

## **Jobsheet 09: Authentication, Middleware**

Dosen Pengampu Mata Kuliah :

M. Hasyim Ratsanjani, S.Kom., M.Kom.



Dibuat Oleh :

Daffa Yudisa Akbar : TI-2H / 10 : 2241720008

Politeknik Negeri Malang

Jurusan Teknologi Informasi

D-IV Teknik Informatika

2024

## A. Praktikum – Auth dan Middleware

No.	Langkah Praktikum
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> 60  */ 61 62  'providers' =&gt; [ 63      'users' =&gt; [ 64          'driver' =&gt; 'eloquent', 65          'model' =&gt; App\Models\UserModel::class, 66      ], 67 68      // 'users' =&gt; [ 69      //     'driver' =&gt; 'database', 70      //     'table' =&gt; 'users', 71      // ], 72  ], 73 </pre>
2.	<p>Membuat middleware dengan perintah pada cmd atau terminal</p>  <p>Setelah itu cek didalam folder app/Http/Middleware apakah sudah ada file bernama Cek_login.php</p> 
3.	<p>Kemudian tambahkan kode dibawah ini, letakkan di function handle()</p>  <pre> 1  &lt;?php 2 3  namespace App\Http\Middleware; 4 5  use Closure; 6  use Illuminate\Http\Request; 7  use Symfony\Component\HttpFoundation\Response; 8  use Illuminate\Support\Facades\Auth; 9 10 class Cek_login 11 { 12     /** 13      * Handle an incoming request. 14      * 15      * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)  \$next 16      */ 17     public function handle(Request \$request, Closure \$next, \$roles): Response 18     { 19         if (!Auth::check()){ 20             return redirect('login'); 21         } 22 23         \$user = Auth::user(); 24 25         if (\$user-&gt;level_id == \$roles){ 26             return \$next(\$request); 27         } 28 29         return redirect('login')-&gt;with('error', 'Maaf Anda Tidak Memiliki Akses'); 30     } 31 } </pre>

	<ul style="list-style-type: none"> <li>Koode program di atas memeriksa apakah pengguna telah login dan memiliki peran (role) yang sesuai untuk mengakses halaman tertentu. Jika pengguna tidak login, mereka akan diarahkan ke halaman login. Jika mereka telah login tetapi tidak memiliki peran yang sesuai, mereka akan diberikan pesan error dan diarahkan kembali ke halaman login.</li> </ul>
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> 54  // 55  protected \$middlewareAliases = [ 56      'auth' =&gt; \App\Http\Middleware\Authenticate::class, 57      'auth.basic' =&gt; \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class, 58      'auth.session' =&gt; \Illuminate\Session\Middleware\AuthenticateSession::class, 59      'cache.headers' =&gt; \Illuminate\Http\Middleware\SetCacheHeaders::class, 60      'can' =&gt; \Illuminate\Auth\Middleware\Authorize::class, 61      'guest' =&gt; \App\Http\Middleware\RedirectIfAuthenticated::class, 62      'password.confirm' =&gt; \Illuminate\Auth\Middleware\RequirePassword::class, 63      'precognitive' =&gt; \Illuminate\Foundation\Http\Middleware\HandlePrecognitiveRequests::class, 64      'signed' =&gt; \App\Http\Middleware\ValidateSignature::class, 65      'throttle' =&gt; \Illuminate\Routing\Middleware\ThrottleRequests::class, 66      'verified' =&gt; \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class, 67 68      // Menambahkan Middleware Alias dengan nama Cek_login 69      'cek_login' =&gt; \App\Http\Middleware\Cek_login::class, 70  ]; 71 </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>  <pre> PS C:\laragon\www\PWL_POS&gt; php artisan make:controller AuthController INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\AuthController.php] created successfully. PS C:\laragon\www\PWL_POS&gt; </pre> <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.	<p>Kemudian buat code AuthController.php seperti dibawah ini</p>

```

app > Http > Controllers > AuthController.php > PHP Intelephense > AuthController > proses_register
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Auth;
8  use Illuminate\Support\Facades\Hash;
9  use Illuminate\Support\Facades\Validator;
10
11  class AuthController extends Controller
12  {
13      public function index()
14      {
15          //mengambil data user dan disimpan pada variabel $user
16          $user = Auth::user();
17
18          //kondisi jika user ada
19          if ($user){
20              //jika user memiliki level admin
21              if ($user->level_id == '1'){
22                  return redirect()->intended('admin');
23              }
24
25              //jika user memiliki level manager
26              else if ($user->level_id == '2'){
27                  return redirect()->intended('manager');
28              }
29          }
30          return view('login');
31      }
32  }

