



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

Kelas : SIB
NIM : 2241760026
Nama : Siti Faiqoh
Jobsheet Ke- : 7

Link Github : https://github.com/faiqoh04/PWL_POS

Laporan Jobsheet

Praktikum 1 – Implementasi Authentication :

Langkah	Jawaban/Deskripsi
1	<p>Kita buka project laravel PWL_POS kita, dan kita modifikasi konfigurasi aplikasi kita di config/auth.php</p> <pre>config > auth.php 3 return [62 'providers' => [63 'users' => [64 'driver' => 'eloquent', 65 'model' => App\Models\User::class, 66], 67],</pre> <p>Pada bagian ini kita sesuaikan dengan Model untuk tabel m_user yang sudah kita buat</p> <pre>config > auth.php 3 return [62 'providers' => [63 'users' => [64 'driver' => 'eloquent', 65 // 'model' => App\Models\User::class, 66 'model' => App\Models\UserModel::class, 67], 68],</pre>
2	<p>Selanjutnya kita modifikasi sedikit pada UserModel.php untuk bisa melakukan proses autentikasi</p>



	<pre>app > Models > UserModel.php > ... 32 33 // JS 7 PRAK 1 34 use Illuminate\Foundation\Auth\User as Authenticatable; // implementasi class autgenticatable 35 class UserModel extends Authenticatable { 36 use HasFactory; 37 38 protected \$table = 'm_user'; // koreksi tanda kutip 39 protected \$primaryKey = 'user_id'; // tambahkan tanda '=' 40 41 protected \$fillable = [42 'username', 43 'password', 44 'nama', // koreksi penulisan field 45 'level_id', 46 'created_at', 47 'updated_at' 48]; 49 50 protected \$hidden = ['password']; // jangan ditampilkan saat select 51 52 protected \$casts = [53 'password' => 'hashed' // casting password agar otomatis di-hash 54]; 55 56 // Relasi ke tabel level 57 public function level(): BelongsTo 58 { 59 return \$this->belongsTo(LevelModel::class, 'level_id', 'level_id'); 60 } 61 }</pre>
3	<p>Selanjutnya kita buat AuthController.php untuk memproses login yang akan kita lakukan</p> <pre>app > Models > AuthController.php > ... 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 use Illuminate\Support\Facades\Auth; 7 8 class AuthController extends Controller 9 { 10 public function login() 11 { 12 // Jika sudah login, maka redirect ke halaman home 13 if (Auth::check()) { 14 return redirect('/'); 15 } 16 17 return view('auth.login'); 18 } 19 20 public function postLogin(Request \$request) 21 { 22 if (\$request->ajax() \$request->wantsJson()) { 23 \$credentials = \$request->only('username', 'password'); 24 25 if (Auth::attempt(\$credentials)) { 26 return response()->json([27 'status' => true, 28 'message' => 'Login Berhasil', 29 'redirect' => url('/') 30]); 31 } 32 33 return response()->json([34 'status' => false, 35 'message' => 'Login Gagal' 36]); 37 } 38 39 return redirect('login'); 40 } 41 42 public function logout(Request \$request) 43 { 44 Auth::logout(); 45 \$request->session()->invalidate(); 46 \$request->session()->regenerateToken(); 47 48 return redirect('login'); 49 } 50 }</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)

```
resources > views > auth > login.blade.php > html > body.hold-transition.login-page > div.login-box > div.card.card-outline.card-primary > div.card-body > form#form-login >
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <title>Login Pengguna</title>
8 <!-- Google Font: Source Sans Pro -->
9 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
10 <!-- Font Awesome -->
11 <link rel="stylesheet" href="{{ asset('plugins/fontawesome-free/css/all.min.css') }}">
12 <!-- icheck bootstrap -->
13 <link rel="stylesheet" href="{{ asset('plugins/icheck-bootstrap/icheck-bootstrap.min.css') }}">
14 <!-- SweetAlert2 -->
15 <link rel="stylesheet" href="{{ asset('plugins/sweetalert2-theme-bootstrap-4/bootstrap-4.min.css') }}">
16 <!-- Theme style -->
17 <link rel="stylesheet" href="{{ asset('dist/css/adminlte.min.css') }}">
18 </head>
19
20 <body class="hold-transition login-page">
21 <div class="login-box">
22 <!-- /.login-logo -->
23 <div class="card card-outline card-primary">
24 <div class="card-header text-center">
25 <a href="{{ url('/') }}" class="h1"><b>Admin</b>LTE</a>
26 </div>
27 <div class="card-body">
28 <p class="login-box-msg">Sign in to start your session</p>
29 <form action="{{ url('login') }}" method="POST" id="form-login">
30 @csrf
31 <div class="input-group mb-3">
32 <input type="text" id="username" name="username" class="form-control" placeholder="Username">
33 <div class="input-group-append">
34 <div class="input-group-text">
35 <span class="fas fa-envelope"></span>
36 </div>
37 </div>
38 <small id="error-username" class="error-text text-danger"></small>
39 </div>
40 <div class="input-group mb-3">
41 <input type="password" id="password" name="password" class="form-control" placeholder="Password">
42 <div class="input-group-append">
43 <div class="input-group-text">
44 <span class="fas fa-lock"></span>
45 </div>
46 </div>
47 <small id="error-password" class="error-text text-danger"></small>
48 </div>
49 <div class="row">
50 <div class="col-8">
```



```
51 <div class="icheck-primary">
52   <input type="checkbox" id="remember">
53   <label for="remember">Remember Me</label>
54 </div>
55 </div>
56 <!-- /.col -->
57 <div class="col-4">
58   <button type="submit" class="btn btn-primary btn-block">Sign In</button>
59 </div>
60 <!-- /.col -->
61 </div>
62 </form>
63 </div>
64 <!-- /.card-body -->
65 </div>
66 <!-- /.card -->
67 </div>
68 <!-- /.login-box -->
69
70 <!-- jQuery -->
71 <script src="{{ asset('plugins/jquery/jquery.min.js') }}"></script>
72 <!-- Bootstrap 4 -->
73 <script src="{{ asset('plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
74 <!-- jquery-validation -->
75 <script src="{{ asset('plugins/jquery-validation/jquery.validate.min.js') }}"></script>
76 <script src="{{ asset('plugins/jquery-validation/additional-methods.min.js') }}"></script>
77 <!-- SweetAlert2 -->
78 <script src="{{ asset('plugins/sweetalert2/sweetalert2.min.js') }}"></script>
79 <!-- AdminLTE App -->
80 <script src="{{ asset('dist/js/adminlte.min.js') }}"></script>
81
82 <script>
83   $.ajaxSetup({
84     headers: {
85       'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
86     }
87   });
88
89   $(document).ready(function () {
90     $("#form-login").validate({
91       rules: {
92         username: {
93           required: true,
94           minlength: 4,
95           maxlength: 20
96         },
```



```
97     password: {
98         required: true,
99         minlength: 6,
100        maxlength: 20
101    },
102 },
103 submitHandler: function (form) { // ketika valid, maka bagian yg akan dijalankan
104     $.ajax({
105         url: form.action,
106         type: form.method,
107         data: $(form).serialize(),
108         success: function (response) {
109             if (response.status) { // jika sukses
110                 Swal.fire({
111                     icon: 'success',
112                     title: 'Berhasil',
113                     text: response.message,
114                 }).then(function () {
115                     window.location = response.redirect;
116                 });
117             } else { // jika error
118                 $('.error-text').text('');
119                 $.each(response.msgField, function (prefix, val) {
120                     $('#error-' + prefix).text(val[0]);
121                 });
122                 Swal.fire({
123                     icon: 'error',
124                     title: 'Terjadi Kesalahan',
125                     text: response.message
126                 });
127             }
128         }
129     });
130     return false;
131 },
132 errorElement: 'span',
133 errorPlacement: function (error, element) {
134     error.addClass('invalid-feedback');
135     element.closest('.input-group').append(error);
136 },
137 highlight: function (element, errorClass, validClass) {
138     $(element).addClass('is-invalid');
139 },
140 unhighlight: function (element, errorClass, validClass) {
141     $(element).removeClass('is-invalid');
142 }
143 });
```

```
144     });
145     </script>
146 </body>
147
148 </html>
149
```

