



Mata Kuliah : Pemrograman Web Lanjut (PWL)
Program Studi : D4 – Sistem Informasi Bisnis
Semester : 5

Kelas : SIB
NIM : 2241760035
Nama : Fifi Novitasari
Jobsheet Ke- : 7

Laporan Jobsheet

Praktikum Ke-1 Implementasi Authentication

Langkah	Jawaban/Deskripsi
1	<p>Membuka project laravel PWL_POS kita, dan kita modifikasi konfigurasi aplikasi kita di config/auth.php</p> <p>sesuaikan dengan Model untuk tabel m_user</p> <pre>62 'providers' => [63 'users' => [64 'driver' => 'eloquent', 65 'model' => App\Models\UserModel::class, 66], 67],</pre>
2	<p>Modifikasi sedikit pada UserModel.php untuk bisa melakukan proses otentikasi</p>



	<pre>app > Models > UserModel.php > UserModel > level 1 <?php 2 3 namespace App\Models; 4 5 use Illuminate\Database\Eloquent\Factories\HasFactory; 6 use Illuminate\Database\Eloquent\Model; 7 use Illuminate\Database\Eloquent\Relations\BelongsTo; 8 use Illuminate\Foundation\Auth\User as Authenticatable; //Implementasi class Authenticatable 9 10 class UserModel extends Model 11 { 12 use HasFactory; 13 14 protected \$table = 'm_user'; //mendefinisikan nama tabel yang digunakan UserModel 15 protected \$primaryKey = 'user_id'; //mendefinisikan primary key dari tabel yang digunakan 16 protected \$fillable = ['username', 'password', 'nama', 'level_id', 'created_at', 'updated_at']; 17 18 protected \$hidden = ['password']; // jangan di tampilkan saat select 19 20 protected \$casts = ['password' => 'hashed']; // casting password agar otomatis di hash 21 22 public function level(): BelongsTo 23 { 24 return \$this->belongsTo(LevelModel::class, 'level_id', 'level_id'); 25 } 26 }</pre>
3	<p>uat <code>AuthController.php</code> untuk memproses login yang akan kita lakukan</p> <pre>UserController.php auth.php AuthController.php X login.blade.php welcome.blade.php header.blade.php web.php UserMod app > Http > Controllers > AuthController.php > AuthController > postlogin 1 <?php 2 3 namespace App\Http\Controllers; 4 use Illuminate\Http\Request; 5 use Illuminate\Support\Facades\Auth; 6 7 class AuthController extends Controller 8 { 9 public function login() 10 { 11 if(Auth::check()){ // jika sudah login, maka redirect ke halaman home 12 return redirect('/'); 13 } 14 return view('auth.login'); 15 } 16 17 public function postlogin(Request \$request) 18 { 19 if(\$request->ajax() \$request->wantsJson()){ 20 \$credentials = \$request->only('username', 'password'); 21 22 if (Auth::attempt(\$credentials)) { 23 return response()->json([24 'status' => true, 25 'message' => 'Login Berhasil', 26 'redirect' => url('/') 27]); 28 } 29 30 return response()->json([31 'status' => false, 32 'message' => 'Login Gagal' 33]); 34 } 35 return redirect('login'); 36 } 37 38 public function logout(Request \$request) 39 { 40 Auth::logout(); 41 42 \$request->session()->invalidate(); 43 \$request->session()->regenerateToken(); 44 return redirect('login'); 45 } 46 }</pre>