```

```

app > Http > Controllers > AuthController.php > PHP Intelephense > AuthController > proses_register
11  class AuthController extends Controller
33      public function proses_login(Request $request)
34      {
35          //membuat validasi ketika tombol login diklik
36          $request->validate([
37              'username' => 'required',
38              'password' => 'required'
39          ]);
40
41          //ambil data request username dan password saja
42          $credential = $request->only('username', 'password');
43
44          //cek jika data username dan password valid
45          if (Auth::attempt($credential)){
46
47              //kalau berhasil simpan data user pada variabel $user
48              $user = Auth::user();
49
50              //cek jika level user adalah admin dan diarahkan menuju halaman admin
51              if ($user->level_id == '1'){
52                  return redirect()->intended('admin');
53              }
54
55              //cek jika level user adalah manager dan diarahkan menuju halaman user
56              else if ($user->level_id == '2'){
57                  return redirect()->intended('manager');
58              }
59
60              //jika belum ada role
61              return redirect()->intended('/');
62          }
63
64          //jika tidak ada data user yang valid maka kembali ke halaman login
65          return redirect('login')
66              ->withInput()
67              ->withErrors(['login_gagal' => 'Pastikan Kembali Username dan Password yang Dimasukkan Sudah Benar!']);
68      }

```

```

app > Http > Controllers > AuthController.php > PHP Intelephense > AuthController > proses_register
11  class AuthController extends Controller
70      public function register()
71      {
72          //tampilkan view register
73          return view('register');
74      }
75
76      public function proses_register(Request $request)
77      {
78
79          //membuat validasi ketika proses registrasi
80          $validator = Validator::make($request->all(), [
81              'name' => 'required',
82              'username' => 'required|unique:m_user',
83              'password' => 'required'
84          ]);
85
86          //jika gagal kembali ke halaman register
87          if ($validator->fails()){
88              return redirect('/register')
89                  ->withErrors($validator)
90                  ->withInput();
91          }
92
93          //kalau berhasil isi level dan hash password
94          $request['level_id'] = '2';
95          $request['password'] = Hash::make($request->password);
96
97          //memasukkan semua data ke tabel user
98          UserModel::create($request->all());
99
100          return redirect()->route('login');
101      }

```

	<pre> 102 103     public function logout(Request \$request) 104     { 105         //logout dan menghapus sessionnya 106         \$request-&gt;session()-&gt;flush(); 107 108         //jalankan juga fungsi logout pada authnya 109         Auth::logout(); 110 111         //kembali ke halaman login 112         return Redirect('login'); 113     } 114 } 115 </pre> <ul style="list-style-type: none"> <li>• Pada AuthController ini digunakan untuk menangani fungsi login, register, dan logout dari pengguna.</li> </ul>
7.	<p>Kemudian tambahkan kode pada route pada file routes/web.php seperti kode program dibawah</p> <pre> 114 115 Route::get('login', [AuthController::class, 'index'])-&gt;name('login'); 116 Route::get('register', [AuthController::class, 'register'])-&gt;name('register'); 117 Route::post('proses_login', [AuthController::class, 'proses_login'])-&gt;name('proses_login'); 118 Route::post('proses_register', [AuthController::class, 'proses_register'])-&gt;name('proses_register'); 119 Route::get('logout', [AuthController::class, 'logout'])-&gt;name('logout'); 120 121 Route::group(['middleware'=&gt;['cek_login:1']], function () { 122     Route::resource('admin', AdminController::class); 123 }); 124 Route::group(['middleware'=&gt;['cek_login:2']], function () { 125     Route::resource('manager', ManagerController::class); 126 });  </pre> <ul style="list-style-type: none"> <li>• kode program tersebut secara terperinci mengatur autentikasi pengguna, dengan rute-rute yang mengarah ke fungsi-fungsi dalam AuthController untuk proses login, registrasi, dan logout. Selain itu, pengelompokan rute berdasarkan peran (role) pengguna menggunakan middleware cek_login memastikan bahwa pengguna hanya dapat mengakses rute-rute tertentu sesuai dengan peran mereka.</li> </ul>
8.	<p>Kemudian buat file controller dengan nama AdminController sebagai halaman yang boleh diakses oleh admin saja. Jalankan perintah dibawah</p> <pre> PS C:\laragon\www\PWL_POS&gt; php artisan make:controller AdminController  INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\AdminController.php] created successfully.  PS C:\laragon\www\PWL_POS&gt; </pre>
9.	<p>Buka folder App/Http/Controllers dan isi file AdminController seperti dibawah</p> <pre> app &gt; Http &gt; Controllers &gt; AdminController.php &gt; ... 1  &lt;?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     { 11         return view('admin'); 12     } 13 } 14 </pre> <ul style="list-style-type: none"> <li>• Fungsi index di atas adalah mengarahkan menuju halaman admin</li> </ul>

