

PEMROGRAMAN WEB LANJUT

PERTEMUAN 9



MUHAMMAD IQBAL MAKMUR AL-MUNIRI
TI-2C / 15 / 2241720099
TEKNIK INFORMATIKA
TEKNOLOGI INFORMASI

PRAKTIKUM – (AUTH & 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.

Jawab:

```
61
62     'providers' => [
63         'users' => [
64             'driver' => 'eloquent',
65             'model' => App\Models\UserModel::class,
66         ],

```

2. Membuat middleware dengan perintah pada cmd atau terminal

Jawab:

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

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

PS D:\Polinema\Semester 4\Pemrograman Web Lanjut\pwl_pos>

```

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

Jawab:

```
10 class Cek_login
11 {
12     /**
13      * Handle an incoming request.
14      *
15      * @param \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)
16      */
17     public function handle(Request $request, Closure $next): Response
18     {
19         if (Auth::check()) {
20             return redirect('login');
21         }
22
23         $user = Auth::user();
24
25         if ($user->level_id == $roles) {
26             return $next($request);
27         }
28
29         return redirect('login')->with('error', 'Maaf anda tidak memiliki akses');
30     }
31 }
32

```

4. 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 \$middleAliases seperti di bawah.

Jawab:

```

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         'cek_login' => \App\Http\Middleware\Cek_Login::class,
68     ];

```

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`

Jawab:

```

PS D:\Polinema\Semester 4\Pemrograman Web Lanjut\pwl_pos> php artisan make:controller AuthController

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

PS D:\Polinema\Semester 4\Pemrograman Web Lanjut\pwl_pos>

```

6. Kemudian buat kode AuthController.php seperti di bawah ini.

Jawab:

```

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         $user = Auth::user();
16
17         if ($user) {
18             if ($user->level_id == '1') {
19                 return redirect()->intended('admin');
20             } else if ($user->level_id == '2') {
21                 return redirect()->intended('manager');
22             }
23         }
24         return view('login');
25     }
26
27     public function proses_login(Request $request)
28     {
29         $request->validate([
30             'username' => 'required',
31             'password' => 'required'
32         ]);
33
34         $credential = $request->only('username', 'password');
35
36         if (Auth::attempt($credential)) {
37             $user = Auth::user();
38
39             if ($user->level_id == '1') {
40                 return redirect()->intended('manager');
41             } else if ($user->level_id == '2') {
42                 return redirect()->intended('manager');
43             }
44             return redirect()->intended('/');
45         }
46         return redirect('login');
47     }

```

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

Jawab:

```
66 Route::get('/', [UserController::class, 'index']);
67 Route::get('login', [AuthController::class, 'index'])->name('login');
68 Route::post('proses_login', [AuthController::class, 'register'])->name('register');
69 Route::get('logout', [AuthController::class, 'logout'])->name('logout');
70 Route::post('proses_register', [AuthController::class, 'proses_register'])->name('proses_register');
71
72 Route::group(['middleware' => ['auth']], function () {
73     Route::group(['middleware' => ['cek_login:1']], function () {
74         Route::resource('admin', AdminController::class);
75     });
76 });
```

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

Jawab:

```
PS D:\Polinema\Semester 4\Pemrograman Web Lanjut\pwl_pos> php artisan make:controller AdminController

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

PS D:\Polinema\Semester 4\Pemrograman Web Lanjut\pwl_pos>
```

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

Jawab:

```
p M web.php M AdminController.php U X LevelModel.php M show.blade.php
app > Http > Controllers > AdminController.php > PHP Intelephense > AdminController > index
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    {
11        return view('admin');
12    }
13 }
14
```

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

Jawab:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS powershell + - - - ^ x
PS D:\Polinema\Semester 4\Pemrograman Web Lanjut\pwl_pos> php artisan make:controller ManagerController

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

PS D:\Polinema\Semester 4\Pemrograman Web Lanjut\pwl_pos>
```

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

Jawab:

```

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     {
11         return view('manager');
12     }
13 }
14