4

Setelah kita membuat `AuthController.php`, kita buat view untuk menampilkan halaman login. View kita buat di `auth/login.blade.php`, tampilan login bisa kita ambil dari contoh login di template **AdminLTE** seperti berikut (pada contoh login ini, kita gunakan page `login-V2` di **AdminLTE**)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Login Pengguna</title>
  <!-- Google Font: Source Sans Pro -->
  <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallbac
k">
  <!-- Font Awesome -->
  <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css')
}}">
  <!-- icheck bootstrap -->
  <link rel="stylesheet" href="{{ asset('adminlte/plugins/icheck-bootstrap/icheck-
bootstrap.min.css') }}">
  <!-- SweetAlert2 -->
  <link rel="stylesheet" href="{{ asset('adminlte/plugins/sweetalert2-theme-bootstrap-
4/bootstrap-4.min.css') }}">
  <!-- Theme style -->
  <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}">
</head>
<body class="hold-transition login-page">
<div class="login-box">
  <!-- /.login-logo -->
  <div class="card card-outline card-primary">
    <div class="card-header text-center"><a href="{{ url('/') }}"
class="h1"><b>Admin</b>LTE</a></div>
    <div class="card-body">
      <p class="login-box-msg">Sign in to start your session</p>
      <form action="{{ url('login') }}" method="POST" id="form-login">
        @csrf
        <div class="input-group mb-3">
          <input type="text" id="username" name="username" class="form-control"
placeholder="Username">
          <div class="input-group-append">
            <div class="input-group-text">
              <span class="fas fa-envelope"></span>
            </div>
          </div>
        </div>
      </form>
    </div>
  </div>
</div>
```



```
<small id="error-username" class="error-text text-danger"></small>
</div>
<div class="input-group mb-3">
  <input type="password" id="password" name="password" class="form-
control" placeholder="Password">
  <div class="input-group-append">
    <div class="input-group-text">
      <span class="fas fa-lock"></span>
    </div>
  </div>
  <small id="error-password" class="error-text text-danger"></small>
</div>
<div class="row">
  <div class="col-8">
    <div class="icheck-primary">
      <input type="checkbox" id="remember"><label
for="remember">Remember Me</label>
    </div>
  </div>
  <!-- /.col -->
  <div class="col-4">