5 Kemudian kita modifikasi route/web.php agar semua route masuk dalam auth



```
routes > web.php > ...
1  k?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
12 // Mendefinisikan pola untuk parameter id
13 Route::pattern('id', '[0-9]+');
14
15 // Route untuk halaman login
16 Route::get('login', [AuthController::class, 'login'])->name('login');
17 Route::post('login', [AuthController::class, 'postlogin']);
18 Route::post('logout', [AuthController::class, 'logout'])->middleware('auth');
19
20 // Route yang memerlukan autentikasi
21 Route::middleware(['auth'])->group(function () {
22     Route::get('/', [WelcomeController::class, 'index']);
23
24     // Route untuk manajemen user
25     Route::group(['prefix' => 'user', 'middleware' => 'authorize:ADM'], function () {
26         Route::get('/', [UserController::class, 'index']);
27         Route::post('/list', [UserController::class, 'list']);
28         Route::get('/create', [UserController::class, 'create']);
29         Route::post('/', [UserController::class, 'store']);
30
31         // AJAX routes for user management
32         Route::get('/create_ajax', [UserController::class, 'create_ajax']);
33         Route::post('/ajax', [UserController::class, 'store_ajax']);
34         Route::get('/{id}/edit_ajax', [UserController::class, 'edit_ajax']);
35         Route::put('/{id}/update_ajax', [UserController::class, 'update_ajax']);
36         Route::get('/{id}/delete_ajax', [UserController::class, 'confirm_ajax']);
37         Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']);
38
39         Route::get('/{id}', [UserController::class, 'show']);
40         Route::get('/{id}/edit', [UserController::class, 'edit']);
41         Route::put('/{id}', [UserController::class, 'update']);
42         Route::delete('/{id}', [UserController::class, 'destroy']);
43     });
44
45     // Route untuk manajemen level
```