10.	<p>Kemudian buat file controller dengan nama ManagerController sebagai halaman yang boleh diakses oleh admin saja. Jalankan perintah dibawah</p> <pre>PS C:\laragon\www\PWL_POS&gt; php artisan make:controller ManagerController</pre> <p>INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\ManagerController.php] created successfully.</p> <pre>PS C:\laragon\www\PWL_POS&gt;</pre>
11.	<p>Buka folder App/Http/Controllers dan isi file ManagerController seperti dibawah</p> <pre>app &gt; Http &gt; Controllers &gt; ManagerController.php &gt; PHP Intelephense &gt;</pre> <pre> 1  &lt;?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     { 11         return view('manager'); 12     } 13 } 14</pre> <ul style="list-style-type: none"> <li>Fungsi index di atas adalah mengarahkan menuju halaman manager</li> </ul>
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> <pre>resources &gt; views &gt; login.blade.php &gt; ...</pre> <pre> 1  @extends('adminlte::auth.auth-page', ['auth_type' =&gt; 'login']) 2 3  @section('adminlte_css_pre') 4      &lt;link rel="stylesheet" href="{{ asset('vendor/checkbox-bootstrap/checkbox-bootstrap.min.css') }}"&gt; 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          &lt;div class="alert alert-warning alert-dismissible fade show" role="alert"&gt; 26              &lt;span class="alert-inner--text"&gt;&lt;strong&gt;Warning!&lt;/strong&gt; {{ \$message }} &lt;/span&gt; 27              &lt;button type="button" class="close" data-dismiss="alert" aria-label="close"&gt; 28                  &lt;span aria-hidden="true"&gt;&amp;times;&lt;/span&gt; 29              &lt;/button&gt; 30          &lt;/div&gt; 31      @enderror 32      &lt;form action="{{ url('proses_login') }}" method="post"&gt; 33          @csrf 34 35          {{-- Email field --}} 36          &lt;div class="input-group mb-3"&gt; 37              &lt;input type="text" name="username" class="form-control @error('username') is-invalid @enderror" 38                  value="{{ old('username') }}" placeholder="Username" autofocus&gt; 39 40              &lt;div class="input-group-append"&gt; 41                  &lt;div class="input-group-text"&gt; 42                      &lt;span class="fas fa-envelope {{ config('adminlte.classes_auth_icon', '') }}"&gt;&lt;/span&gt; 43                  &lt;/div&gt; 44              &lt;/div&gt; 45 46              @error('username') 47                  &lt;span class="invalid-feedback" role="alert"&gt; 48                      &lt;strong&gt;{{ \$message }}&lt;/strong&gt; 49                  &lt;/span&gt; 50              @enderror 51          &lt;/div&gt; </pre>

```

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             placeholder="{{ __('adminlte::adminlte.password') }}">
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
64         @error('password')
65             <span class="invalid-feedback" role="alert">
66                 <strong>{{ $message }}</strong>
67             </span>
68         @enderror
69     </div>
70
71     {{-- Login field --}}
72     <div class="row">
73         <div class="col-7">
74             <div class="icheck-primary" title="{{ __('adminlte::adminlte.remember_me_hint') }}">
75                 <input type="checkbox" name="remember" id="remember" {{ old('remember') ? 'checked' : '' }}>
76
77                 <label for="remember">
78                     {{ __('adminlte::adminlte.remember_me') }}
79                 </label>
80             </div>
81
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     </div>
90 </form>
91 @stop

```

- Kode program di atas menyiapkan halaman login, dimana akan ada form username dan password yang disediakan pada halaman tersebut yang digunakan untuk login pengguna. Serta mempersiapkan pesan ketika terjadi error atau gagal dalam login.