    <button type="submit" class="btn btn-primary btn-block">Sign
In</button>
  </div>
  <!-- /.col -->
</div>
<br>
<p>Don't have an account? <a href="{{ url('signup') }}">Register</a></p>
</form>
</div>
<!-- /.card-body -->
</div>
<!-- /.card -->
</div>
<!-- /.login-box -->
<!-- jQuery -->
<script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}"></script>
<!-- Bootstrap 4 -->
<script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
<!-- jquery-validation -->
<script src="{{ asset('adminlte/plugins/jquery-validation/jquery.validate.min.js')
}}"></script>
<script src="{{ asset('adminlte/plugins/jquery-validation/additional-methods.min.js')
}}"></script>
<!-- SweetAlert2 -->
```



```
<script src="{{ asset('adminlte/plugins/sweetalert2/sweetalert2.min.js') }}"></script>
<!-- AdminLTE App -->
<script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}"></script>

<script>
    $.ajaxSetup({
        headers: {
            'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
        }
    });

    $(document).ready(function() {
        $("#form-login").validate({
            rules: {
                username: {required: true, minlength: 4, maxlength: 20},
                password: {required: true, minlength: 5, maxlength: 20}
            },
            submitHandler: function(form) { // ketika valid, maka bagian yg akan dijalankan
                $.ajax({
                    url: form.action,
                    type: form.method,
                    data: $(form).serialize(),
                    success: function(response) {
                        if(response.status){ // jika sukses
                            Swal.fire({
                                icon: 'success',
                                title: 'Berhasil',
                                text: response.message,
                            }).then(function() {
                                window.location = response.redirect;
                            });
                        }else{ // jika error
                            $('.error-text').text('');
                            $.each(response.msgField, function(prefix, val) {
                                $('#error-'+prefix).text(val[0]);
                            });
                            Swal.fire({
                                icon: 'error',
                                title: 'Terjadi Kesalahan',
                                text: response.message
                            });
                        }
                    }
                });
            }
        });
        return false;
    }
}
```



