrf
34 |
35 |     {{-- email field --}}
36 |     <div class="input-group mb-3">
37 |         <input type="text" name="username" class="form-control @error('username') is-invalid @enderror"
38 |             value="{{old('username')}}" placeholder="Username" autofocus>
39 |
40 |         <div class="input-group-append">
41 |             <div class="input-group-text">
42 |                 <span class="fas fa-envelope {{config('adminlte.classes_auth_icon', '')}}"></span>
43 |             </div>
44 |         </div>
45 |
46 |         @error('username')
47 |         <span class="invalid-feedback" role="alert">
48 |             <strong>{{ $message }}</strong>
49 |         </span>
50 |         @enderror
51 |     </div>
52 |
53 |     {{-- password field --}}
54 |     <div class="input-group mb-3">
55 |         <input type="password" name="password" class="form-control @error('password') is-invalid @enderror"
56 |             value="{{old('password')}}" placeholder="{{__('adminlte::adminlte.password')}}" autofocus>
57 |
58 |         <div class="input-group-append">
59 |             <div class="input-group-text">
60 |                 <span class="fas fa-lock {{config('adminlte.classes_auth_icon', '')}}"></span>
61 |             </div>
62 |         </div>
63 |     </div>
64 |
65 |     @error("password")
66 |     <span class="invalid-feedback" role="alert">
67 |         <strong>{{ $message }}</strong>
68 |     </span>
69 |     @enderror
70 | </div>
71 |
72 |     {{-- login field --}}
73 |     <div class="row">
74 |         <div class="col-7">
75 |             <div class="icheck-primary" title="{{__('adminlte::adminlte.remember_me_hint')}}">
76 |                 <input type="checkbox" name="remember" id="remember" {{old('remember') ? 'checked' : ''}}>
77 |                 <label for="remember">
78 |                     {{__('adminlte::adminlte.remember_me')}}
79 |                 </label>
80 |             </div>
81 |         </div>
82 |         <div class="col-5">
83 |             <button type="submit" class="btn btn-block {{ config('adminlte.classes_auth_btn', 'btn-flat btn-primary')}}">
84 |                 <span class="fas fa-sign-in-alt"></span>
85 |                 {{__('adminlte::adminlte.sign_in')}}
86 |             </button>
87 |         </div>
88 |     </div>
89 | </form>
90 | @stop
91 |
92 | @section('auth_footer')
93 | @if ($register_url)
94 | <p class="my-0">
95 |     <a href="{{ route('register') }}">
96 |         {{__('adminlte::adminlte.register_a_new_membership')}}
97 |     </a>
98 | </p>
99 | @endif
100 | @stop
```

14.

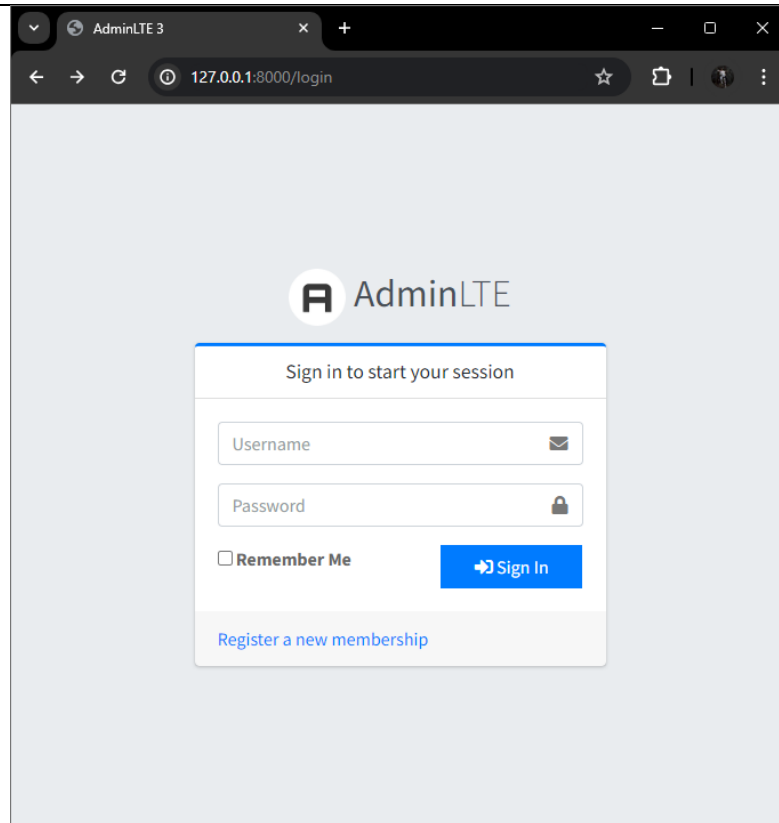
Buatlah halaman register dengan membuka folder resources/views dengan nama file register.blade.php. Lalu kita isi filenya seperti contoh dibawah

```
Pertemuan2 > POS > resources > views > register.blade.php > ...