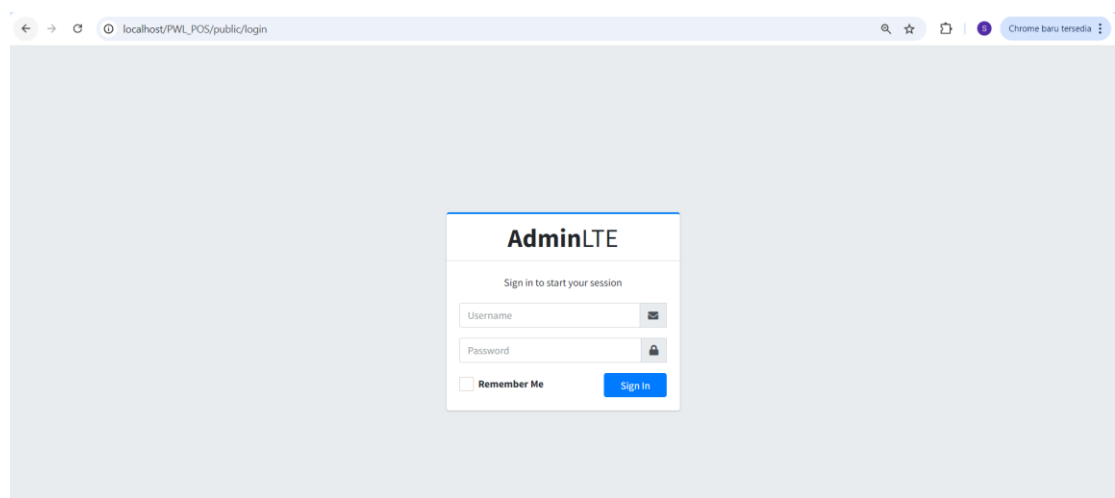
```
46 // Route::middleware(['authorize:ADM'])->group(function () { // JS 7 PRAK 2
47     Route::group(['prefix' => 'level', 'middleware' => 'authorize:ADM'], function () {
48         Route::get('/', [LevelController::class, 'index']);
49         Route::post('/list', [LevelController::class, 'list']);
50         Route::get('/create', [LevelController::class, 'create']);
51         Route::post('/', [LevelController::class, 'store']);
52
53         // AJAX routes for level management
54         Route::get('/create_ajax', [LevelController::class, 'create_ajax']);
55         Route::post('/ajax', [LevelController::class, 'store_ajax']);
56         Route::get('/{id}/edit_ajax', [LevelController::class, 'edit_ajax']);
57         Route::put('/{id}/update_ajax', [LevelController::class, 'update_ajax']);
58         Route::get('/{id}/delete_ajax', [LevelController::class, 'confirm_ajax']);
59         Route::delete('/{id}/delete_ajax', [LevelController::class, 'delete_ajax']);
60
61         Route::get('/{id}', [LevelController::class, 'show']);
62         Route::get('/{id}/edit', [LevelController::class, 'edit']);
63         Route::put('/{id}', [LevelController::class, 'update']);
64         Route::delete('/{id}', [LevelController::class, 'destroy']);
65     });
66 });
67
68 // Route untuk manajemen kategori
69 Route::group(['prefix' => 'kategori', 'middleware' => 'authorize:ADM,MNG'], function () {
70     Route::get('/', [KategoriController::class, 'index']);
71     Route::post('/list', [KategoriController::class, 'list']);
72     Route::get('/create', [KategoriController::class, 'create']);
73     Route::post('/', [KategoriController::class, 'store']);
74
75     // AJAX routes for kategori management
76     Route::get('/create_ajax', [KategoriController::class, 'create_ajax']);
77     Route::post('/ajax', [KategoriController::class, 'store_ajax']);
78     Route::get('/{id}/edit_ajax', [KategoriController::class, 'edit_ajax']);
79     Route::put('/{id}/update_ajax', [KategoriController::class, 'update_ajax']);
80     Route::get('/{id}/delete_ajax', [KategoriController::class, 'confirm_ajax']);
81     Route::delete('/{id}/delete_ajax', [KategoriController::class, 'delete_ajax']);
82
83     Route::get('/{id}', [KategoriController::class, 'show']);
84     Route::get('/{id}/edit', [KategoriController::class, 'edit']);
85     Route::put('/{id}', [KategoriController::class, 'update']);
86     Route::delete('/{id}', [KategoriController::class, 'destroy']);
87 });
88
89 // Route untuk manajemen barang
```



```
90 Route::group(['prefix' => 'barang', 'middleware' => 'authorize:ADM,MNG'], function () {
91     Route::get('/', [BarangController::class, 'index']);
92     Route::post('/list', [BarangController::class, 'list']);
93     Route::get('/create', [BarangController::class, 'create']);
94     Route::post('/', [BarangController::class, 'store']);
95
96     // AJAX routes for barang management
97     Route::get('/create_ajax', [BarangController::class, 'create_ajax']);
98     Route::post('/ajax', [BarangController::class, 'store_ajax']);
99     Route::get('/{id}/edit_ajax', [BarangController::class, 'edit_ajax']);
100    Route::put('/{id}/update_ajax', [BarangController::class, 'update_ajax']);
101    Route::get('/{id}/delete_ajax', [BarangController::class, 'confirm_ajax']);
102    Route::delete('/{id}/delete_ajax', [BarangController::class, 'delete_ajax']);
103
104    Route::get('/{id}', [BarangController::class, 'show']);
105    Route::get('/{id}/edit', [BarangController::class, 'edit']);
106    Route::put('/{id}', [BarangController::class, 'update']);
107    Route::delete('/{id}', [BarangController::class, 'destroy']);
108 });
109
110 // Route untuk manajemen supplier
111 Route::group(['prefix' => 'supplier', 'middleware' => 'authorize:ADM,MNG'], function () {
112     Route::get('/', [SupplierController::class, 'index']);
113     Route::post('/list', [SupplierController::class, 'list']);
114     Route::get('/create', [SupplierController::class, 'create']);
115     Route::post('/', [SupplierController::class, 'store']);
116
117     // AJAX routes for supplier management
118     Route::get('/create_ajax', [SupplierController::class, 'create_ajax']);
119     Route::post('/ajax', [SupplierController::class, 'store_ajax']);
120     Route::get('/{id}/edit_ajax', [SupplierController::class, 'edit_ajax']);
121     Route::put('/{id}/update_ajax', [SupplierController::class, 'update_ajax']);
122     Route::get('/{id}/delete_ajax', [SupplierController::class, 'confirm_ajax']);
123     Route::delete('/{id}/delete_ajax', [SupplierController::class, 'delete_ajax']);
124
125     Route::get('/{id}', [SupplierController::class, 'show']);
126     Route::get('/{id}/edit', [SupplierController::class, 'edit']);
127     Route::put('/{id}', [SupplierController::class, 'update']);
128     Route::delete('/{id}', [SupplierController::class, 'destroy']);
129 });
130
```