	<pre> }, errorElement: 'span', errorPlacement: function (error, element) { error.addClass('invalid-feedback'); element.closest('.input-group').append(error); }, highlight: function (element, errorClass, validClass) { \$(element).addClass('is-invalid'); }, unhighlight: function (element, errorClass, validClass) { \$(element).removeClass('is-invalid'); } }); }); </script> </body> </html></pre>
5	<p>Modifikasi <code>route/web.php</code> agar semua route masuk dalam auth</p> <pre>routes > web.php > ... 1 <?php 2 3 use App\Http\Controllers\AuthController; 4 use App\Http\Controllers\BarangController; 5 use App\Http\Controllers\KategoriController; 6 use App\Http\Controllers\LevelController; 7 use App\Http\Controllers\SupplierController; 8 use App\Http\Controllers\UserController; 9 use App\Http\Controllers\WelcomeController; 10 use Illuminate\Support\Facades\Route; 11 use Illuminate\Auth\Events\Logout; 12 13 /* 14 ----- 15 Web Routes 16 ----- 17 18 Here is where you can register web routes for your application. These 19 routes are loaded by the RouteServiceProvider and all of them will 20 be assigned to the "web" middleware group. Make something great! 21 22 */ 23 24 25 Route::pattern('id', '[0-9]+'); // artinya ketika ada parameter (id), maka harus berupa angka 26 27 Route::get('login', [AuthController::class, 'login'])->name('login'); 28 Route::post('login', [AuthController::class, 'postlogin']); 29 Route::get('logout', [AuthController::class, 'logout'])->name('logout')->middleware('auth'); 30 31 32 Route::get('register', [AuthController::class, 'register']); 33 Route::post('register', [AuthController::class, 'postRegister']);</pre>
6	<p>Coba mengakses halaman <code>localhost/PWL_POS/public</code> maka akan tampil halaman awal untuk login ke aplikasi</p>



AdminLTE

Sign in to start your session

admin

☒ Remember Me

Don't have an account? [Register](#)

AdminLTE

Sign in to start your session

admin

☒ Remember Me

Don't have an account? [Register](#)

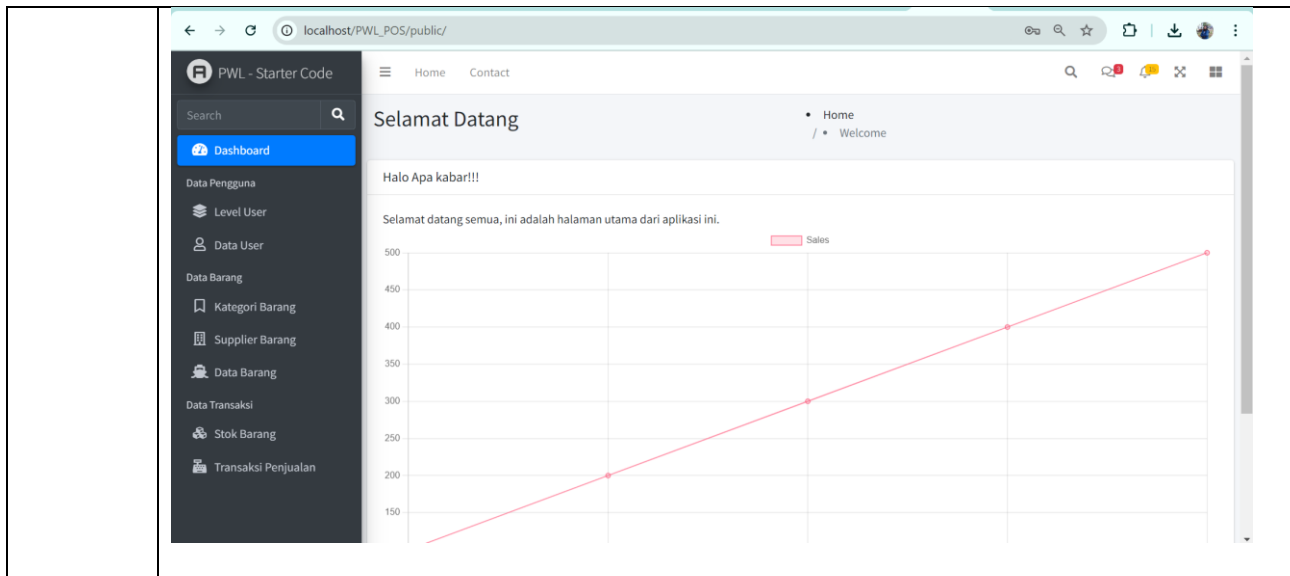
☒ Remember Me

Don't have an account? [Register](#)

Berhasil

Login Berhasil

Klik OK, maka akan masuk ke halaman dashboard



Tugas 1 – Implementasi Authentication

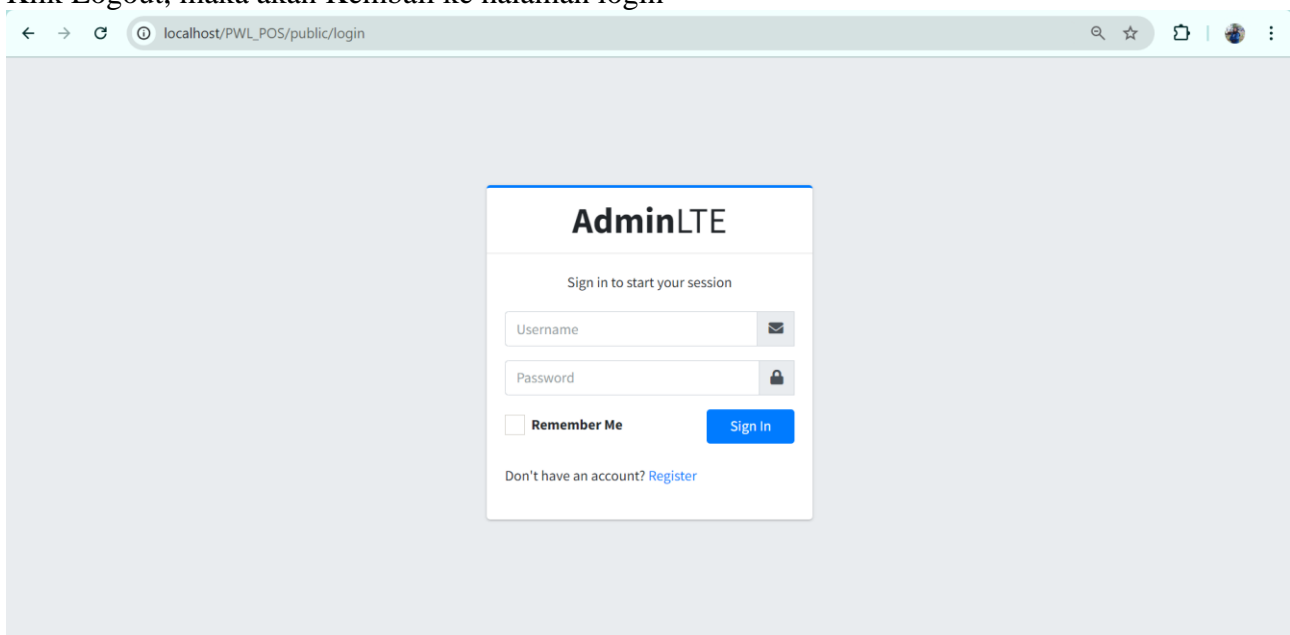
1. Silahkan implementasikan proses login pada project kalian masing-masing
2. Silahkan implementasi proses logout pada halaman web yang kalian buat

Jawab:

- Memodifikasi file header.blade.php
- pilih resources/views/layouts/header.blade.php
- Logout muncul



Klik Logout, maka akan Kembali ke halaman login

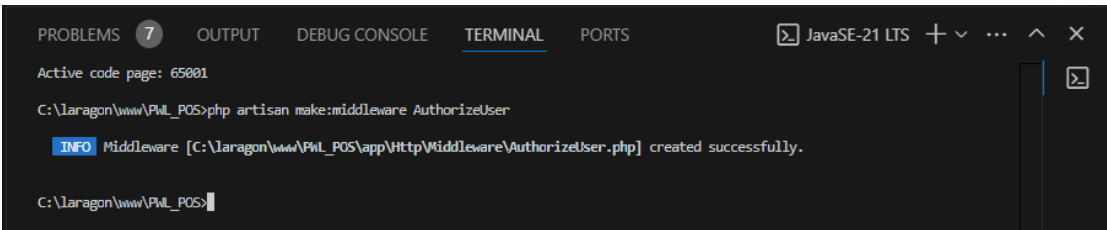


3. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan



4. Submit kode untuk implementasi Authentication pada repository github kalian.

Praktikum Ke-2

Langkah	Jawaban/Deskripsi
1	<p>Modifikasi <code>UserModel.php</code> dengan menambahkan kode berikut</p> <pre>21 //Relasi tabel m_user ke m_level (many-to-one) 22 public function level(): BelongsTo 23 { 24 return \$this->belongsTo(LevelModel::class, 'level_id', 'level_id'); 25 } 26 27 //Mendapatkan nama role 28 public function getRoleName(): string 29 { 30 return \$this->level->level_nama; 31 } 32 33 //Memeriksa bila user memiliki role tertentu 34 public function hasRole(\$role): bool 35 { 36 return \$this->level->level_kode == \$role; 37 } 38 39 //Mendapatkan kode role 40 public function getRole() 41 { 42 return \$this->level->level_kode; 43 } 44 }</pre>
2	<p>Membuat <i>middleware</i> dengan nama <code>AuthorizeUser.php</code>. Kita bisa buat <i>middleware</i> dengan mengetikkan perintah pada terminal/CMD. File <i>middleware</i> akan dibuat di <code>app/Http/Middleware/AuthorizeUser.php</code></p> 
3	<p>Edit <i>middleware</i> <code>AuthorizeUser.php</code> untuk bisa mengecek apakah pengguna yang mengakses memiliki Level/Role/Group yang sesuai</p>



	<pre>app > Http > Middleware > AuthorizeUser.php > AuthorizeUser > handle 9 ss AuthorizeUser 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); //ambil data user yang login 19 //fungsi user() diambil dari Usermodel.php 20 if(\$user->\$hasRole(\$role)){ //cek apakah user punya role yang diinginkan 21 return \$next(\$request); 22 } 23 // jika tidak punya role, maka ditampilkam error 403 24 abort(403, 'Forbidden. Kamu tidak punya akses ke halaman ini'); 25 } 26</pre>																																																					
4	<p>Daftarkan ke <code>app/Http/Kernel.php</code> untuk <i>middleware</i> yang kita buat barusan</p> <pre>54 protected \$middlewareAliases = [55 'auth' => \App\Http\Middleware\Authenticate::class, 56 'authorize' => \App\Http\Middleware\AuthorizeUser::class, //middleware yang kita buat 57 'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class, 58 'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,</pre>																																																					
5	<p>Perhatikan tabel <code>m_level</code> yang menjadi tabel untuk menyimpan level/group/role dari user ada</p> <table><thead><tr><th></th><th></th><th></th><th>level_id</th><th>level_kode</th><th>level_nama</th><th>created_at</th><th>updated_at</th></tr></thead><tbody><tr><td><input type="checkbox"/></td><td>Ubah</td><td>Salin</td><td>Hapus</td><td>1</td><td>ADM</td><td>Administrator</td><td>NULL</td><td>NULL</td></tr><tr><td><input type="checkbox"/></td><td>Ubah</td><td>Salin</td><td>Hapus</td><td>2</td><td>MNG</td><td>Manager</td><td>NULL</td><td>NULL</td></tr><tr><td><input type="checkbox"/></td><td>Ubah</td><td>Salin</td><td>Hapus</td><td>3</td><td>STF</td><td>Staff/Kasirr</td><td>NULL</td><td>2024-10-04 04:32:57</td></tr><tr><td><input type="checkbox"/></td><td>Ubah</td><td>Salin</td><td>Hapus</td><td>4</td><td>CST</td><td>Customer</td><td>NULL</td><td>NULL</td></tr><tr><td><input type="checkbox"/></td><td>Ubah</td><td>Salin</td><td>Hapus</td><td>6</td><td>Manager2</td><td>Mitra Wonosantri</td><td>2024-10-04 03:09:34</td><td>2024-10-04 03:09:34</td></tr></tbody></table>				level_id	level_kode	level_nama	created_at	updated_at	<input type="checkbox"/>	Ubah	Salin	Hapus	1	ADM	Administrator	NULL	NULL	<input type="checkbox"/>	Ubah	Salin	Hapus	2	MNG	Manager	NULL	NULL	<input type="checkbox"/>	Ubah	Salin	Hapus	3	STF	Staff/Kasirr	NULL	2024-10-04 04:32:57	<input type="checkbox"/>	Ubah	Salin	Hapus	4	CST	Customer	NULL	NULL	<input type="checkbox"/>	Ubah	Salin	Hapus	6	Manager2	Mitra Wonosantri	2024-10-04 03:09:34	2024-10-04 03:09:34
			level_id	level_kode	level_nama	created_at	updated_at																																															
<input type="checkbox"/>	Ubah	Salin	Hapus	1	ADM	Administrator	NULL	NULL																																														
<input type="checkbox"/>	Ubah	Salin	Hapus	2	MNG	Manager	NULL	NULL																																														