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

```

1  @extends('adminlte::auth.auth-page', ['auth_type' => 'register'])
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
10 @if (config('adminlte.use_route_url', false))
11     @php( $login_url = $login_url ? route($login_url) : '' )
12     @php( $register_url = $register_url ? route($register_url) : '' )
13 else
14     @php( $login_url = $login_url ? url($login_url) : '' )
15     @php( $register_url = $register_url ? url($register_url) : '' )
16 @endif
17
18 @section('auth-header', __('adminlte::adminlte.register_message'))
19

```

```

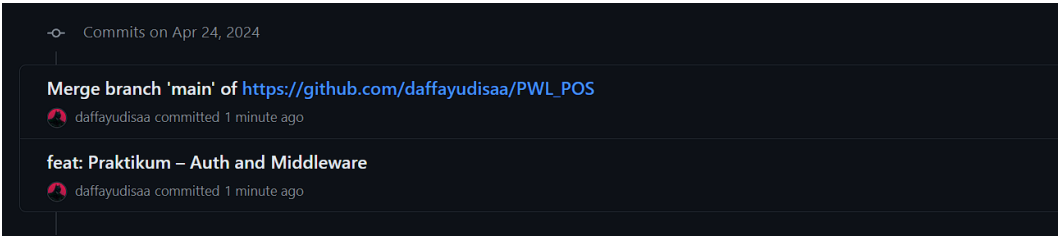
20 @section('auth_body')
21 <form action="{{ url('proses_register') }}" method="post">
22     @csrf
23
24     {{-- Nama field --}}
25     <div class="input-group mb-3">
26         <input type="text" name="name" class="form-control @error('name') is-invalid @enderror"
27             value="{{ old('name') }}" placeholder="Nama" autofocus>
28
29         <div class="input-group-append">
30             <div class="input-group-text">
31                 <span class="fas fa-user {{ config('adminlte.classes_auth_icon', '') }}"></span>
32             </div>
33         </div>
34
35         @error('name')
36             <span class="invalid-feedback" role="alert">
37                 <strong>{{ $message }}</strong>
38             </span>
39         @enderror
40     </div>
41
42     {{-- Username field --}}
43     <div class="input-group mb-3">
44         <input type="text" name="username" class="form-control @error('username') is-invalid @enderror"
45             value="{{ old('username') }}" placeholder="Username">
46
47         <div class="input-group-append">
48             <div class="input-group-text">
49                 <span class="fas fa-user {{ config('adminlte.classes_auth_icon', '') }}"></span>
50             </div>
51         </div>
52
53         @error('username')
54             <span class="invalid-feedback" role="alert">
55                 <strong>{{ $message }}</strong>
56             </span>
57         @enderror
58     </div>
59
60     {{-- Password field --}}
61     <div class="input-group mb-3">
62         <input type="password" name="password" class="form-control @error('password') is-invalid @enderror"
63             placeholder="{{ __('adminlte::adminlte.password') }}">
64
65         <div class="input-group-append">
66             <div class="input-group-text">
67                 <span class="fas fa-lock {{ config('adminlte.classes_auth_icon', '') }}"></span>
68             </div>
69         </div>
70
71         @error('password')
72             <span class="invalid-feedback" role="alert">
73                 <strong>{{ $message }}</strong>
74             </span>
75         @enderror
76     </div>
77
78     {{-- Register button --}}
79     <button type="submit" class="btn btn-block {{ config('adminlte.classes_auth_btn', 'btn-flat btn-primary') }}">
80         <span class="fas fa-user-plus"></span>
81         {{ __('adminlte::adminlte.register') }}
82     </button>
83 </form>
84 @stop
85
86 @section('auth_footer')
87 <p class="my-0">
88     <a href="{{ route('login') }}">
89         {{ __('adminlte::adminlte.i_already_have_a_membership') }}
90     </a>
91 </p>
92 @stop

```

- Kode program di atas menyiapkan halaman registrasi, dimana akan ada form untuk mengisi nama, username, dan password yang disediakan pada halaman tersebut yang digunakan untuk mendaftarkan pengguna dalam sistem. Kemudian jika register berhasil, user tersebut akan ditambahkan pada user yang ada dan akan dikembalikan menuju halaman login.