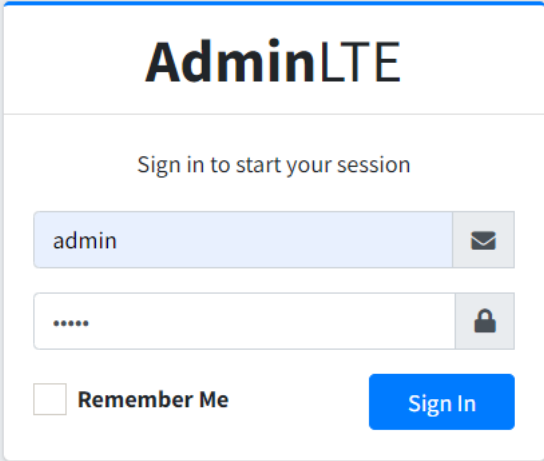
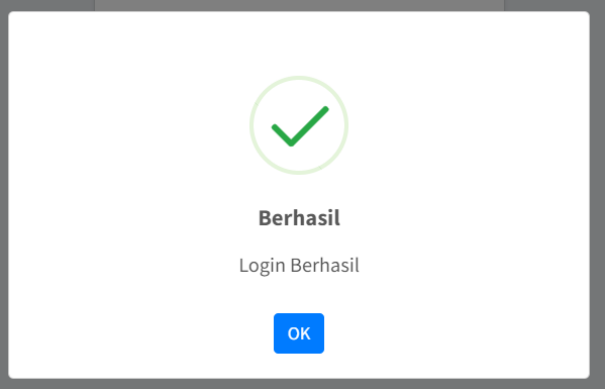
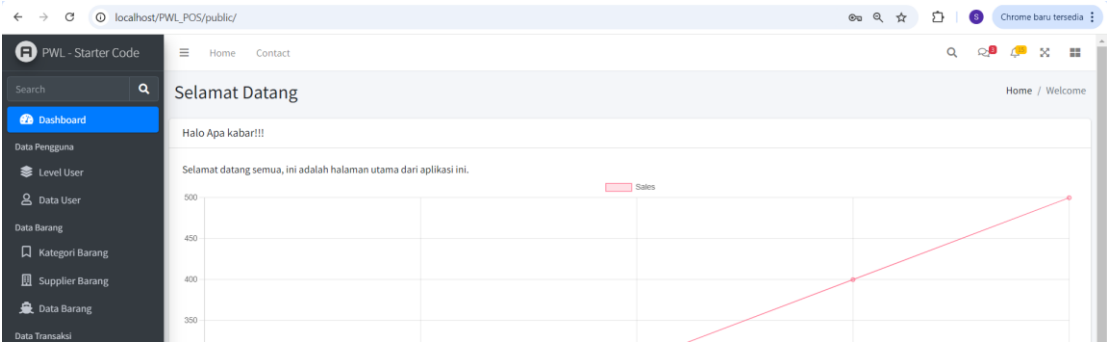
6

Ketika kita coba mengakses halaman localhost/PWL_POS/public makan akan tampil halaman awal untuk login ke aplikasi

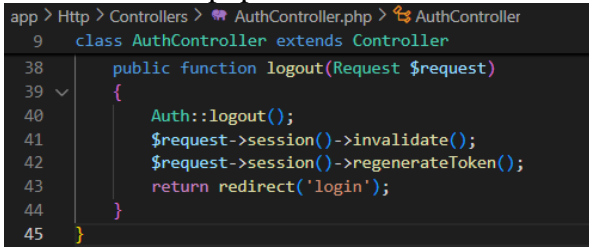
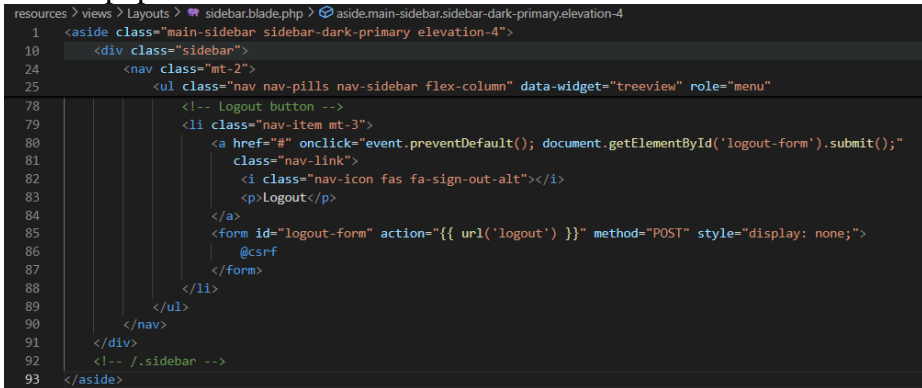
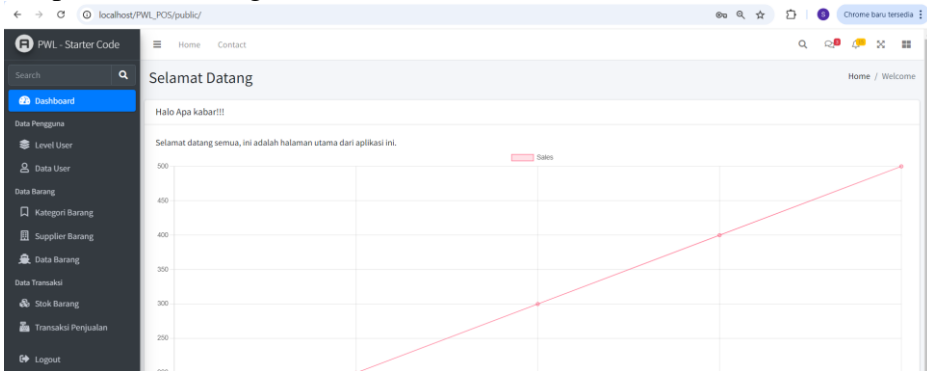




Tugas 1 – Implementasi Authentication :

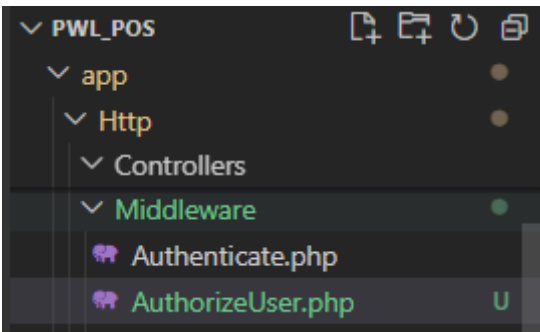
Langkah	Jawaban/Deskripsi
1	<p>Silahkan implementasikan proses login pada project kalian masing-masing</p>    <p>The first screenshot shows the AdminLTE login interface. It has a title 'AdminLTE' and a subtitle 'Sign in to start your session'. There are two input fields: one for the username 'admin' and another for a password represented by dots. Below the password field is a 'Remember Me' checkbox and a 'Sign In' button. The second screenshot shows a success message with a green checkmark icon, the text 'Berhasil' (Successful), 'Login Berhasil' (Login Successful), and an 'OK' button. The third screenshot shows the AdminLTE dashboard. It has a sidebar with a search bar and a list of menu items: Dashboard, Data Pengguna, Level User, Data User, Data Barang, Kategori Barang, Supplier Barang, Data Barang, and Data Transaksi. The main content area has a title 'Selamat Datang' (Welcome) and a message 'Halo Apa kabar!!!' (Hello, how are you!!!). It also says 'Selamat datang semua, ini adalah halaman utama dari aplikasi ini.' (Welcome everyone, this is the main page of the application). There is a line chart showing 'Sales' data.</p>
2	<p>Silahkan implementasi proses logout pada halaman web yang kalian buat</p> <ul style="list-style-type: none">- Web.php <pre>18 Route::post('logout', [AuthController::class, 'logout'])->middleware('auth');</pre>