<input type="checkbox"/>	Ubah	Salin	Hapus	3	STF	Staff/Kasirr	NULL	2024-10-04 04:32:57																																														
<input type="checkbox"/>	Ubah	Salin	Hapus	4	CST	Customer	NULL	NULL																																														
<input type="checkbox"/>	Ubah	Salin	Hapus	6	Manager2	Mitra Wonosantri	2024-10-04 03:09:34	2024-10-04 03:09:34																																														
6	<p>Untuk mencoba <i>authorization</i> yang telah kita buat, maka perlu kita modifiikasi <code>route/web.php</code> untuk menentukan route mana saja yang akan diberi hak akses sesuai dengan level user</p>																																																					



7	Uji coba menggunakan akun selain level administrator dan akses route menu level

Tugas 2 – Implementasi Authoriization

1. Apa yang kalian pahami pada praktikum 2 ini?
2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
3. Submit kode untuk impementasi Authorization pada repository github kalian.

Praktikum Ke-3 Implementasi Multi Level Authorization di Lawavel dengan Middleware

Langkah	Jawaban/Deskripsi
1	<p>Modifikasi <code>UserModel.php</code> untuk mendapatkan <code>level_kode</code> dari user yang sudah login. Jadi kita buat fungsi dengan nama <code>getRole()</code></p> <pre>27 //Mendapatkan nama role 28 public function getRoleName(): string 29 { 30 return \$this->level->level_nama; 31 } 32 33 //Memeriksa bila user memiliki role tertentu 34 public function hasRole(\$role): bool 35 { 36 return \$this->level->level_kode == \$role; 37 } 38 39 //Mendapatkan kode role 40 public function getRole() 41 { 42 return \$this->level->level_kode; 43 } 44 }</pre>
2	Selanjutnya, modifikasi middleware <code>AuthorizeUser.php</code> dengan kode berikut



	<pre>app > Http > Middleware > AuthorizeUser.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 AuthorizeUser 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, ... \$roles): Response 17 { 18 // return \$next(\$request); //ambil data user yang login 19 // //fungsi user() diambil dari Usermodel.php 20 // if(\$user->hasRole(\$role)){ //cek apakah user punya role yang diinginkan 21 22 //Praktiku 3 JS7 23 \$user_role = \$request->user()->getRole(); //ambil data level_kode dari user yang login 24 if(in_array(\$user_role, \$roles)){ //cek apakah level_kode user ada didalam array roles 25 return \$next(\$request); //jika ada, maka lanjutkan request 26 } 27 // jika tidak punya role, maka ditampilkam error 403 28 abort(403, 'Forbidden. Kamu tidak punya akses ke halaman ini'); 29 } 30 }</pre>
3	<p>Setelah itu tinggal kita perbaiki route/web.php sesuaikan dengan role/level yang diinginkan.</p> <pre>108 //Semua route di grup ini harus punya role ADM (Administrator) dan MNG (Manager) 109 Route::group(['prefix' => 'barang', 'middleware'=> 'authorize:ADM,MNG'], function(){ 110 Route::get('/', [BarangController::class, 'index']); //menampilkan laman awal barang 111 Route::post('/list', [BarangController::class, 'list']); //menampilkan data barang dalam bentuk json untuk datatables 112 Route::get('/create_ajax', [BarangController::class, 'create_ajax']); //menampilkan laman form tambah barang AJAX 113 Route::post('/ajax', [BarangController::class, 'store_ajax']); //menyimpan data barang baru AJAX 114 Route::get('/{id}/edit_ajax', [BarangController::class, 'edit_ajax']); //menampilkan laman form edit barang AJAX 115 Route::put('/{id}/update_ajax', [BarangController::class, 'update_ajax']); //menyimpan perubahan data barang AJAX 116 Route::get('/{id}/delete_ajax', [BarangController::class, 'confirm_ajax']); //menampilkan form confirm hapus data barang AJAX 117 Route::delete('/{id}/delete_ajax', [BarangController::class, 'delete_ajax']); //menghapus data barang AJAX 118 }); 119 });</pre>
4	<p>Sekarang kita sudah bisa memberikan hak akses menu/route ke beberapa level user</p>

Praktikum 4 Implementasi Form Registrasi

Langkah	Jawaban/Deskripsi
1	Membuat file RegistrasiController.php di menu app/http/Controllers



```
web.php  RegistrasiController.php  signup.blade.php