```

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

Jawab:

```
File Edit Selection View Go Run Terminal Help
loginMade.php U X
resources > views > loginMade.php > form > div.row > div.col > div.check-primary > input.member
1 @extends('adminlte.auth-page', ['auth_type' => 'login'])
2
3 @section('adminlte_css_pre')
4 <link rel="stylesheet" href="{{ asset('vendor/checkbox-bootstrap/checkbox-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 = $login_url ? route($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_error')
25 <div class="alert alert-warning alert-dismissible fade show" role="alert">
26 <span class="alert-link">{{ trans('auth.messages.forgot_password') }}</span>
27 <button type="button" class="close" data-dismiss="alert" aria-label="Close">
28 <span aria-hidden="true">{{ trans('auth.messages.close') }}</span>
29 </button>
30 </div>
31 @enderror
32 <form action="{{ url('process_login') }}" method="post">
33 <div>
34 <!-- Email Field -->
35 <div class="input-group">
36 <input type="text" name="username" class="form-control @error('username') is-invalid @enderror"
37 value="{{ old('username') }}" placeholder="Username" autofocus>
38 <div class="input-group-append">
39 <div class="input-group-text">
40 <span class="fas fa-envelope @config('adminlte.classes_auth_icon', '')"></span>
41 </div>
42 </div>
43 </div>
44 @error('username')
45 <div class="invalid-feedback" role="alert">
46 {{ trans('auth.messages.username_invalid') }}
47 </div>
48 </div>
```

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

Jawab:

```
resources > views > register.blade.php
1 @extends('adminlte::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     <form action="{{ route('process_register') }}" method="post">
18         @csrf
19
20         <!-- Name Field -->
21         <div class="input-group mb-3">
22             <input type="text" name="name" class="form-control @error('name') is-invalid @enderror"
23                 value="{{ old('name') }}" placeholder="Name" autofocus>
24
25             <div class="input-group-append">
26                 <div class="input-group-text">
27                     <span class="fas fa-user {{ config('adminlte.classes_auth_icon', '') }}"></span>
28                 </div>
29             </div>
30
31             @error('name')
32                 <span class="invalid-feedback" role="alert">
33                     <strong>{{ $message }}</strong>
34                 </span>
35             @enderror
36         </div>
37
38         <!-- Username Field -->
39         <div class="input-group mb-3">
40             <input type="text" name="username" class="form-control @error('username') is-invalid @enderror"
41                 value="{{ old('username') }}" placeholder="Username" autofocus>
42
43             <div class="input-group-append">
44                 <div class="input-group-text">
45                     <span class="fas fa-user {{ config('adminlte.classes_auth_icon', '') }}"></span>
46                 </div>
47             </div>
48         </div>
49     </form>
50 </section>
```

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

Jawab:

```
resources > views > admin.blade.php
1 @extends('layout.app')
2
3 {{-- Customize layout sections --}}
4
5 @section('subtitle', 'Admin')
6 @section('content_header_title', 'Name')
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';>
16                 <a href="{{ route('logout') }}">Logout</a>
17             </div>
18         </div>
19     </div>
20 @endsection
21
22 @push('js')
23 @endpush
24
```