1 | @extends('adminlte::auth.auth-page', ['auth_type' => 'register'])
2 |
3 | @php( $login_url = View::getSection('login_url') ?? config('adminlte.login_url', 'login'))
4 | @php( $register_url = View::getSection('register_url') ?? config('adminlte.register_url', 'register'))
5 |
6 | @if (config('adminlte.use_route_url', false))
7 |     @php( $login_url = $login_url ? route($login_url): '' )
8 |     @php( $register_url = $register_url ? route($register_url): '' )
9 | @else
10 |     @php( $login_url = $login_url ? url($login_url): '' )
11 |     @php( $register_url = $register_url ? url($register_url): '' )
12 | @endif
13 |
14 | @section('auth_header', __('adminlte::adminlte.register_message'))
15 |
16 | @section('auth_body')
17 |
18 | <form action="{{url('proses_register')}}" method="POST">
19 |     @csrf
20 |
21 |     {{-- nama field --}}
22 |     <div class="input-group mb-3">
23 |         <input type="text" name="nama" class="form-control @error('nama') is-invalid @enderror"
24 |             value="{{old('nama')}}" placeholder="Nama" autofocus>
25 |
26 |         <div class="input-group-append">
27 |             <div class="input-group-text">
28 |                 <span class="fas fa-user {{config('adminlte.classes_auth_icon', '')}}"></span>
29 |             </div>
30 |         </div>
31 |
32 |         @error('nama')
33 |             <span class="invalid-feedback" role="alert">
34 |                 <strong>{{ $message }}</strong>
35 |             </span>
36 |         @enderror
37 |     </div>
38 |
39 |     {{-- username field --}}
40 |     <div class="input-group mb-3">
41 |         <input type="text" name="username" class="form-control @error('username') is-invalid @enderror"
42 |             value="{{old('name')}}" placeholder="Username" autofocus>
43 |
44 |         <div class="input-group-append">
45 |             <div class="input-group-text">
46 |                 <span class="fas fa-user {{config('adminlte.classes_auth_icon', '')}}"></span>
47 |             </div>
48 |         </div>
49 |
50 |         @error('username')
51 |             <span class="invalid-feedback" role="alert">
52 |                 <strong>{{ $message }}</strong>
53 |             </span>
54 |         @enderror
55 |     </div>
56 |
57 |     {{-- password field --}}
58 |     <div class="input-group mb-3">
59 |         <input type="text" name="password" class="form-control @error('password') is-invalid @enderror"
60 |             placeholder="{{ __('adminlte::adminlte.password')}}" autofocus>
61 |
62 |         <div class="input-group-append">
63 |             <div class="input-group-text">
64 |                 <span class="fas fa-lock {{config('adminlte.classes_auth_icon', '')}}"></span>
65 |             </div>
66 |         </div>
67 |
68 |         @error('password')
69 |             <span class="invalid-feedback" role="alert">
70 |                 <strong>{{ $message }}</strong>
71 |             </span>
72 |         @enderror
73 |     </div>
74 |
75 |     {{-- register field --}}
76 |     <button type="submit" class="btn btn-block {{config('adminlte.classes_auth_btn', 'btn-flat btn-primary')}}">
77 |         <span class="fas fa-user-plus"></span>
78 |         {{ __('adminlte::adminlte.register')}}
79 |     </button>
80 |
81 | @stop
82 |
83 | @section('auth_footer')
84 | <p class="my-0">
85 |     <a href="{{ route('login')}}">
86 |         {{ __('adminlte::adminlte.i_already_have_a_membership')}}
87 |     </a>
88 | </p>
89 | @stop
```

15.

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

	<pre> Pertemuan2 > POS > resources > views > admin.blade.php > div.container 1 @extends('layout.app') 2 3 {{-- Customize layout section --}} 4 5 @section('subtitle', 'Admin') 6 @section('content_header_title', 'Home') 7 @section('content_header_subtitle', 'Admin') 8 9 @section('content') 10 <div class="container"> 11 <div class="card"> 12 <div class="card-header">Tampilan <?php echo (Auth::user()->level_id == 1)'Admin':'Manager'; ?> 13 <div class="card-body"> 14 <h1>Login Sebagai: 15 <?php echo (Auth::user()->level_id == 1)? 'Admin':'Manager'; ?></h1> 16 Logout 17 </div> 18 </div> 19 </div> 20 @endsection 21 22 @push('js') 23 24 @endpush </pre>
16.	<p>Buatlah halaman register dengan membuka folder resources/views dengan nama file manager.blade.php. Lalu kita isi filenya seperti contoh dibawah</p> <pre> Pertemuan2 > POS > resources > views > manager.blade.php > div.container > div.card > div.card-header > div.card-body > h1 4 5 @section('subtitle', 'Manager') 6 @section('content_header_title', 'Home') 7 @section('content_header_subtitle', 'Manager') 8 9 @section('content') 10 <div class="container"> 11 <div class="card"> 12 <div class="card-header">Tampilan <?php echo (Auth::user()->level_id == 1)'Admin':'Manager'; ?> 13 <div class="card-body"> 14 <h1>Login Sebagai: 15 <?php echo (Auth::user()->level_id == 1)? 'Admin':'Manager'; ?></h1> 16 Logout 17 </div> 18 </div> 19 </div> 20 @endsection 21 22 @push('js') 23 24 @endpush </pre>
17.	<p>Jalankan kode diatas dan buat laporan dengan isi screenshoot hasil dan beri penjelasan pada setiap screenshoot hasil dan commit perubahan pada git.</p> <ul style="list-style-type: none"> • Halaman login



- Halaman register