app > Http > Controllers > RegistrasiController.php > RegistrasiController
11 class RegistrasiController extends Controller
12 {
13     // Menampilkan halaman form tambah user
14     public function registration()
15     {
16         $level = LevelModel::select('level_id', 'level_nama')->get();
17
18         return view('auth.signup',
19             ['level' => $level]);
20     }
21
22     // Menyimpan data user baru
23     public function store(Request $request)
24     {
25         $validator = Validator::make($request->all(), [
26             'username' => 'required|string|min:3|unique:m_user,username',
27             'nama' => 'required|string|max:100', //nama harus diisi, berupa string, dan maksimal 100 karakter
28             'password' => 'required|min:5', //password harus diisi dan minimal 5 karakter
29             'level_id' => 'required|integer' //level_id harus diisi dan berupa angka
30         ]);
31
32         if ($validator->fails()) {
33             // Jika validasi gagal, kirim respons JSON dengan pesan error
34             return response()->json([
35                 'status' => false,
36                 'message' => 'Validasi gagal.',
37                 'msgfield' => $validator->messages()
38             ], 422);
39         }
40
41         UserModel::create([
42             'username' => $request->username,
43             'nama' => $request->nama,
44             'password' => bcrypt($request->password), //password dienkripsi sebelum disimpan
45             'level_id' => $request->level_id
46         ]);
47
48         // Kirim respons sukses
49         return response()->json([
50             'status' => true,
51             'message' => 'Registrasi berhasil dilakukan!',
52             'redirect' => url('/login') // Redirect ke laman login
53         ]);
54     }
55 }
56 }
```

3

Menambahkan dan membuat file signup.blade.php pada resources/views/auth

```
web.php  RegistrasiController.php  signup.blade.php
resources > Views > auth > signup.blade.php > HTML
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>Sign Up</title>
7     <!-- Google Font: Source Sans Pro -->
8     <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=block">
9     <!-- Font Awesome -->
10    <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}">
11    <!-- iCheck bootstrap -->
12    <link rel="stylesheet" href="{{ asset('adminlte/plugins/icheck-bootstrap/icheck-bootstrap.min.css') }}">
13    <!-- SweetAlert2 -->
14    <link rel="stylesheet" href="{{ asset('adminlte/plugins/sweetalert2-theme-bootstrap-4/bootstrap-4.min.css') }}">
15    <!-- Theme style -->
16    <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}">
17 </head>
18 <body class="hold-transition login-page">
19 <div class="login-box">
20     <div class="card card-outline card-primary">
21         <div class="card-header text-center"><a href="{{ url('/') }}" class="h1">bAdmin</a></div>
22         <div class="card-body">
23             <p class="login-box-msg">Sign up to create your account</p>
24             <form action="{{ url('signup') }}" method="POST" id="form-tambah">
25                 @csrf
26                 <div class="form-group">
27                     <label level_id="level_id" name="level_id" required>
28                         <select class="form-control" id="level_id" name="level_id" required>
29                             <option value="">Pilih Level </option>
30                             @foreach($level as $item)
31                                 <option value="{{ $item->level_id }}">{{ $item->level_nama }}</option>
32                             @endforeach
33                         </select>
34                         <small id="error-level_id" class="error-text form-text text-danger"></small>
35                     </div>
36                     <div class="form-group">
37                         <label username="username" name="username" required>
38                             <input type="text" class="form-control" id="username" name="username" value="{{ old('username') }}" required>
