



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

Kelas : SIB  
NIM : 2241760030  
Nama : Fardiyani Afro'ul Yasinta  
Jobsheet Ke- : 7  
Link Github : [https://github.com/Fafayasinta/PWL\\_POS.git](https://github.com/Fafayasinta/PWL_POS.git)

## Laporan Jobsheet

### Praktikum Ke-1 | Implementasi Authentication

Langkah	Jawaban/Deskripsi
1.	<p>Kita buka project laravel <b>PWL_POS</b> kita, dan kita modifikasi konfigurasi aplikasi kita di <b>config/auth.php</b></p> <p>Pada bagian ini kita sesuaikan dengan Model untuk tabel <b>m_user</b> yang sudah kita buat</p> <pre>52 providers' =&gt; [ 53     'users' =&gt; [ 54         'driver' =&gt; 'eloquent', 55         'model' =&gt; App\Models\UserModel::class, 56     ],</pre>
2.	Selanjutnya kita modifikasi sedikit pada <b>UserModel.php</b> untuk bisa melakukan proses otentikasi



	<pre>app &gt; Models &gt; UserModel.php &gt; PHP Intelephense &gt; UserModel &gt; level  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 14 references   0 implementations 10 class UserModel extends Authenticatable 11 { 12     use HasFactory; 13     0 references 14     protected \$table = 'm_user'; // Mendefinisikan nama tabel yang digunakan oleh model ini 15     0 references 16     protected \$primaryKey = 'user_id'; // Mendefinisikan primary key dari tabel yang digunakan 17     0 references 18     protected \$fillable = ['username','password','nama','level_id','created_at', 'updated_at']; 19     0 references 20     protected \$hidden = ['password']; // jangan di tampilkan select 21     0 references 22     protected \$casts = ['password' =&gt; 'hashed']; // casting password agar otomatis di hash 23 24     0 references   0 overrides 25     public function level(): BelongsTo { 26         return \$this-&gt;belongsTo(related: LevelModel::class, foreignKey: 'level_id'); 27     } 28 }</pre>
3.	Selanjutnya kita buat <code>AuthController.php</code> untuk memproses login yang akan kita lakukan



	<pre>1  &lt;?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-&gt;ajax()    \$request-&gt;wantsJson()){ 20             \$credentials = \$request-&gt;only('username', 'password'); 21 22             if (Auth::attempt(\$credentials)) { 23                 return response()-&gt;json([ 24                     'status' =&gt; true, 25                     'message' =&gt; 'Login Berhasil', 26                     'redirect' =&gt; url('/') 27                 ]); 28             } 29 30             return response()-&gt;json([ 31                 'status' =&gt; false, 32                 'message' =&gt; 'Login Gagal' 33             ]); 34         } 35         return redirect('login'); 36     } 37 38     public function logout(Request \$request) 39     { 40         Auth::logout(); 41 42         \$request-&gt;session()-&gt;invalidate(); 43         \$request-&gt;session()-&gt;regenerateToken(); 44         return redirect('login'); 45     } 46 }</pre>
4.	Setelah kita membuat <code>AuthController.php</code> , kita buat view untuk menampilkan halaman login. View kita buat di <code>auth/login.blade.php</code> , tampilan login bisa kita ambil dari contoh login di template <b>AdminLTE</b> seperti berikut (pada contoh login ini, kita gunakan page <code>login-V2</code> di <b>AdminLTE</b> )



```
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>Login Pengguna</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=fallback">
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   <!-- /.login-logo -->
21   <div class="card card-outline card-primary">
22     <div class="card-header text-center"><a href="{{ url('/') }}" class="h1"><b>Admin</b>LTE</a></div>
23     <div class="card-body">
24       <p class="login-box-msg">Sign in to start your session</p>
25       <form action="{{ url('login') }}" method="POST" id="form-login">
26         @csrf
27         <div class="input-group mb-3">
28           <input type="text" id="username" name="username" class="form-control" placeholder="Username">
29           <div class="input-group-append">
30             <div class="input-group-text">
31               <span class="fas fa-envelope"></span>
32             </div>
33           </div>
34           <small id="error-username" class="error-text text-danger"></small>
35         </div>
36         <div class="input-group mb-3">
37           <input type="password" id="password" name="password" class="form-control" placeholder="Password">
38           <div class="input-group-append">
39             <div class="input-group-text">
40               <span class="fas fa-lock"></span>
41             </div>
42           </div>
43           <small id="error-password" class="error-text text-danger"></small>
44         </div>
45         <div class="row">
46           <div class="col-8">
47             <div class="icheck-primary">
48               <input type="checkbox" id="remember"><label for="remember">Remember Me</label>
49             </div>
50           </div>
51           <!-- /.col -->
52           <div class="col-4">
53             <button type="submit" class="btn btn-primary btn-block">Sign In</button>
54           </div>
55           <!-- /.col -->
56         </div>
57         <br>
58         <p>Don't have an account? <a href="{{ url('signup') }}">Register</a></p>
59       </form>
60     </div>
61   <!-- /.card-body -->
62 </div>
```



	<pre>62     &lt;/div&gt; 63     &lt;!-- /.card --&gt; 64 &lt;/div&gt; 65 &lt;!-- /.login-box --&gt; 66 &lt;!-- jQuery --&gt; 67 &lt;script src="{ asset('adminlte/plugins/jquery/jquery.min.js') }}"&gt;&lt;/script&gt; 68 &lt;!-- Bootstrap 4 --&gt; 69 &lt;script src="{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"&gt;&lt;/script&gt; 70 &lt;!-- jquery-validation --&gt; 71 &lt;script src="{ asset('adminlte/plugins/jquery-validation/jquery.validate.min.js') }}"&gt;&lt;/script&gt; 72 &lt;script src="{ asset('adminlte/plugins/jquery-validation/additional-methods.min.js') }}"&gt;&lt;/script&gt; 73 &lt;!-- SweetAlert2 --&gt; 74 &lt;script src="{ asset('adminlte/plugins/sweetalert2/sweetalert2.min.js') }}"&gt;&lt;/script&gt; 75 &lt;!-- AdminLTE App --&gt; 76 &lt;script src="{ asset('adminlte/dist/js/adminlte.min.js') }}"&gt;&lt;/script&gt; 77 78 &lt;script&gt; 79     \$.ajaxSetup({ 80         headers: { 81             'X-CSRF-TOKEN': \$('meta[name="csrf-token"]').attr('content') 82         } 83     }); 84 85     \$(document).ready(function() { 86         \$("#form-login").validate({ 87             rules: { 88                 username: {required: true, minlength: 4, maxlength: 20}, 89                 password: {required: true, minlength: 5, maxlength: 20} 90             }, 91             submitHandler: function(form) { // ketika valid, maka bagian yg akan dijalankan 92                 \$.ajax({ 93                     url: form.action, 94                     type: form.method, 95                     data: \$(form).serialize(), 96                     success: function(response) { 97                         if(response.status) { // jika sukses 98                             Swal.fire({ 99                                 icon: 'success', 100                                title: 'Berhasil', 101                                text: response.message, 102                            }).then(function() { 103                                window.location = response.redirect; 104                            }); 105                        } else { // jika error 106                            \$(".error-text").text(''); 107                            \$.each(response.msgField, function(prefix, val) { 108                                \$("#error-"+prefix).text(val[0]); 109                            }); 110                            Swal.fire({ 111                                icon: 'error', 112                                title: 'Terjadi Kesalahan', 113                                text: response.message 114                            }); 115                        } 116                    } 117                }); 118                return false; 119            }, 120            errorElement: 'span', 121            errorPlacement: function (error, element) { 122                error.addClass('invalid-feedback'); 123                element.closest('.input-group').append(error); 124            }, 125            highlight: function (element, errorClass, validClass) { 126                \$(element).addClass('is-invalid'); 127            }, 128            unhighlight: function (element, errorClass, validClass) { 129                \$(element).removeClass('is-invalid'); 130            } 131        }); 132    }); 133 &lt;/script&gt; 134 &lt;/body&gt; 135 &lt;/html&gt;</pre>
5.	Kemudian kita modifikasi <a href="#">route/web.php</a> agar semua route masuk dalam auth