	<ul style="list-style-type: none">- AuthController.php <pre>app > Http > Controllers > AuthController.php > AuthController 9 class AuthController extends Controller 38 39 { 40 public function logout(Request \$request) 41 { 42 Auth::logout(); 43 \$request->session()->invalidate(); 44 \$request->session()->regenerateToken(); 45 return redirect('login'); 46 } 47 }</pre>- sidebar.php <pre>resources > views > Layouts > sidebar.blade.php > aside.main-sidebar.sidebar-dark-primary.elevation-4 1 <aside class="main-sidebar sidebar-dark-primary elevation-4"> 2 <div class="sidebar"> 3 <nav class="mt-2"> 4 <ul class="nav nav-pills nav-sidebar flex-column" data-widget="treeview" role="menu" 5 > 6 <!-- Logout button --> 7 <li class="nav-item mt-3"> 8 10 <i class="nav-icon fas fa-sign-out-alt"></i> 11 <p>Logout</p> 12 13 <form id="logout-form" action="{{ url('logout') }}" method="POST" style="display: none;" 14 @csrf 15 </form> 16 17 18 </nav> 19 </div> 20 </aside></pre>- tampilan tombol logout 
3	<p>Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan jawab</p> <p>Untuk memperbarui tampilan login dan menambahkan fitur logout, beberapa file perlu diedit. Di config/auth.php dan UserModel.php, pastikan autentikasi dan sesi pengguna berjalan dengan baik. Di AuthController.php, tambahkan logika login dan logout. Ubah tampilan di auth/login.blade.php agar sesuai desain, dan tambahkan routing di route/web.php. Untuk logout, modifikasi AuthController.php dan tambahkan link logout di sidebar.php, serta pastikan routing logout ada di Web.php.</p>
4	<p>Submit kode untuk impementasi Authentication pada repository github kalian.</p>



Praktikum 2 – Implementasi Authorizaton di Laravel dengan Middleware

Langkah	Jawaban/Deskripsi
1	<p>Kita modifikasi UserModel.php dengan menambahkan kode berikut</p> <pre>app > Models > UserModel.php > UserModel 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; 9 10 class UserModel extends Authenticatable 11 { 12 use HasFactory; 13 14 protected \$table = 'm_user'; //Mendefinisikan nama tabel yang digunakan oleh model ini 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 protected \$hidden = ['password']; 18 protected \$casts= ['password'=> 'hashed']; 19 20 public function level(): BelongsTo{ 21 return \$this->belongsTo(LevelModel::class, 'level_id', 'level_id'); 22 } 23 24 public function getRoleName(): String{ 25 return \$this->level->level_nama; 26 } 27 28 public function hasRole(\$role): bool{ 29 return \$this->level->level_kode == \$role; 30 } 31 }</pre>
2	<p>Kemudian kita buat middleware dengan nama AuthorizeUser.php. Kita bisa buat middleware dengan mengetikkan perintah pada terminal/CMD</p> <p>php artisan make:middleware AuthorizeUser</p> <p>File middleware akan dibuat di app/Http/Middleware/AuthorizeUser.php</p> 
3	<p>Kemudian kita edit middleware AuthorizeUser.php untuk bisa mengecek apakah pengguna yang mengakses memiliki Level/Role/Group yang sesuai</p>



```
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, $role = ''): Response
17     {
18         // Mengambil pengguna yang sedang login dari permintaan
19         $user = $request->user();
20
21         // Periksa apakah pengguna memiliki role yang diberikan
22         if ($user->hasRole($role)) {
23             // Jika memiliki role, lanjutkan ke permintaan berikutnya
24             return $next($request);
25         }
26
27         // Jika tidak memiliki role, tampilkan error 403
28         abort(403, 'Forbidden. Kamu tidak punya akses ke halaman ini');
29     }
30 }
```

4 Kita daftarkan ke app/Http/Kernel.php untuk middleware yang kita buat barusan

```
app > Http > Kernel.php > Kernel
7  class Kernel extends HttpKernel
55     protected $middlewareAliases = [
56         'auth' => \App\Http\Middleware\Authenticate::class,
57         'authorize' => \App\Http\Middleware\AuthorizeUser::class, // middleware yg kita buat
58         'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
```

5 Sekarang kita perhatikan tabel m_level yang menjadi tabel untuk menyimpan level/group/role dari user ada

localhost/phpmyadmin/index.php?route=/sql&db=pwl_pos&table=m_level

Server: localhost:3306 Database: pwl_pos Table: m_level

Showing rows 0 - 3 (4 total, Query took 0.0008 seconds)

SELECT * FROM 'm_level'

Number of rows: 25 Filter rows: Search this table Sort by key: None

	level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	MNG	Manager	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	STF	StaffKasir	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	CST	Customer	NULL	NULL