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

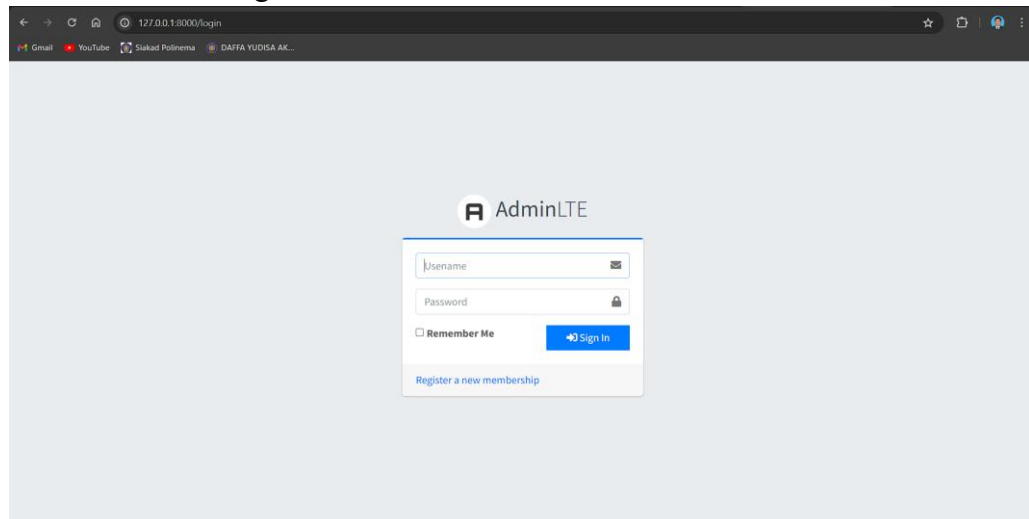


	<pre> resources &gt; views &gt; admin.blade.php &gt; ... 1 @extends('layout.app') 2 3 {{-- Customize Layout Sections --}} 4 5 @section('subtitle', 'Admin') 6 @section('content_header_title', 'Home') 7 @section('content_header_subtitle', 'Admin') 8 9 @section('content') 10 &lt;div class="container"&gt; 11   &lt;div class="card"&gt; 12     &lt;div class="card-header"&gt;Tampilan &lt;?php echo (Auth::user()-&gt;level_id == 1)?'Admin':'Manager'; ?&gt; 13     &lt;div class="card-body"&gt; 14       &lt;h1&gt;Login Sebagai: 15       &lt;?php echo (Auth::user()-&gt;level_id == 1)?'Admin':'Manager'; ?&gt; 16     &lt;/h1&gt; 17     &lt;a href="{{ route('logout') }}"&gt;Logout&lt;/a&gt; 18   &lt;/div&gt; 19 &lt;/div&gt; 20 &lt;/div&gt; 21 @endsection 22 @push('js') 23 24 @endpush </pre> <ul style="list-style-type: none"> <li>Kode program di atas menyiapkan halaman admin, dimana setelah pengguna masuk ke dalam sistem dan level_id dari user tersebut teridentifikasi sebagai admin, maka halaman admin ini akan ditampilkan.</li> </ul>
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> resources &gt; views &gt; manager.blade.php &gt; ... 1 @extends('layout.app') 2 3 {{-- Customize Layout Sections --}} 4 5 @section('subtitle', 'Manager') 6 @section('content_header_title', 'Home') 7 @section('content_header_subtitle', 'Manager') 8 9 @section('content') 10 &lt;div class="container"&gt; 11   &lt;div class="card-header"&gt;Tampilan &lt;?php echo (Auth::user()-&gt;level_id == 1)?'Admin':'Manager'; ?&gt; 12   &lt;div class="card-body"&gt; 13     &lt;h1&gt;Login Sebagai: 14     &lt;?php echo (Auth::user()-&gt;level_id == 1)?'Admin':'Manager'; ?&gt; 15   &lt;/h1&gt; 16   &lt;a href="{{ route('logout') }}"&gt;Logout&lt;/a&gt; 17 &lt;/div&gt; 18 &lt;/div&gt; 19 @endsection 20 @push('js') 21 22 @endpush </pre> <ul style="list-style-type: none"> <li>Kode program di atas menyiapkan halaman manager, dimana setelah pengguna masuk ke dalam sistem dan level_id dari user tersebut teridentifikasi sebagai manager, maka halaman manager ini akan ditampilkan.</li> </ul>
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> 