	<pre>// JS 4.1 Auth Route::pattern('id', '[0-9]*'); //ketika ada parameter {id}, maka harus berupa angka  // Route::get('signup', [RegistrationController::class, 'registration']-&gt;name('signup')); // Route::post('signup', [RegistrationController::class, 'store']);  Route::get('login', [AuthController::class, 'login']-&gt;name('login')); Route::post('login', [AuthController::class, 'postlogin']); Route::get('logout', [AuthController::class, 'logout']-&gt;middleware('auth'));  Route::middleware(['auth'])-&gt;group(function(){     //semua route yang perlu autentikasi     Route::get('/', [WelcomeController::class, 'index']);      //Semua route di grup ini harus punya role ADM (Administrator)     Route::group(['prefix' =&gt; 'user', 'middleware'=&gt; 'authorize:ADM'], function(){         Route::get('/', [UserController::class, 'index']); //menampilkan laman awal user         Route::post('/list', [UserController::class, 'list']); //menampilkan data user dalam bentuk json untuk datatables         Route::get('/create', [UserController::class, 'create']); //menampilkan laman form tambah user         Route::post('/', [UserController::class, 'store']); //menyimpan data user baru         Route::get('/create_ajax', [UserController::class, 'create_ajax']); //menampilkan laman form tambah user AJAX         Route::post('/ajax', [UserController::class, 'store_ajax']); //menyimpan data user baru AJAX         Route::get('/{id}', [UserController::class, 'show']); //menampilkan detail user         Route::get('/{id}/edit', [UserController::class, 'edit']); //menampilkan laman form edit user         Route::put('/{id}', [UserController::class, 'update']); //menyimpan perubahan data user         Route::get('/{id}/edit_ajax', [UserController::class, 'edit_ajax']); //menampilkan laman form edit user AJAX         Route::put('/{id}/update_ajax', [UserController::class, 'update_ajax']); //menyimpan perubahan data user AJAX         Route::get('/{id}/delete_ajax', [UserController::class, 'confirm_ajax']); //menampilkan form confirm hapus data user AJAX         Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']); //menghapus data user AJAX         Route::delete('/{id}', [UserController::class, 'destroy']); //menghapus data user     });      //Semua route di grup ini harus punya role ADM (Administrator)     Route::group(['prefix' =&gt; 'level', 'middleware'=&gt; 'authorize:ADM'], function(){         Route::get('/', [LevelController::class, 'index']); //menampilkan laman awal level         Route::post('/list', [LevelController::class, 'list']); //menampilkan data level dalam bentuk json untuk datatables         Route::get('/create_ajax', [LevelController::class, 'create_ajax']); //menampilkan laman form tambah level AJAX         Route::post('/ajax', [LevelController::class, 'store_ajax']); //menyimpan data level baru AJAX         Route::get('/{id}/edit_ajax', [LevelController::class, 'edit_ajax']); //menampilkan laman form edit level AJAX         Route::put('/{id}/update_ajax', [LevelController::class, 'update_ajax']); //menyimpan perubahan data level AJAX         Route::get('/{id}/delete_ajax', [LevelController::class, 'confirm_ajax']); //menampilkan form confirm hapus data level AJAX         Route::delete('/{id}/delete_ajax', [LevelController::class, 'delete_ajax']); //menghapus data level AJAX     });      //Semua route di grup ini harus punya role ADM (Administrator) dan MNG (Manager)     Route::group(['prefix' =&gt; 'kategori', 'middleware'=&gt; 'authorize:ADM,MNG'], function(){         Route::get('/', [KategoriController::class, 'index']); //menampilkan laman awal kategori         Route::post('/list', [KategoriController::class, 'list']); //menampilkan data kategori dalam bentuk json untuk datatables         Route::get('/create_ajax', [KategoriController::class, 'create_ajax']); //menampilkan laman form tambah kategori AJAX         Route::post('/ajax', [KategoriController::class, 'store_ajax']); //menyimpan data kategori baru AJAX         Route::get('/{id}/edit_ajax', [KategoriController::class, 'edit_ajax']); //menampilkan laman form edit kategori AJAX         Route::put('/{id}/update_ajax', [KategoriController::class, 'update_ajax']); //menyimpan perubahan data kategori AJAX         Route::get('/{id}/delete_ajax', [KategoriController::class, 'confirm_ajax']); //menampilkan form confirm hapus data kategori AJAX         Route::delete('/{id}/delete_ajax', [KategoriController::class, 'delete_ajax']); //menghapus data kategori AJAX     });      //Semua route di grup ini harus punya role ADM (Administrator)     Route::group(['prefix' =&gt; 'supplier', 'middleware'=&gt; 'authorize:ADM'], function(){         Route::get('/', [SupplierController::class, 'index']); //menampilkan laman awal supplier         Route::post('/list', [SupplierController::class, 'list']); //menampilkan data supplier dalam bentuk json untuk datatables         Route::get('/create_ajax', [SupplierController::class, 'create_ajax']); //menampilkan laman form tambah supplier AJAX         Route::post('/ajax', [SupplierController::class, 'store_ajax']); //menyimpan data supplier baru AJAX         Route::get('/{id}/edit_ajax', [SupplierController::class, 'edit_ajax']); //menampilkan laman form edit supplier AJAX         Route::put('/{id}/update_ajax', [SupplierController::class, 'update_ajax']); //menyimpan perubahan data supplier AJAX         Route::get('/{id}/delete_ajax', [SupplierController::class, 'confirm_ajax']); //menampilkan form confirm hapus data supplier AJAX         Route::delete('/{id}/delete_ajax', [SupplierController::class, 'delete_ajax']); //menghapus data supplier AJAX     });      //Semua route di grup ini harus punya role ADM (Administrator) dan MNG (Manager)     Route::group(['prefix' =&gt; 'barang', 'middleware'=&gt; 'authorize:ADM,MNG'], function(){         Route::get('/', [BarangController::class, 'index']); //menampilkan laman awal barang         Route::post('/list', [BarangController::class, 'list']); //menampilkan data barang dalam bentuk json untuk datatables         Route::get('/create_ajax', [BarangController::class, 'create_ajax']); //menampilkan laman form tambah barang AJAX         Route::post('/ajax', [BarangController::class, 'store_ajax']); //menyimpan data barang baru AJAX         Route::get('/{id}/edit_ajax', [BarangController::class, 'edit_ajax']); //menampilkan laman form edit barang AJAX         Route::put('/{id}/update_ajax', [BarangController::class, 'update_ajax']); //menyimpan perubahan data barang AJAX         Route::get('/{id}/delete_ajax', [BarangController::class, 'confirm_ajax']); //menampilkan form confirm hapus data barang AJAX         Route::delete('/{id}/delete_ajax', [BarangController::class, 'delete_ajax']); //menghapus data barang AJAX     }); });</pre>
6.	Ketika kita coba mengakses halaman <a href="http://localhost/PWL_POS/public">localhost/PWL_POS/public</a> makan akan tampil halaman awal untuk login ke aplikasi