Check all With selected: Edit Copy Delete Export



6

Untuk mencoba authorization yang telah kita buat, maka perlu kita modifikasi route/web.php untuk menentukan route mana saja yang akan diberi hak akses sesuai dengan level user

```
routes > web.php > Closure
21 Route::middleware(['auth'])->group(function () {
45     // Route untuk manajemen level
46     Route::middleware(['authorize:ADM'])->group(function () { // JS 7 PRAK 2
47         Route::group(['prefix' => 'level', 'middleware' => 'authorize:ADM'], function () {
48             Route::get('/', [LevelController::class, 'index']);
49             Route::post('/list', [LevelController::class, 'list']);
50             Route::get('/create', [LevelController::class, 'create']);
51             Route::post('/', [LevelController::class, 'store']);
52
53             // AJAX routes for level management
54             Route::get('/create_ajax', [LevelController::class, 'create_ajax']);
55             Route::post('/ajax', [LevelController::class, 'store_ajax']);
56             Route::get('/{id}/edit_ajax', [LevelController::class, 'edit_ajax']);
57             Route::put('/{id}/update_ajax', [LevelController::class, 'update_ajax']);
58             Route::get('/{id}/delete_ajax', [LevelController::class, 'confirm_ajax']);
59             Route::delete('/{id}/delete_ajax', [LevelController::class, 'delete_ajax']);
60
61             Route::get('/{id}', [LevelController::class, 'show']);
62             Route::get('/{id}/edit', [LevelController::class, 'edit']);
63             Route::put('/{id}', [LevelController::class, 'update']);
64             Route::delete('/{id}', [LevelController::class, 'destroy']);
65         });
66     });
67 }
```

Pada kode yang ditandai merah, terdapat authorize:ADM . Kode ADM adalah nilai dari level_kode pada tabel m_level. Yang artinya, user yang bisa mengakses route untuk manage data level, adalah user yang memiliki level sebagai Administrator.

7

Untuk membuktikannya, sekarang kita coba login menggunakan akun selain level administrator, dan kita akses route menu level tersebut

AdminLTE

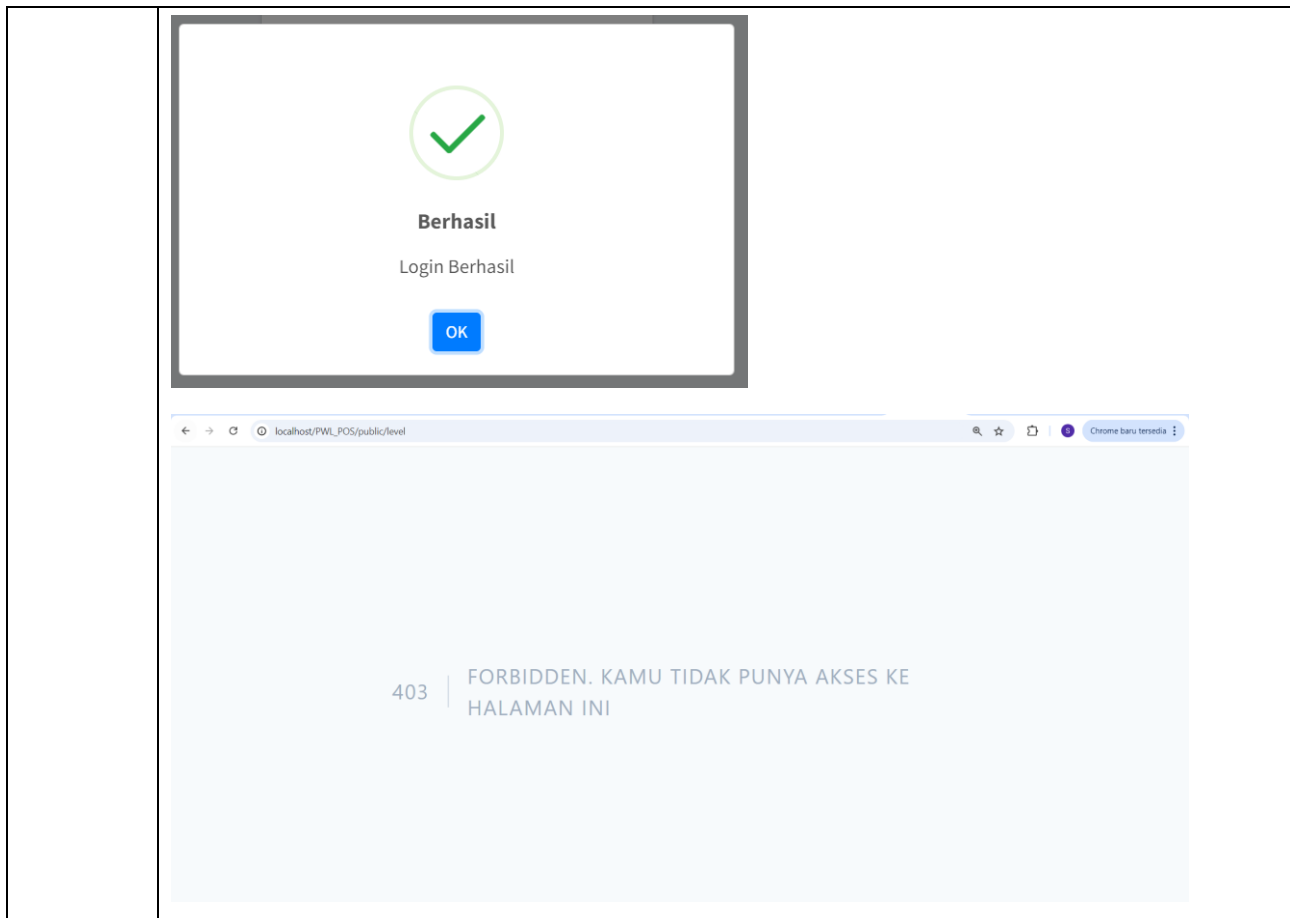
Sign in to start your session

Manager

.....

☐ Remember Me

Sign In



Tugas 2 – Implementasi Authoriization

Langkah	Jawaban/Deskripsi
1	<p>Apa yang kalian pahami pada praktikum 2 ini?</p> <p>jawab</p> <p>membuat sistem yang hanya membolehkan pengguna tertentu (misalnya admin) untuk mengakses halaman atau fitur tertentu di aplikasi Laravel.</p>
2	<p>Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan</p> <p>jawab</p> <p>Untuk membatasi akses berdasarkan level pengguna, pertama tambahkan logika di UserModel.php untuk memeriksa level atau peran (role) pengguna. Lalu, buat middleware baru dengan perintah php artisan make:middleware AuthorizeUser. Di AuthorizeUser.php, tambahkan logika untuk memeriksa apakah pengguna memiliki</p>



	level yang sesuai, misalnya "Administrator". Jika tidak, akses akan ditolak. Daftarkan middleware di Kernel.php agar bisa digunakan. Buat tabel m_level untuk menyimpan informasi level pengguna. Terakhir, di route/web.php, tambahkan middleware ke rute yang hanya boleh diakses oleh pengguna dengan level tertentu, seperti "ADM".
3	Submit kode untuk impementasi Authorization pada repository github kalian.