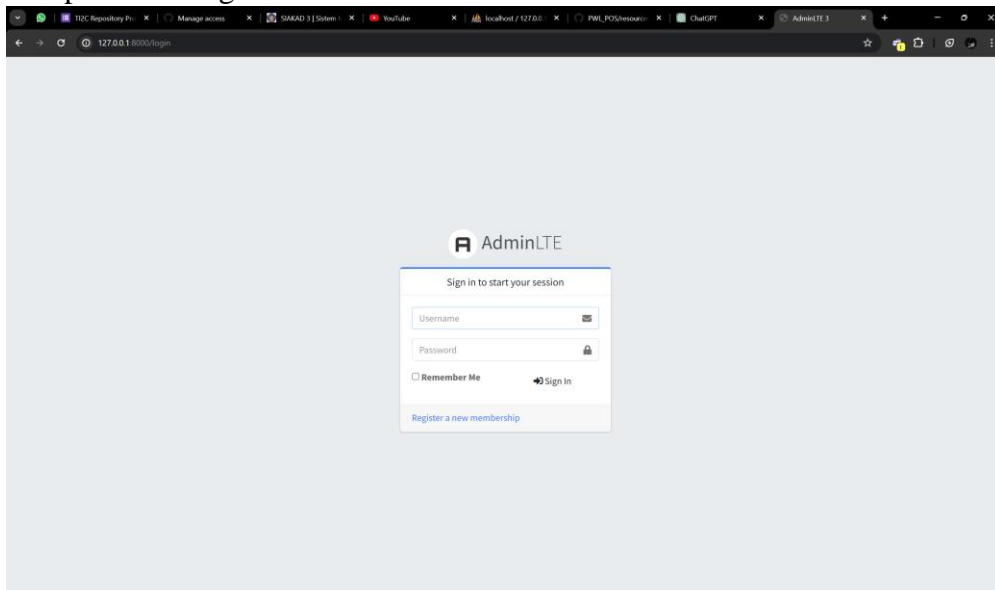
16. Buatlah halaman register dengan membuka folder resources/views dengan nama file manager.blade.php lalu kita isi filenya seperti contoh dibawah
Jawab:

```
admin.blade.php U x manager.blade.php U x
resources > views > manager.blade.php > ...
1 @extends('layout.app')
2
3 @section('subtitle', 'Manager')
4 @section('content_header_title', 'Home')
5 @section('content_header_subtitle', 'Manager')
6
7 @section('content')
8 <div class="container">
9   <div class="card">
10     <div class="card-header">Tampilan <? echo (Auth::user()->level_id == 1)?'Admin' : 'Manager'; ?>
11     <div class="card-body">
12       <h1>Login Sebagai:
13       <?php echo (Auth::user()->level_id == 1)?'Admin': 'Manager'; ?>
14     </h1>
15     <a href="{{ route('logout') }}">Logout</a>
16   </div>
17 </div>
18 </div>
19 @endsection
20
21 @push('js')
22 @endpush
23
```

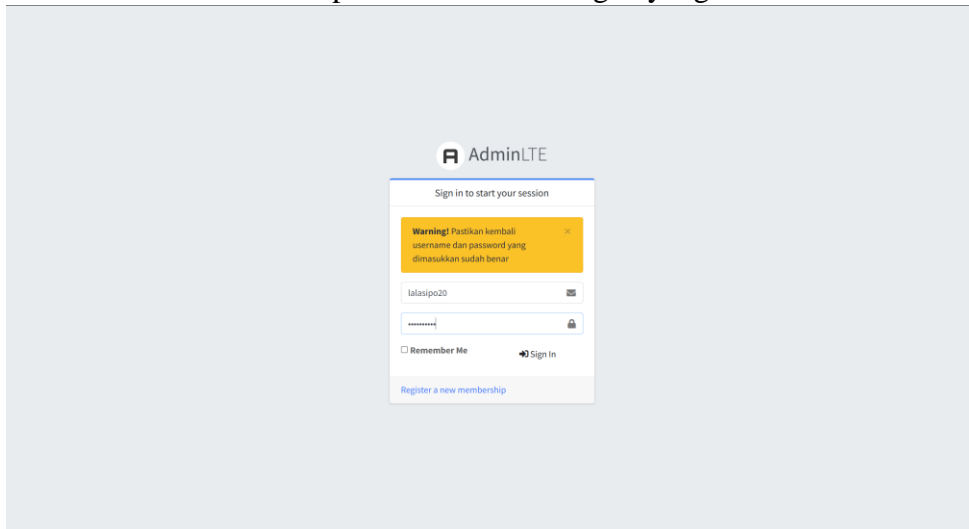
17. Jalankan kode diatas dan buat laporan dengan nusi screenshot hasil dan beri penjelasan pada setiap screenshot hasil dan commit perubahan pada git.