## Halaman Login

# AdminLTE

Sign in to start your session



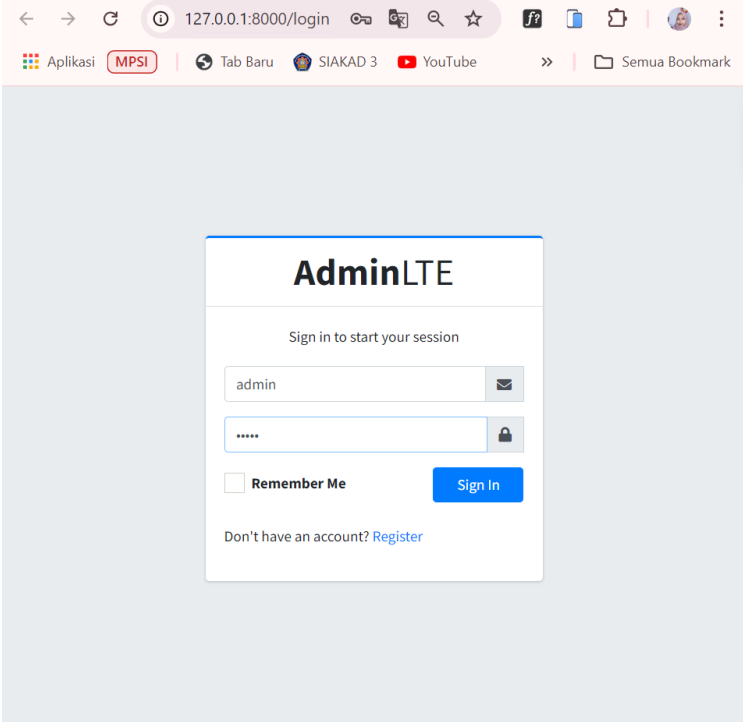
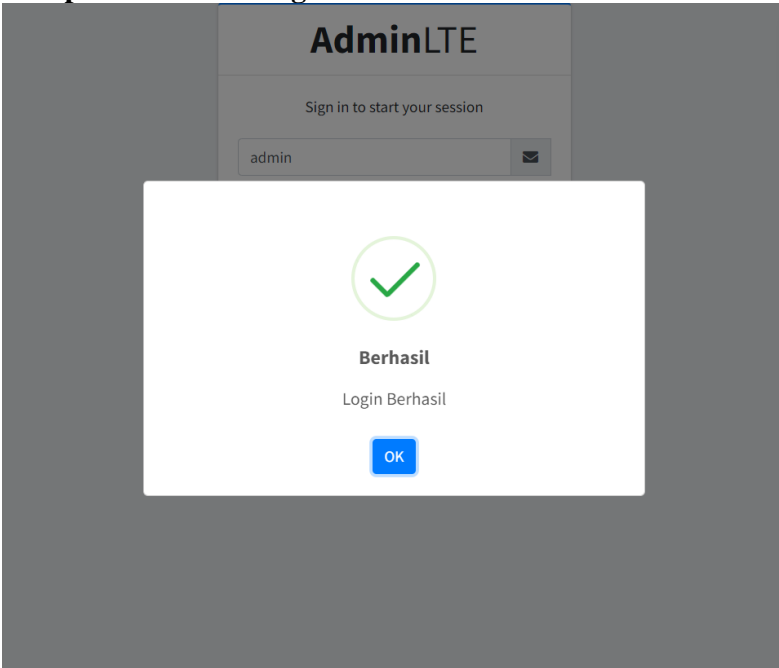


☐ **Remember Me**

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

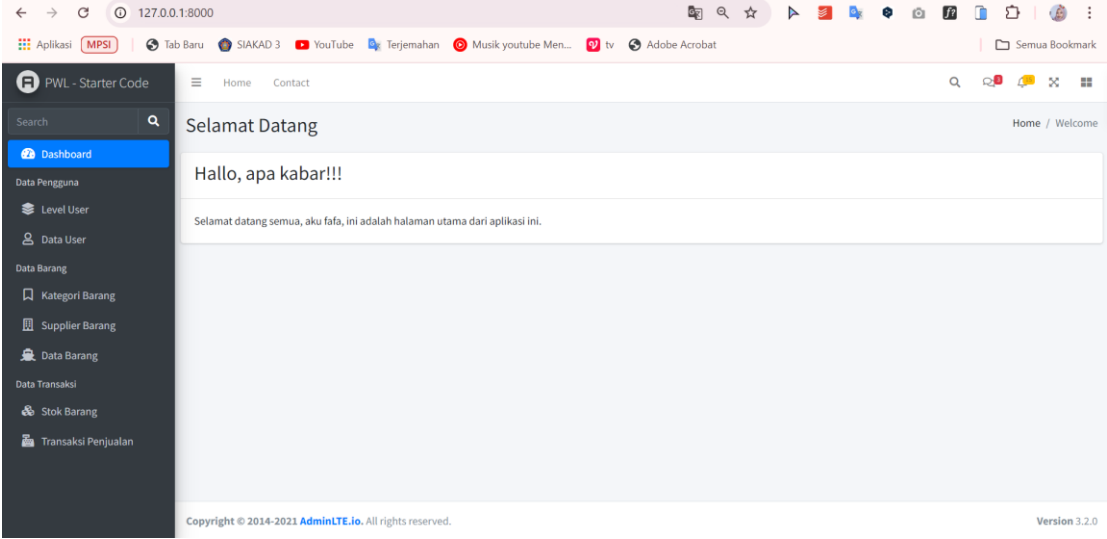


## Tugas Ke-1 | Implementasi Authentication

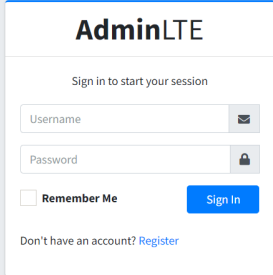
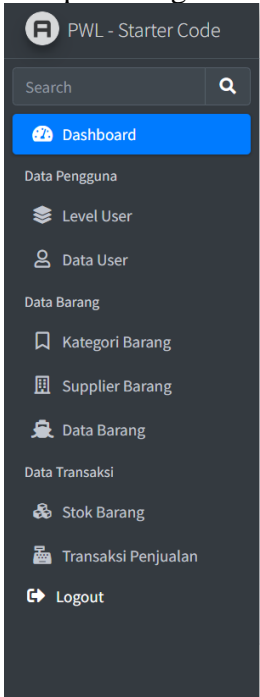
Langkah	Jawaban/Deskripsi
1.	<p>Silahkan implementasikan proses login pada project kalian masing-masing</p> <p><b>Tampilan Login</b></p>  <p><b>Tampilan Berhasil Login</b></p> 





	<p><b>Tampilan Setelah Login Berhasil Diarahkan Ke Beranda</b></p> 
2.	<p>Silahkan implementasi proses logout pada halaman web yang kalian buat</p> <ul style="list-style-type: none"><li>• Web.php<pre>Route::get(uri: 'logout', action: [AuthController::class, 'logout'])-&gt;middleware(middleware: 'a</pre></li><li>• AuthController.php<pre>1 reference   0 overrides public function logout(Request \$request): Redirector RedirectResponse {     Auth::logout();      \$request-&gt;session()-&gt;invalidate();     \$request-&gt;session()-&gt;regenerateToken();     return redirect(to: 'login'); }</pre></li><li>• sidebar.php<pre>&lt;/li&gt; &lt;li class="nav-item"&gt;     &lt;form action="{ url('/logout') }}" method="POST" id="logout-form" style="display: inline;"&gt;         @csrf         &lt;button type="submit" class="nav-link" style="border: none; background: none; padding: 5px 10px; color: white; text-align: left;"&gt;             &lt;i class="nav-icon fas fa-sign-out-alt"&gt;&lt;/i&gt;             &lt;p style="margin: 0;"&gt;Logout&lt;/p&gt;         &lt;/button&gt;     &lt;/form&gt; &lt;/li&gt; &lt;/ul&gt; &lt;/nav&gt; &lt;/div&gt; &lt;!-- /.sidebar --&gt; &lt;/aside&gt;</pre></li></ul>
3.	<p>Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan</p> <ul style="list-style-type: none"><li>• Tampilan Login</li></ul>