Praktikum 3 – Implementasi Multi-Level Authorizaton di Laravel dengan Middleware

Langkah	Jawaban/Deskripsi
1	<p>Kita modifikasi UserModel.php untuk mendapatkan level_kode dari user yang sudah login. Jadi kita buat fungsi dengan nama getRole()</p> <pre>app > Models > UserModel.php > UserModel 10 class UserModel extends Authenticatable 30 public function getRole(){ 31 return \$this->level->level_kode; 32 } 33 }</pre>
2	<p>Selanjutnya, Kita modifikasi middleware AuthorizeUser.php dengan kode berikut</p> <pre>app > Http > Middleware > AuthorizeUser.php > AuthorizeUser 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 \$user_role = \$request->user()->getRole(); //ambil data level_kode dari user yang login 19 if(in_array(\$user_role, \$roles)){ //cek apakah level_kode ada di dalam array roles 20 return \$next(\$request); //jika ada, maka lanjutkan request 21 } 22 // jika tidak punya role, maka tampilkan error 403 23 abort(403, 'Forbidden. Kamu tidak punya akses ke halaman ini'); 24 } 25 }</pre>
3	<p>Setelah itu tinggal kita perbaiki route/web.php sesuaikan dengan role/level yang diinginkan.</p> <pre>routes > web.php > Closure > Closure 21 Route::middleware(['auth'])->group(function () { 68 // Route untuk manajemen kategori 69 Route::group(['prefix' => 'kategori', 'middleware' => 'authorize:ADM,MNG'], function () { //JS 7 PRAK 3 70 Route::get('/', [KategoriController::class, 'index']);</pre>



4

Sekarang kita sudah bisa memberikan hak akses menu/route ke beberapa level user

- Tampilan fitur data barang saat diakses oleh admin dan manager

```
routes > web.php > Closure > Closure
21 Route::middleware(['auth'])->group(function () {
89     // Route untuk manajemen barang
90     Route::group(['prefix' => 'barang', 'middleware' => 'authorize:ADM,MNG'], function () { //JS 7 PRAK 3
91         Route::get('/', [BarangController::class, 'index']);
92         Route::post('/list', [BarangController::class, 'list']);
    });
});

routes > web.php > Closure > Closure
21 Route::middleware(['auth'])->group(function () {
110    // Route untuk manajemen supplier
111    Route::group(['prefix' => 'supplier', 'middleware' => 'authorize:ADM,MNG'], function () { //JS 7 PRAK 3
112        Route::get('/', [SupplierController::class, 'index']);
    });
});
```

The screenshot displays two views of the PWL - Starter Code application. The top view shows the 'Daftar kategori barang' (List of goods categories) screen, which includes a sidebar menu with options like Dashboard, Data Pengguna, Level User, Data User, Data Barang, Kategori Barang, Supplier Barang, Data Barang, Data Transaksi, Stok Barang, Transaksi Penjualan, and Logout. The main content area shows a table of categories with columns for ID, Kode Kategori, Nama Kategori, and Aksi. The bottom view shows the 'Daftar supplier' (List of suppliers) screen, which has a similar sidebar and a table of suppliers with columns for ID, Kode Supplier, Nama Supplier, Alamat Supplier, and Aksi.

ID	Kode Kategori	Nama Kategori	Aksi
1	ELEC	Elektronik	Detail Edit Hapus
2	FURN	Furniture	Detail Edit Hapus
3	FOOD	Makanan	Detail Edit Hapus
4	CLOT	Pakaian	Detail Edit Hapus
5	BOOK	Buku	Detail Edit Hapus

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	SUP001	Supplier A	Jl. Meangga No. 1	Detail Edit Hapus
2	SUP002	Supplier B	Jl. Pepaya No. 2	Detail Edit Hapus
3	SUP003	Supplier C	Jl. Timun No. 3	Detail Edit Hapus



localhost/PWL_POS/public/barang

PWL - Starter Code

Home Contact

Daftar Barang

Daftar Barang yang terdaftar dalam sistem

Filter: - Semua -

Kategori Barang

Show 10 entries

ID	Kategori Barang	Kode	Nama	Harga Beli	Harga Jual	Aksi
1	Furniture	BRG1	Barang 1	40951	90560	Detail Edit Hapus
2	Furniture	BRG2	Barang 2	16217	68037	Detail Edit Hapus
3	Furniture	BRG3	Barang 3	27940	93297	Detail Edit Hapus
4	Pakaian	BRG4	Barang 4	13874	93160	Detail Edit Hapus
5	Furniture	BRG5	Barang 5	47666	98674	Detail Edit Hapus

Search:

- Tampilan fitur data barang saat diakses oleh staff

AdminLTE

Sign in to start your session

Staff

.....

☐ Remember Me

Sign In

Berhasil

Login Berhasil

OK

403 | FORBIDDEN. KAMU TIDAK PUNYA AKSES KE HALAMAN INI



Tugas 3 – Implementasi Multi-Level Authorization :

Langkah	Jawaban/Deskripsi
1	Silahkan implementasikan multi-level authorization pada project kalian masing-masing
2	<p>Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan jawab</p> <p>Menambahkan akses untuk 3 user (admin manager dan staff) pada stok dan penjualan pada web.php</p> <pre>routes > web.php > Closure > Closure 24 Route::middleware(['auth'])->group(function () { 134 // Route untuk manajemen stok 135 Route::group(['prefix'=> 'stok','middleware'=> 'authorize:ADM,MNG,STF'], function() { 136 Route::get('/', [StokController::class, 'index']); //menampilkan halaman awal Stok 137 Route::post('/list', [StokController::class, 'list']); //menampilkan data stok dalam b</pre>



3

Implementasikan multi-level authorization untuk semua Level/Jenis User dan Menu yang sesuai dengan Level/Jenis User

jawab

Untuk implementasi multi-level authorization, setiap user role seperti admin, manager, dan staff memiliki akses yang dibatasi sesuai dengan tugas mereka. Admin memiliki akses penuh terhadap manajemen user, level, kategori, barang, supplier, stok, dan penjualan. Manager memiliki akses terhadap manajemen kategori, barang, supplier, stok, dan penjualan. Sementara itu, staff hanya memiliki akses untuk mengelola stok dan penjualan. Semua ini diatur dalam rute web.php dengan middleware yang memeriksa role user, dan dipastikan hanya role yang sesuai yang bisa mengakses fitur yang ditentukan.