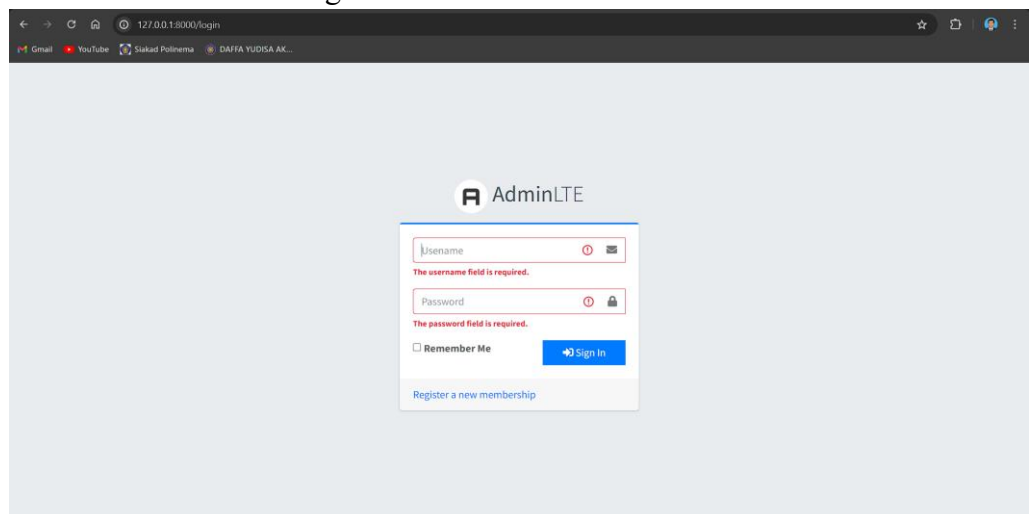
18.

Hasil:

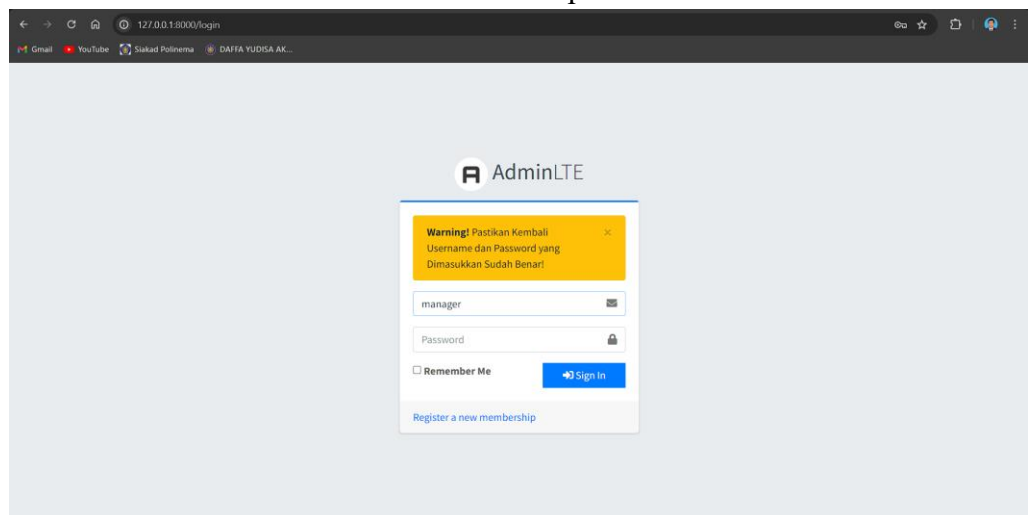
- Halaman login



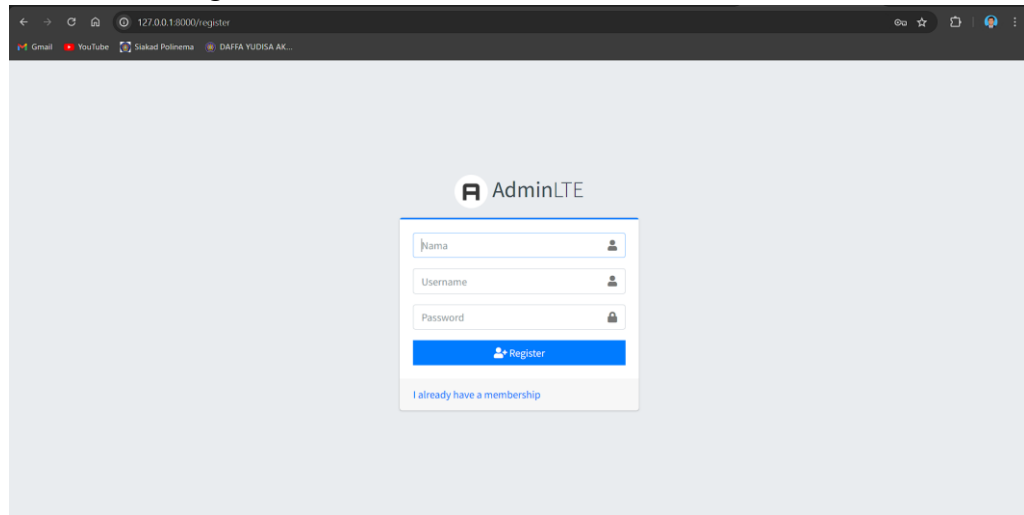
- Ketika tombol Sign In diklik namun form belum terisi



- Ketika memasukkan username atau password salah

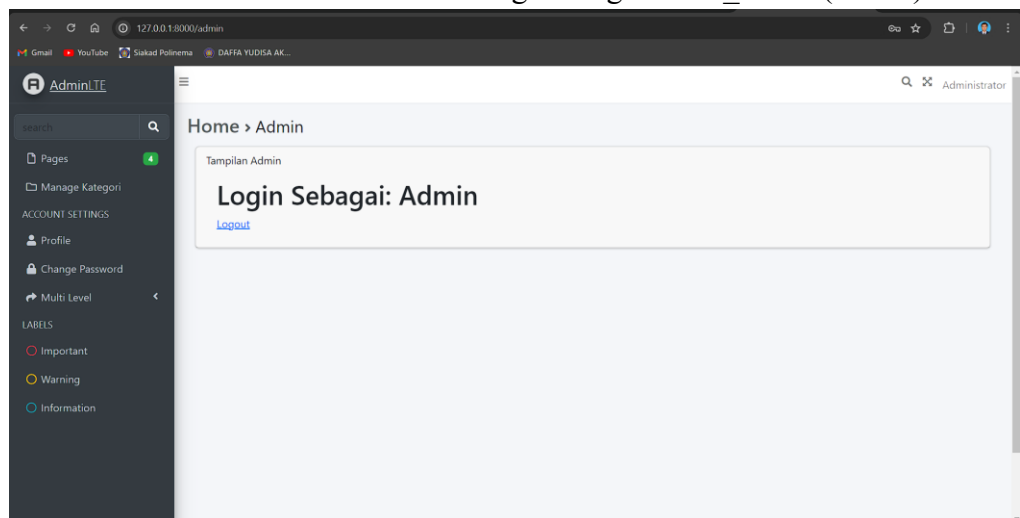


- Form register



The screenshot shows a web browser window with the URL 127.0.0.1:8000/register. The page features the AdminLTE logo at the top. Below the logo is a registration form with three input fields: 'Nama' (Name), 'Username', and 'Password'. Each field has a small user icon to its right. Below the 'Password' field is a blue 'Register' button with a user icon. At the bottom of the form, there is a link that says 'I already have a membership'.

- Halaman ketika user berhasil login dengan level\_id = 1 (admin)



- Halaman ketika user berhasil login dengan level\_id = 2 (manager)