	<div><p>AdminLTE</p><p>Sign in to start your session</p><p>Username <input type="text"/></p><p>Password <input type="password"/></p><p><input type="checkbox"/> Remember Me <input type="button" value="Sign In"/></p><p>Don't have an account? <a href="#">Register</a></p></div> <ul style="list-style-type: none"><li>• <b>Tampilan Logout</b><div><p>PWL - Starter Code</p><p>Search <input type="text"/></p><p><b>Dashboard</b></p><p>Data Pengguna</p><ul style="list-style-type: none"><li>Level User</li><li>Data User</li></ul><p>Data Barang</p><ul style="list-style-type: none"><li>Kategori Barang</li><li>Supplier Barang</li><li>Data Barang</li></ul><p>Data Transaksi</p><ul style="list-style-type: none"><li>Stok Barang</li><li>Transaksi Penjualan</li></ul><p>Logout</p></div></li></ul>
4.	Submit kode untuk impementasi Authentication pada repository github kalian.

## Praktikum Ke-2 | Implementasi *Authorizaton* di Laravel dengan Middleware

Langkah	Jawaban/Deskripsi
1.	Kita modifikasi <code>UserModel.php</code> dengan menambahkan kode berikut



	<pre>// Relasi ke tabel level 0 references   0 overrides public function level(): BelongsTo {     return \$this-&gt;belongsTo(related: LevelModel::class, foreignKey: 'level_id'); }  // Mendapatkan nama role 0 references   0 overrides public function getRoleName(): string{     return \$this-&gt;level-&gt;level_nama; }  // Cek apakah user memiliki role tertentu 0 references   0 overrides public function hasRole(\$role): bool{     return \$this-&gt;level-&gt;level_kode == \$role; } }</pre>
2.	<p>Kemudian kita buat <i>middleware</i> dengan nama <b>AuthorizeUser.php</b>. Kita bisa buat <i>middleware</i> dengan mengetikkan perintah pada terminal/CMD</p> <pre>PS C:\laragon\www\PWL_POS&gt; php artisan make:middleware AuthorizeUser</pre> <p><b>INFO</b> Middleware [C:\laragon\www\PWL_POS\app\Http\Middleware\AuthorizeUser.php] created successfully.</p> <pre>PS C:\laragon\www\PWL_POS&gt;</pre>
3.	<p>Kemudian kita edit <i>middleware</i> <b>AuthorizeUser.php</b> untuk bisa mengecek apakah pengguna yang mengakses memiliki Level/Role/Group yang sesuai</p> <pre>0 references   0 overrides public function handle(Request \$request, Closure \$next, \$role = ''): Response 'har {     \$user = \$request-&gt;user();     if(\$user-&gt;hasRole(\$role)){         return \$next(\$request);     } }</pre>
4.	<p>Kita daftarkan ke <b>app/Http/Kernel.php</b> untuk <i>middleware</i> yang kita buat barusan</p> <pre>0 references protected \$middlewareAliases = [     'auth' =&gt; \App\Http\Middleware\Authenticate::class,     'authorize'=&gt;\App\Http\Middleware\AuthorizeUser::class,     'auth.basic' =&gt; \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,     'auth.session' =&gt; \Illuminate\Session\Middleware\AuthenticateSession::class,     'cache.headers' =&gt; \Illuminate\Http\Middleware\SetCacheHeaders::class,     'can' =&gt; \Illuminate\Auth\Middleware\Authorize::class,     'guest' =&gt; \App\Http\Middleware\RedirectIfAuthenticated::class,     'password.confirm' =&gt; \Illuminate\Auth\Middleware\RequirePassword::class,     'precognitive' =&gt; \Illuminate\Foundation\Http\Middleware\HandlePrecognitiveRequests::c     'signed' =&gt; \App\Http\Middleware\ValidateSignature::class,     'throttle' =&gt; \Illuminate\Routing\Middleware\ThrottleRequests::class,     'verified' =&gt; \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class, ]; }</pre>
5.	<p>Sekarang kita perhatikan tabel <b>m_level</b> yang menjadi tabel untuk menyimpan level/group/role dari user ada</p>



	<div><div>← T →</div><table><tr><th></th><th>level_id</th><th>level_kode</th><th>level_nama</th><th>created_at</th><th>updated_at</th></tr><tr><td><input type="checkbox"/>  Edit  Copy  Delete</td><td>1</td><td>ADM</td><td>Administrator</td><td>NULL</td><td>NULL</td></tr><tr><td><input type="checkbox"/>  Edit  Copy  Delete</td><td>2</td><td>MNG</td><td>Manager</td><td>NULL</td><td>NULL</td></tr><tr><td><input type="checkbox"/>  Edit  Copy  Delete</td><td>3</td><td>STF</td><td>Staff/kasir</td><td>NULL</td><td>NULL</td></tr><tr><td><input type="checkbox"/>  Edit  Copy  Delete</td><td>4</td><td>PLG</td><td>Pelanggan</td><td>NULL</td><td>NULL</td></tr></table></div>		level_id	level_kode	level_nama	created_at	updated_at	<input type="checkbox"/> Edit  Copy  Delete	1	ADM	Administrator	NULL	NULL	<input type="checkbox"/> Edit  Copy  Delete	2	MNG	Manager	NULL	NULL	<input type="checkbox"/> Edit  Copy  Delete	3	STF	Staff/kasir	NULL	NULL	<input type="checkbox"/> Edit  Copy  Delete	4	PLG	Pelanggan	NULL	NULL
	level_id	level_kode	level_nama	created_at	updated_at																										
<input type="checkbox"/> Edit  Copy  Delete	1	ADM	Administrator	NULL	NULL																										
<input type="checkbox"/> Edit  Copy  Delete	2	MNG	Manager	NULL	NULL																										
<input type="checkbox"/> Edit  Copy  Delete	3	STF	Staff/kasir	NULL	NULL																										
<input type="checkbox"/> Edit  Copy  Delete	4	PLG	Pelanggan	NULL	NULL																										
6.	<p>Untuk mencoba <i>authorization</i> yang telah kita buat, maka perlu kita modifikasi <code>route/web.php</code> untuk menentukan route mana saja yang akan diberi hak akses sesuai dengan level user</p> <p>Hanya Bisa diakses ADM</p> <pre>//Semua route di grup ini harus punya role ADM (Administrator) Route::group(attributes: ['prefix' =&gt; 'user', 'middleware' =&gt; 'authorize:ADM'], routes: functi     Route::get(uri: '/', action: [UserController::class, 'index']);     Route::post(uri: '/list', action: [UserController::class, 'list']);     Route::get(uri: '/create', action: [UserController::class, 'create']);     Route::post(uri: '/', action: [UserController::class, 'store']);     Route::get(uri: '/create_ajax', action: [UserController::class, 'create_ajax']);     Route::post(uri: '/ajax', action: [UserController::class, 'store_ajax']);     Route::get(uri: '/{id}', action: [UserController::class, 'show']);     Route::get(uri: '/{id}/edit', action: [UserController::class, 'edit']);     Route::put(uri: '/{id}', action: [UserController::class, 'update']);     Route::get(uri: '/{id}/edit_ajax', action: [UserController::class, 'edit_ajax']);     Route::put(uri: '/{id}/update_ajax', action: [UserController::class, 'update_ajax']);     Route::get(uri: '/{id}/delete_ajax', action: [UserController::class, 'confirm_ajax']);     Route::delete(uri: '/{id}/delete_ajax', action: [UserController::class, 'delete_ajax']);     Route::delete(uri: '/{id}', action: [UserController::class, 'destroy']); });</pre> <p>Pada kode yang ditandai merah, terdapat <code>authorize:ADM</code> . Kode <code>ADM</code> adalah nilai dari <code>level_kode</code> pada tabel <code>m_level</code>. Yang artinya, user yang bisa mengakses route untuk manage data level, adalah user yang memiliki level sebagai Administrator.</p>																														
7.	<p>Untuk membuktikannya, sekarang kita coba login menggunakan akun selain level administrator, dan kita akses route menu level tersebut</p> <ul style="list-style-type: none"><li>• Login Menggunakan Manager</li></ul>																														