- Tampilan fitur stok saat diakses oleh staff

ID	Barang	Supplier	User	Tanggal	Jumlah	Aksi
1	Barang 1	Supplier B	Administrator	2024-09-12 05:57:24	90	Detail Edit Hapus
2	Barang 2	Supplier C	Administrator	2024-08-13 05:57:24	80	Detail Edit Hapus
3	Barang 3	Supplier A	Administrator	2024-07-15 05:57:24	89	Detail Edit Hapus
4	Barang 4	Supplier B	Administrator	2024-06-15 05:57:24	91	Detail Edit Hapus
5	Barang 5	Supplier C	Administrator	2024-05-19 05:57:24	49	Detail Edit Hapus
6	Barang 6	Supplier A	Administrator	2024-05-08 05:57:24	95	Detail Edit Hapus
7	Barang 7	Supplier B	Administrator	2024-04-24 05:57:24	34	Detail Edit Hapus

- Tampilan fitur penjualan saat diakses oleh staff



	<p>- Tampilan fitur selain stok dan penjualan saat diakses oleh staff</p>
4	Submit kode untuk impementasi Authorization pada repository github kalian.

Tugas 4 – Implementasi Form Registrasi

Langkah	Jawaban/Deskripsi
1	Silahkan implementasikan form untuk registrasi user. AuthController.php auth/login.blade.php auth/register.blade.php route/web.php
2	Screenshot hasil yang kalian kerjakan



- Form register

The first screenshot shows the login page of the ProjectPWL application. It has a title 'ProjectPWL' and a subtitle 'Sign in to start your session'. There are input fields for 'Username' and 'Password', a 'Remember Me' checkbox, and a 'Sign in' button. A link 'Don't have an account? Sign up' is also present.

The second screenshot shows the registration page of the AdminLTE application. It has a title 'AdminLTE' and a subtitle 'Sign up to create your account'. There are input fields for 'Level' (set to Administrator), 'Username' (Toni23), 'Nama' (Toni), and 'Password'. There are 'Kembali' and 'Sign Up' buttons.

The third screenshot shows the same registration page as the second, but with a success message overlay. The message says 'Berhasil Registrasi berhasil dilakukan!' and has an 'OK' button.

- Maka data user baru tersimpan di database



localhost/phpmyadmin/index.php?route=/sql&pos=0&db=pwl_pos&table=m_user

Server: localhost:3306 Database: pwl_pos Table: m_user

Showing rows 0 - 10 (11 total, Query took 0.0026 seconds)

SELECT * FROM `m_user`

Number of rows: 25 Filter rows: Search this table Sort by key: None

	user_id	level_id	username	nama	password	created_at	updated_at
<input type="checkbox"/>	1	1	admin	Administrator	\$2y\$12\$KeoQ2dwaUZKLusYRjXPPuP8JS2dyNtbEGPgcjW2...	NULL	2024-10-04 07:02:07
<input type="checkbox"/>	2	2	manager	Manager	\$2y\$12\$MeZVnTghDgKVFEDQws4NO9RwayZuB.8W3M5.k93s...	NULL	NULL
<input type="checkbox"/>	3	3	staff	Staff Kasir	\$2y\$12\$SaQlmmTZn/iHsdGh.6MOJO1nvgSVvPQWVEXDsUJfAY...	NULL	NULL
<input type="checkbox"/>	7	4	customer-1	Pelanggan Pertama	\$2y\$12\$Xg6rBID07c638why6m75eX82T0FCj9XhKGe8O9YL...	NULL	2024-09-14 09:01:28
<input type="checkbox"/>	11	2	manager_dua	Manager 2	\$2y\$12\$2wTkbJiw9dTSpnkW4GvruWgAHnXdxPTFTreUod...	2024-09-19 17:25:05	2024-09-19 17:25:05
<input type="checkbox"/>	12	2	manager22	Manager Dua Dua	\$2y\$12\$il13ZJpxk1BslMwGfM0oD7UNJzUj3FSEKXvIM...	2024-09-20 05:39:56	2024-09-20 05:39:56
<input type="checkbox"/>	15	2	manager33	Manager Tiga Tiga	\$2y\$12\$6SvD6Z7.BSS.84XYgbAejumqPzn9PadiorQHeaCYBd...	2024-09-20 07:28:32	2024-09-20 07:28:32
<input type="checkbox"/>	16	2	manager56	Manager 55	\$2y\$12\$Wn8pmpQPdgm0HfLjz5PhtzuEj7YrotuARmx2WNUJNGM...	2024-09-20 07:39:38	2024-09-20 07:39:38
<input type="checkbox"/>	19	2	manager12	Manager 11	\$2y\$12\$PbF46szmHt644TVcTR0eKpv0kbQ0m/YZ/1LMyyVh...	2024-09-20 07:51:42	2024-09-20 07:51:42
<input type="checkbox"/>	20	2	manager11	Manager 11	\$2y\$12\$aldDjk1B0NXqvlxixBuFId.gEMXsgUgM3MBH4b...	2024-09-20 07:53:43	2024-09-20 07:53:43
<input type="checkbox"/>	31	1	Sitii	Siti	\$2y\$12\$AmsO3SnhWCqzSIHg6VGHbU4shUJJe7Bdhj2gaVWQJ3T...	2024-10-16 04:47:14	2024-10-16 04:47:14

- Tampilan setelah kembali ke halaman login

localhost/PWL_POS/public/login

ProjectPWL

Sign in to start your session

☐ Remember Me

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

- Tampilan setelah login



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>

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