39                             <small id="error-username" class="error-text form-text text-danger"></small>
40                         </div>
41                     <div class="form-group">
42                         <label nama="nama" name="nama" required>
43                             <input type="text" class="form-control" id="nama" name="nama" value="{{ old('nama') }}" required>
44                             <small id="error-nama" class="error-text form-text text-danger"></small>
45                         </div>
46                     <div class="form-group">
47                         <input type="submit" value="Sign Up" class="btn btn-primary">
48                     </div>
49                 </form>
50             </div>
51         </div>
52     </div>
53 </div>
```



	<pre>53 <div class="form-group"> 54 <label>Password</label> 55 <div class="col-11"> 56 <input type="password" class="form-control" id="password" name="password" required> 57 <small id="error-password" class="error-text form-text text-danger"></small> 58 </div> 59 </div> 60 <div class="row"> 61 <div class="col-8"> 62 <div class="checkbox-check-primary"> 63 Kembali 64 </div> 65 </div> 66 <!-- /.col --> 67 <div class="col-4"> 68 <button type="submit" class="btn btn-primary btn-block">Sign Up</button> 69 </div> 70 <!-- /.col --> 71 </div> 72 </form> 73 <!-- jQuery --> 74 <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}"></script> 75 <!-- Bootstrap 4 --> 76 <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script> 77 <!-- jQuery validation --> 78 <script src="{{ asset('adminlte/plugins/jquery-validation/jquery.validate.min.js') }}"></script> 79 <script src="{{ asset('adminlte/plugins/jquery-validation/additional-methods.min.js') }}"></script> 80 <!-- SweetAlert2 --> 81 <script src="{{ asset('adminlte/plugins/sweetalert2/sweetalert2.min.js') }}"></script> 82 <!-- AdminLTE App --> 83 <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}"></script> 84 <script> 85 \$.ajaxSetup({ 86 headers: { 87 'X-CSRF-TOKEN': \$('meta[name="csrf-token"]').attr('content') 88 } 89 }); 90</pre> <pre>91 \$(document).ready(function() { 92 \$("#form-tambah").validate({ 93 rules: { 94 level_id: {required: true, number: true}, 95 username: {required: true, minlength: 3, maxlength: 20}, 96 nama: {required: true, minlength: 3, maxlength: 100}, 97 password: {required: true, minlength: 5, maxlength: 20} 98 }, 99 submitHandler: function(form) { 100 \$.ajax({ 101 url: form.action, 102 type: form.method, 103 data: \$(form).serialize(), 104 success: function(response) { 105 if(response.status){ 106 Swal.fire({ 107 icon: 'success', 108 title: 'Berhasil', 109 text: response.message 110 }).then(function() { 111 window.location = response.redirect; 112 }); 113 }else{ 114 \$('.error-text').text(''); 115 \$.each(response.msgField, function(prefix, val) { 116 \$('#error-'+prefix).text(val[0]); 117 }); 118 Swal.fire({ 119 icon: 'error', 120 title: 'Terjadi Kesalahan', 121 text: response.message 122 }); 123 } 124 } 125 }); 126 return false; 127</pre> <pre>128 }, 129 errorElement: 'span', 130 errorPlacement: function (error, element) { 131 error.addClass('invalid-feedback'); 132 element.closest('.form-group').append(error); 133 }, 134 highlight: function (element, errorClass, validClass) { 135 \$(element).addClass('is-invalid'); 136 }, 137 unhighlight: function (element, errorClass, validClass) { 138 \$(element).removeClass('is-invalid'); 139 } 140 }); 141 </script> 142 </div> 143 <!-- /.card-body --> 144 </div> 145 <!-- /.card --> 146 </div> 147 <!-- /.login-box --> 148 149 </body> 150 </html></pre>
4	Memodifikasi file web.php di routes dengan menambahkan kode registrasi yang sudah kita buat



	<pre>use App\Http\Controllers\RegistrasiController; Route::get('signup', [RegistrasiController::class, 'registration'])->name('signup'); Route::post('signup', [RegistrasiController::class, 'store']);</pre>
5	<p>Berikut hasil implementasi form untuk registrasi user.</p> 