## AdminLTE


Sign in to start your session

☐ Remember Me

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

## AdminLTE

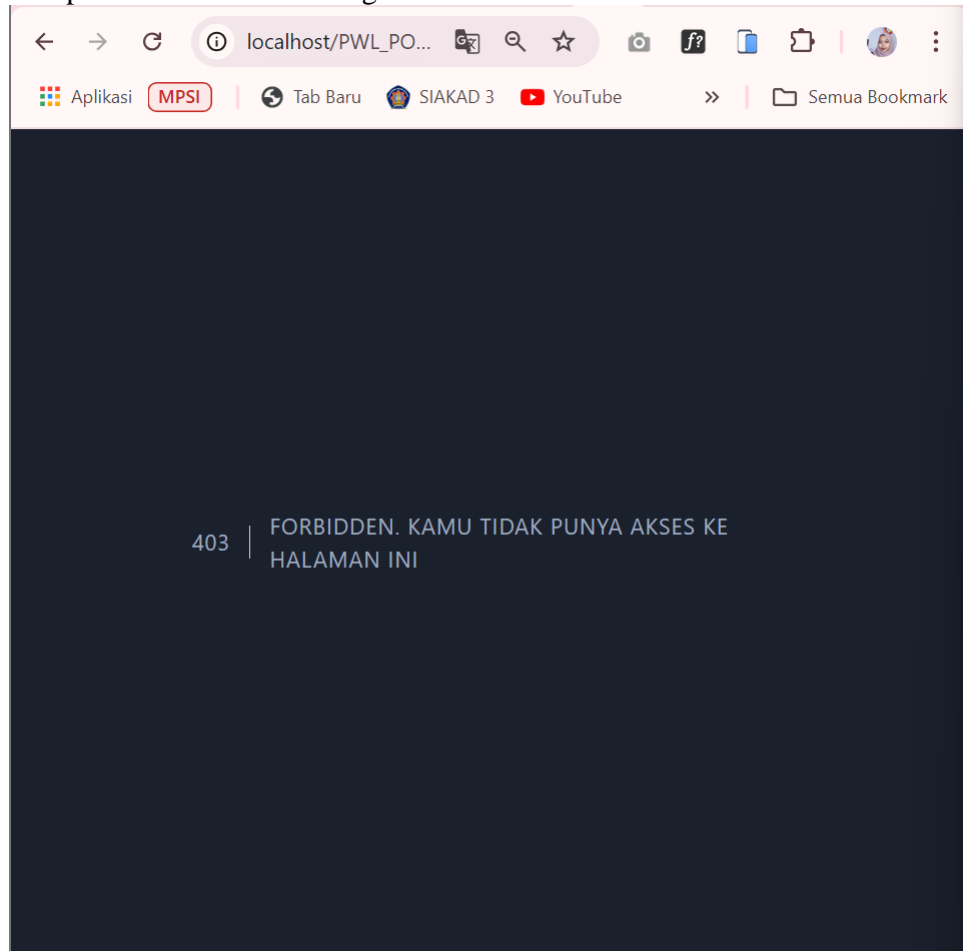
Sign in to start your session



**Berhasil**  
Login Berhasil



- Tampilan Menu Level Manager



Hal tersebut terjadi karena yang memiliki akses hanyalah administrator, manager tidak memiliki akses untuk masuk ke halaman

## Tugas Ke-2 | Implementasi Authoriization

Langkah	Jawaban/Deskripsi
1.	Apa yang kalian pahami pada praktikum 2 ini? <b>Jawab:</b> Autorhize digunakan untuk role role tertentu saja
2.	Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan <b>Jawab:</b> Hasil dari praktikum 2 menampilkan pesan 403, hal ini dikarenakan saya masuk dengan role manager. Akan tetapi halaman web hanya bisa di akses oleh role admin 'ADM'
3.	Submit kode untuk impementasi Authorization pada repository github kalian.



### Praktikum Ke-3 | Implementasi Multi-Level Authorizaton di Laravel dengan Middleware

Langkah	Jawaban/Deskripsi
1.	<p>Kita 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>//Mendapatkan kode role 0 references   0 overrides public function getRole(): mixed {     return \$this-&gt;level-&gt;level_kode; }</pre>
2.	<p>Selanjutnya, Kita modifikasi middleware <code>AuthorizeUser.php</code> dengan kode berikut</p> <pre>0 references   0 overrides public function handle(Request \$request, Closure \$next, ... \$roles): Response {     // \$user = \$request-&gt;user();     // if(\$user-&gt;hasRole(\$role)){     //     return \$next(\$request);     // }      \$user_role = \$request-&gt;user()-&gt;getRole();     if (in_array(needle: \$user_role, haystack: \$roles)){         return \$next(\$request);     }      //jika tidak punya role, maka tampilkan error 403     abort(code: 403, message: 'Forbidden. Kamu tidak punya akses ke halaman ini'); }</pre>
3.	<p>Setelah itu tinggal kita perbaiki <code>route/web.php</code> sesuaikan dengan role/level yang diinginkan. Contoh</p> <pre>//Semua route di grup ini harus punya role ADM (Administrator) dan MNG (Manager) Route::group(attributes: ['prefix' =&gt; 'kategori', 'middleware'=&gt; 'authorize:ADM,MNG'], ro Route::get(uri: '/', action: [KategoriController::class, 'index']); Route::post(uri: '/list', action: [KategoriController::class, 'list']); Route::get(uri: '/create_ajax', action: [KategoriController::class, 'create_ajax']); Route::post(uri: '/ajax', action: [KategoriController::class, 'store_ajax']); Route::get(uri: '/{id}/edit_ajax', action: [KategoriController::class, 'edit_ajax']); Route::put(uri: '/{id}/update_ajax', action: [KategoriController::class, 'update_ajax Route::get(uri: '/{id}/delete_ajax', action: [KategoriController::class, 'confirm_aja Route::delete(uri: '/{id}/delete_ajax', action: [KategoriController::class, 'delete_a ));</pre>
4.	<p>Sekarang kita sudah bisa memberikan hak akses menu/route ke beberapa level user</p>



PWL - Starter Code

HomeContact

Search

Dashboard

Data Pengguna

Level User

Data User

Data Barang

Kategori Barang

Supplier Barang

Data Barang

Data Transaksi

Stok Barang

Transaksi Penjualan

Logout

Daftar Kategori

Home / Kategori

Daftar Kategori yang terdaftar dalam sistem

TambahTambah Ajax

Show 10 entries

Search:

ID	Kode	Nama Kategori	Aksi
1	EK	Elektronik	DetailEditHapus
2	KC	Kecantikan	DetailEditHapus
3	OB	Obat-obatan	DetailEditHapus
4	MM	Makanan dan Minuman	DetailEditHapus
5	PK	Pakaian	DetailEditHapus

Showing 1 to 5 of 5 entries

Previous1Next

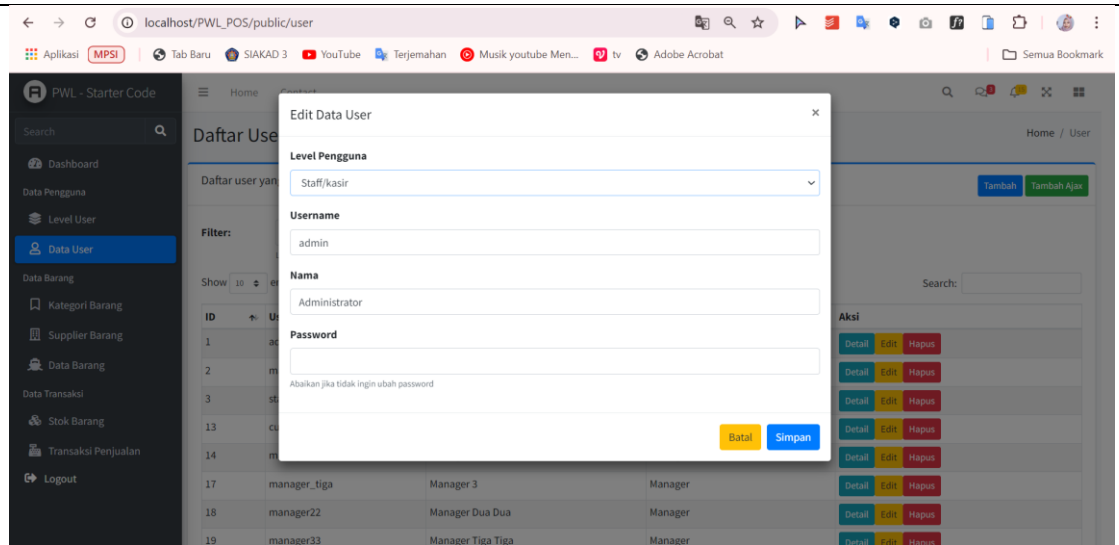
Copyright © 2014-2021 AdminLTE.io. All rights reserved.

Version 3.2.0

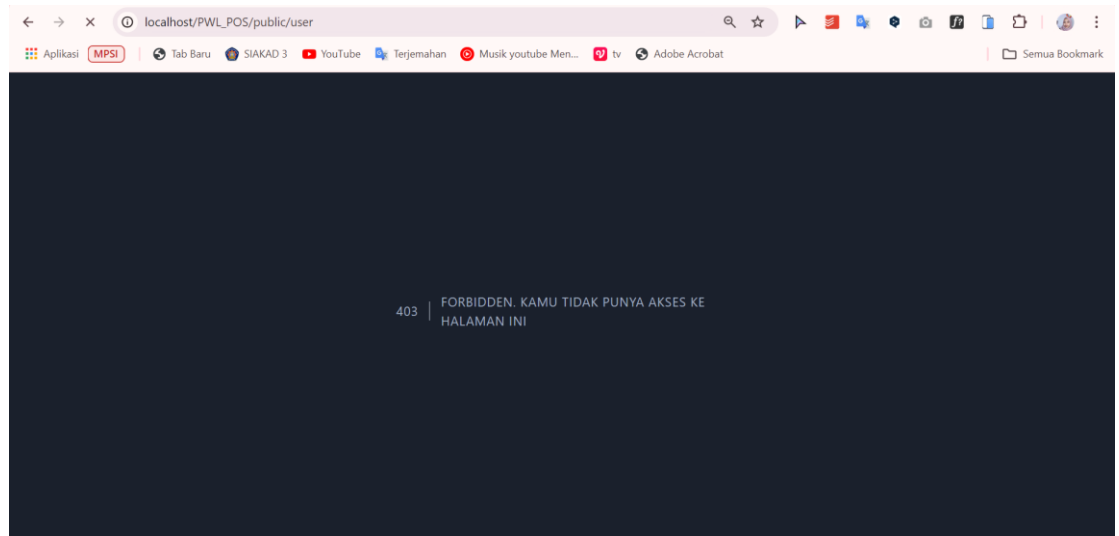
### Tugas Ke-3 | Implementasi Multi-Level Authorization

Langkah	Jawaban/Deskripsi																								
1.	Silahkan implementasikan multi-level authorization pada project kalian masing-masing																								
2.	<div>Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan</div> <div><div><div><div>PWL - Starter Code</div><div>Home Contact</div><div>Search</div><div>Dashboard</div><div>Data Pengguna</div><div>Level User</div><div>Data User</div><div>Data Barang</div><div>Kategori Barang</div><div>Supplier Barang</div><div>Data Barang</div><div>Data Transaksi</div><div>Stok Barang</div><div>Transaksi Penjualan</div><div>Logout</div></div><div><div>Daftar Kategori</div><div>Home / Kategori</div><div>Daftar Kategori yang terdaftar dalam sistem</div><div>Tambah Tambah Ajax</div><div>Show 10 entries</div><div>Search:</div><table><tr><th>ID</th><th>Kode</th><th>Nama Kategori</th><th>Aksi</th></tr><tr><td>1</td><td>EK</td><td>Elektronik</td><td>Detail Edit Hapus</td></tr><tr><td>2</td><td>KC</td><td>Kecantikan</td><td>Detail Edit Hapus</td></tr><tr><td>3</td><td>OB</td><td>Obat-obatan</td><td>Detail Edit Hapus</td></tr><tr><td>4</td><td>MM</td><td>Makanan dan Minuman</td><td>Detail Edit Hapus</td></tr><tr><td>5</td><td>PK</td><td>Pakaian</td><td>Detail Edit Hapus</td></tr></table><div>Showing 1 to 5 of 5 entries</div><div>Previous 1 Next</div><div>Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0</div></div></div><div>Saya mencoba masuk dengan role manager, dan dapat menampilkan kategori barang. Hal ini dikarenakan level_code manager yaitu MNG di tambahkan dalam middleware pada route kategori barang.</div></div>	ID	Kode	Nama Kategori	Aksi	1	EK	Elektronik	Detail Edit Hapus	2	KC	Kecantikan	Detail Edit Hapus	3	OB	Obat-obatan	Detail Edit Hapus	4	MM	Makanan dan Minuman	Detail Edit Hapus	5	PK	Pakaian	Detail Edit Hapus
ID	Kode	Nama Kategori	Aksi																						
1	EK	Elektronik	Detail Edit Hapus																						
2	KC	Kecantikan	Detail Edit Hapus																						
3	OB	Obat-obatan	Detail Edit Hapus																						
4	MM	Makanan dan Minuman	Detail Edit Hapus																						
5	PK	Pakaian	Detail Edit Hapus																						
3.	<div>Implementasikan multi-level authorization untuk semua Level/Jenis User dan Menu-menu yang sesuai dengan Level/Jenis User</div> <div><ul style="list-style-type: none"><li>Level admin saya ganti dengan staff/kasir</li></ul></div>																								