Jawab:

Tampilan dari login

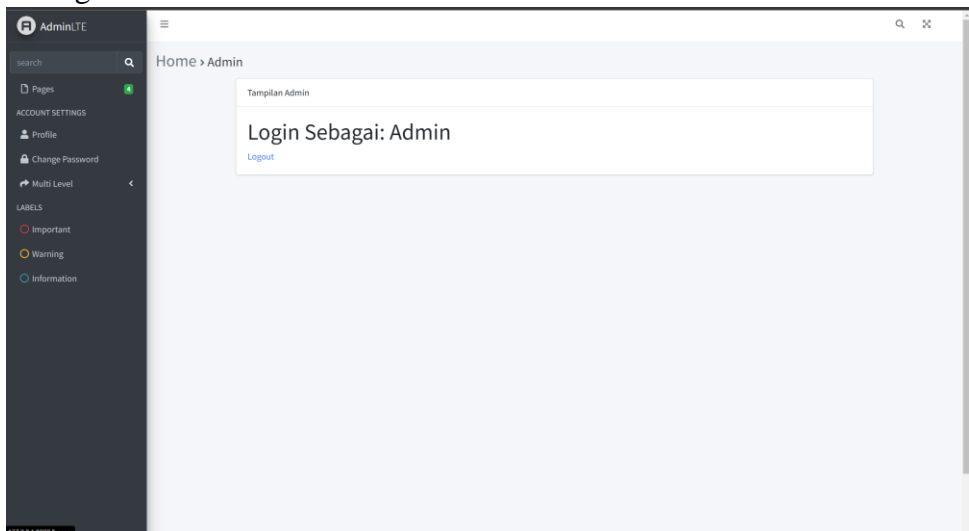


Masukkan username dan password sesuai dengan yang ada di database



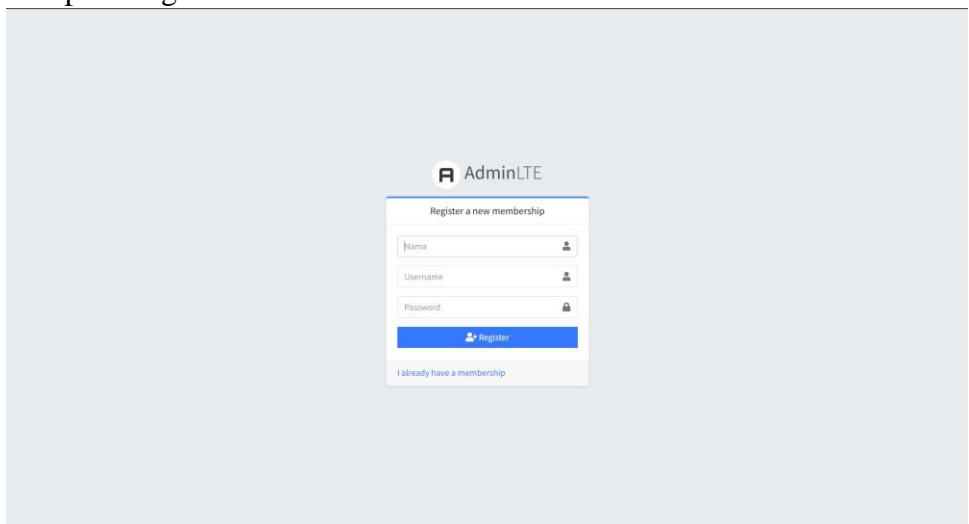
The image shows the AdminLTE login page. At the top, there is a header with the AdminLTE logo. Below the header, there is a login form titled "Sign in to start your session". The form contains a warning message: "Warning! Pastikan kembali username dan password yang dimasukan sudah benar". Below the warning, there are input fields for "Username" (containing "lalasipo20") and "Password" (containing "*****"). There is a "Remember Me" checkbox and a "Sign In" button. At the bottom of the form, there is a link "Register a new membership".

Jika username dan password sesuai maka akan diarahkan ke halaman admin atau manager



The image shows the AdminLTE admin dashboard. On the left, there is a sidebar with a search bar and a list of menu items: "Pages", "ACCOUNT SETTINGS", "Profile", "Change Password", "Multi Level", "LABELS", "Important", "Warning", and "Information". The main content area has a header "Home > Admin" and a box titled "Tampilan Admin" containing the text "Login Sebagai: Admin" and a "Logout" link.

Tampilan register



The image shows the AdminLTE register page. At the top, there is a header with the AdminLTE logo. Below the header, there is a registration form titled "Register a new membership". The form contains input fields for "Name", "Username", and "Password". There is a "Register" button and a link "I already have a membership".