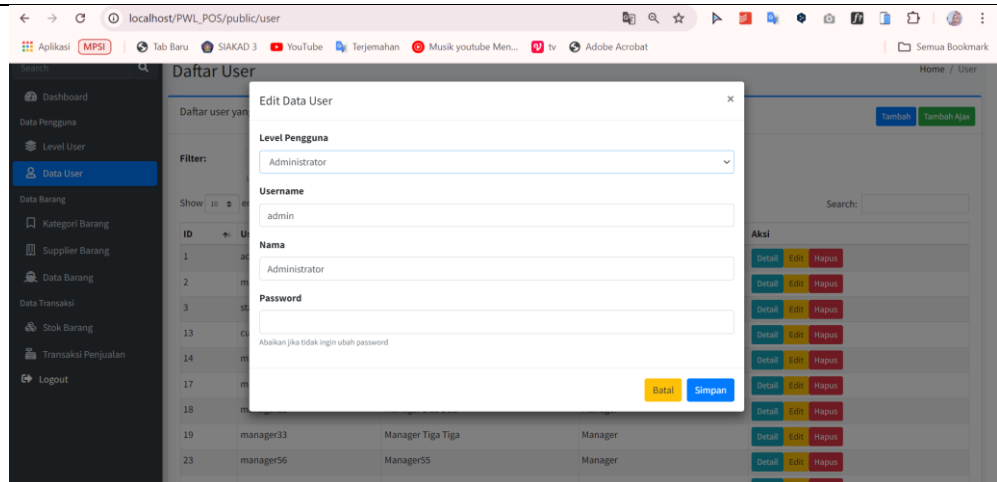




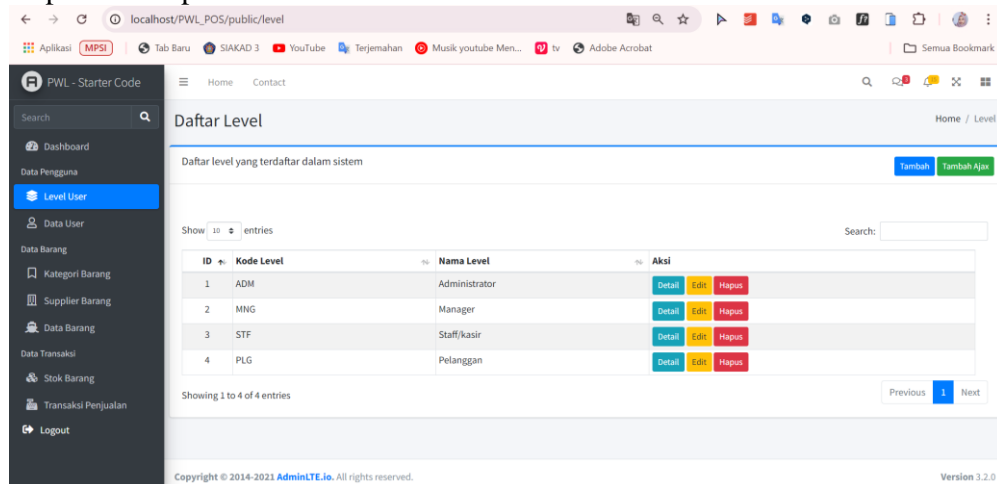
- Hasilnya akan menjadi forbidden 403, karena levelnya diganti dengan staff. Sedangkan, pada code yang dimasukkan level admin dan manager



- Jika saya kembalikan lagi levelnya menjadi admin (ADM) dengan menggunakan role manager.



- Dapat menampilkan level user kembali



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

#### Tugas Ke-4 | Implementasi Form Registrasi

Langkah	Jawaban/Deskripsi
1.	Silahkan implementasikan form untuk registrasi user.
2.	<p>Screenshot hasil yang kalian kerjakan</p> <ul style="list-style-type: none"><li>• Kode login.blade.php</li></ul> <pre>56 &lt;/div&gt; 57 &lt;br&gt; 58 &lt;p&gt;Don't have an account? &lt;a href="{{ url('signup') }}"&gt;Sign up&lt;/a&gt;&lt;/p&gt; 59 &lt;/form&gt; 60 &lt;/div&gt;</pre> <ul style="list-style-type: none"><li>• Kode RegistrationController.php</li></ul>



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141

Telp. (0341) 404424 – 404425, Fax (0341) 404420

<https://jti.polinema.ac.id>

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\LevelModel;
6 use App\Models\UserModel;
7 use Illuminate\Http\Request;
8 use Illuminate\Support\Facades\Auth;
9 use Illuminate\Support\Facades\Validator;
10
11 class RegistrationController 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             ->with('level', $level);
20     }
21
22     // Menyimpan data user baru
23     public function store(Request $request)
24     {
25         $validator = Validator::make($request->all(), [
26             //username harus diisi, berupa string, minimal 3 karakter, dan bernilai unik di tabel m_user kolom username
27             'username' => 'required|string|min:3|unique:m_user,username',
28             'nama' => 'required|string|max:100', //nama harus diisi, berupa string, dan maksimal 100 karakter
29             'password' => 'required|min:5', //password harus diisi dan minimal 5 karakter
30             'level_id' => 'required|integer' //level_id harus diisi dan berupa angka
31         ]);
32
33         if ($validator->fails()) {
34             // Jika validasi gagal, kirim respons JSON dengan pesan error
35             return response()->json([
36                 'status' => false,
37                 'message' => 'Validasi gagal.',
38                 'msgField' => $validator->messages()
39             ], 422);
40         }
41
42         UserModel::create([
43             'username' => $request->username,
44             'nama' => $request->nama,
45             'password' => bcrypt($request->password), //password dienkripsi sebelum disimpan
46             'level_id' => $request->level_id
47         ]);
48
49         // Kirim respons sukses
50         return response()->json([
51             'status' => true,
52             'message' => 'Registrasi berhasil dilakukan!',
53             'redirect' => url('/login') // Redirect ke laman Login
54         ]);
55     }
56 }
```

- Kode web.php

```
54
55 Route::get(uri: 'signup', action: [RegistrationController::class, 'registration'])->name(name: 'signup');
56 Route::post(uri: 'signup', action: [RegistrationController::class, 'store']);
```

- Kode signup.blade.php



```
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>Signup Pengguna</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=fallback">
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   <!-- /.login-logo -->
21   <div class="card card-outline card-primary">
22     <div class="card-header text-center"><a href="{{ url('/') }}" class="h1"><b>Admin</b></a></div>
23     <div class="card-body">
24       <p class="login-box-msg">Sign up to create your account</p>
25       <form action="{{ url('signup') }}" method="POST" id="form-tambah">
26         @csrf
27         <div class="form-group">
28           <label>Level</label>
29           <div class="col-11">
30             <select class="form-control" id="level_id" name="level_id" required>
31               <option value="">- Pilih Level -</option>
32               @foreach($level as $item)
33                 <option value="{{ $item->level_id }}">{{ $item->level_nama }}</option>
34             @endforeach
35           </select>
36           <small id="error-level_id" class="error-text form-text text-danger"></small>
37         </div>
38         <div class="form-group">
39           <label>Username</label>
40           <div class="col-11">
41             <input type="text" class="form-control" id="username" name="username" value="{{ old('username') }}" required>
42             <small id="error-username" class="error-text form-text text-danger"></small>
43           </div>
44         </div>
45         <div class="form-group">
46           <label>Nama</label>
47           <div class="col-11">
48             <input type="text" class="form-control" id="nama" name="nama" value="{{ old('nama') }}" required>
49             <small id="error-nama" class="error-text form-text text-danger"></small>
50           </div>
51         </div>
52         <div class="form-group">
53           <label>Password</label>
54           <div class="col-11">
55             <input type="password" class="form-control" id="password" name="password" required>
56             <small id="error-password" class="error-text form-text text-danger"></small>
57           </div>
58         </div>
59         <div class="row">
60           <div class="col-8">
61             <div class="iCheck-primary">
62               <a class="btn btn-sm btn-default ml-1" href="{{ url('/') }}">Kembali</a>
63             </div>
64           </div>
65           <div class="col-4">
66             <button type="submit" class="btn btn-primary btn-block">Sign Up</button>
67           </div>
68         </div>
69       </form>
70     </div>
71   </div>
72   <!-- jQuery -->
73   <script src="{{ asset('adminlte/plugins/jquery/jquery.min.js') }}"></script>
74   <!-- Bootstrap 4 -->
75   <script src="{{ asset('adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
76   <!-- jquery-validation -->
77   <script src="{{ asset('adminlte/plugins/jquery-validation/jquery.validate.min.js') }}"></script>
78   <script src="{{ asset('adminlte/plugins/jquery-validation/additional-methods.min.js') }}"></script>
79   <!-- SweetAlert2 -->
80   <script src="{{ asset('adminlte/plugins/sweetalert2/sweetalert2.min.js') }}"></script>
81   <!-- AdminLTE App -->
82   <script src="{{ asset('adminlte/dist/js/adminlte.min.js') }}"></script>
83   <script>
84     $.ajaxSetup({
85       headers: {
86         'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
87       }
88     });
89     $(document).ready(function() {
90       $('#form-tambah').validate({
91         rules: {
92           level_id: {required: true, number: true},
93           username: {required: true, minlength: 3, maxlength: 20},
94           nama: {required: true, minlength: 3, maxlength: 100},
95           password: {required: true, minlength: 5, maxlength: 20}
96         }
97       });
98     });
99   </script>
```



```
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         },
128         errorElement: 'span',
129         errorPlacement: function (error, element) {
130             error.addClass('invalid-feedback');
131             element.closest('.form-group').append(error);
132         },
133         highlight: function (element, errorClass, validClass) {
134             $(element).addClass('is-invalid');
135         },
136         unhighlight: function (element, errorClass, validClass) {
137             $(element).removeClass('is-invalid');
138         }
139     });
140 });
141 </script>
142 </div>
143 <!-- /.card-body -->
144 </div>
145 <!-- /.card -->
146 </div>
147 <!-- /.login-box -->
```

## Output

- Tampilan halaman signup



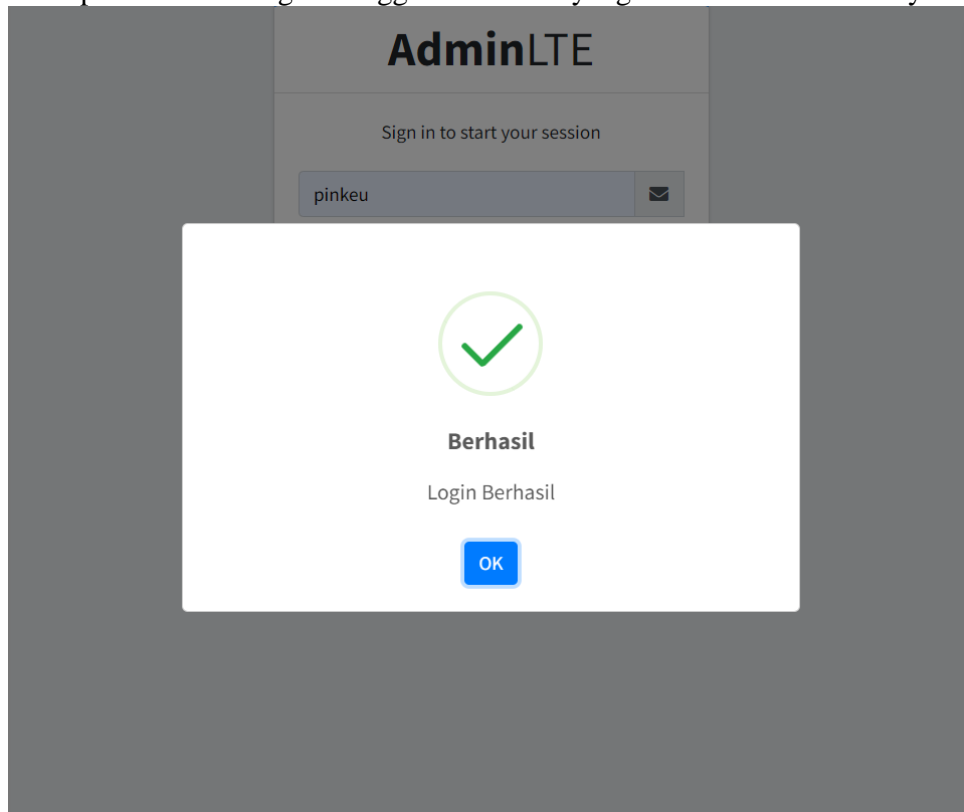
- Tampilan berhasil membuat akun

- Data akun tersimpan di database



	user_id	level_id	username	nama	password	created_at	updated_at
	1	1	admin	Administrator	\$2y\$12\$2W0JrC0ymRtwLcZtHxePbFeGu1FIE9UrtJkGMl8FgJ...	NULL	2024-10-11 07:39:22
	2	2	manager	Manager	\$2y\$12\$5vwwk35GraM02HJQgHE/upoJHGMIXEW2M5H3RzSooK...	NULL	NULL
	3	3	staff	Staff/Kasir	\$2y\$12\$2KNZKxqG77wJR86mBwLNeOLpVL3aOWH0qJvHKAgKN...	NULL	NULL
	13	4	customer 1	Pelanggan pertama	\$2y\$12\$5AgU8mj8BL4Ah9e7wboVv4RM7GsGPILHMc342vn O...	NULL	2024-09-14 08:24:23
	14	2	manager_dua	Manager2	\$2y\$12\$8C3a5A1ZKikEr1gJQgEAI m3NFYBCFnyoJCQ06PW4Oo...	NULL	NULL
	17	2	manager_tiga	Manager 3	\$2y\$12\$5eWmWmGtmMCSDID olak7g 51bzpEV2LEmTAcU s2xV...	NULL	NULL
	18	2	manager22	Manager Dua Dua	\$2y\$12\$26K6c83pFZeLUQys9eJScDhM8bsSY411uizevU...	2024-09-19 12:27:46	2024-09-19 12:27:46
	19	2	manager33	Manager Tiga Tiga	\$2y\$12\$5S 9JJW28JKzLDTHFURFJ S18WEICMpJzVSDKSwBO9...	2024-09-19 13:05:06	2024-09-19 13:05:06
	23	2	manager56	Manager55	\$2y\$12\$K4p9/WW3329i7cuE88hB5 voQyp0ByZSH66GTz3QROp...	2024-09-19 13:31:44	2024-09-19 13:31:44
	24	2	manager12	Manager11	\$2y\$12\$5A6y8ZeOKbJuGjLl4m8zxeA6dDgEGl0DFsO07EM lq...	2024-09-19 13:57:59	2024-09-19 13:57:59
	32	1	pinkeu	Fafa	\$2y\$12\$8Qbp12D1SH4xxHkvfsMKlqQK9Zkqsun.pnTmNQmglCD...	2024-10-16 02:07:47	2024-10-16 02:07:47

- Tampilan berhasil login menggunakan akun yang telah dibuat sebelumnya



3.

Commit dan push hasil tugas kalian ke masing-masing repo github kalian