



Mata Kuliah : Pemrograman Web Lanjut (PWL)  
Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis  
Semester : 4 (empat) / 5 (lima)  
Pertemuan ke- : 7 (tujuh)

## JOBSHEET 07

### Authentication dan Authorization di Laravel

Laravel Authentication dipergunakan untuk memproteksi halaman atau fitur dari web yang hanya diakses oleh orang tertentu yang diberikan hak. Fitur seperti ini biasanya ditemui di sistem yang memiliki fitur administrator atau sistem yang memiliki pengguna yang boleh menambahkan datanya.

Laravel membuat penerapan otentikasi sangat sederhana dan telah menyediakan berbagai fitur yang dapat dimanfaatkan tanpa perlu melakukan penambahan instalasi modul tertentu. File konfigurasi otentikasi terletak di `config / auth.php`, yang berisi beberapa opsi yang terdokumentasi dengan baik untuk mengubah konfigurasi dari layanan otentikasi.

Pada intinya, fasilitas otentikasi Laravel terdiri dari “*guards*” dan “*providers*”. *Guards* menentukan bagaimana pengguna diautentikasi untuk setiap permintaan. Misalnya, Laravel mengirim dengan *guards* untuk sesi dengan menggunakan penyimpanan session dan cookie.

#### Middleware

**Middleware** adalah lapisan perantara antara permintaan **route HTTP** yang masuk dan **action** dari Controller yang akan dijalankan. **Middleware** memungkinkan kita untuk melakukan berbagai tugas baik itu sebelum ataupun sesudah tindakan dilakukan. Kita juga dapat menggunakan **tool CLI** untuk membuat sebuah **Middleware** dalam **Laravel**. Beberapa contoh penggunaan **Middleware** meliputi autentikasi, validasi, manipulasi permintaan, dan lainnya. Berikut di bawah ini adalah manfaat dari **Middleware** :

- **Keamanan** : dalam **Middleware** memungkinkan kita untuk memverifikasi apakah pengguna sudah diautentikasi sebelum mengakses halaman tertentu. Dengan demikian, kita dapat melindungi data sensitif dan mengontrol hak akses pengguna.
- **Pemfilteran Data** : **Middleware** dapat digunakan untuk memanipulasi data permintaan sebelum sebuah **action** dalam **controller** dilakukan. Misalnya, kita dapat memeriksa terlebih dahulu data yang dikirim oleh pengguna sebelum data tersebut diproses lebih



lanjut atau kita ingin memodifikasi data yang akan dikirim lalu kita dapat memeriksa ulang data yang akan dikirim oleh pengguna sebelum data tersebut diproses.

- **Logging dan Audit : Middleware** juga dapat digunakan untuk mencatat aktivitas pengguna atau melakukan audit terhadap permintaan yang masuk. Ini dapat membantu dalam pemantauan dan analisis aplikasi.

### INFO

Kita akan menggunakan Laravel Auth secara manual seperti  
<https://laravel.com/docs/10.x/authentication#authenticating-users>

Sesuai dengan **Studi Kasus PWL.pdf**.

Jadi project Laravel 10 kita masih sama dengan menggunakan repositori **PWL\_POS**.

*Project PWL\_POS* akan kita gunakan sampai pertemuan 12 nanti, sebagai project yang akan kita pelajari

## A. Implementasi Manual Authentication di Laravel

Autentikasi adalah proses untuk memverifikasi identitas pengguna yang mencoba mengakses sistem. Dalam konteks aplikasi web, autentikasi memastikan bahwa pengguna yang mencoba login memiliki hak akses yang sesuai berdasarkan kredensial seperti email dan password. Proses autentikasi berbeda dengan **otorisasi**, yang merupakan langkah lanjutan untuk menentukan hak akses apa yang dimiliki pengguna setelah mereka berhasil diautentikasi.

### Konsep Autentikasi di Laravel

Laravel menawarkan sistem autentikasi yang sangat fleksibel. Laravel menyediakan mekanisme autentikasi bawaan melalui layanan authentication scaffolding seperti Laravel *Jetstream* dan *Breeze*, yang dapat secara otomatis menghasilkan halaman dan logika autentikasi. Namun, terkadang pengembang memerlukan implementasi autentikasi yang lebih manual untuk memberikan kontrol penuh terhadap setiap aspek dari proses tersebut.

Beberapa komponen penting dalam sistem autentikasi Laravel meliputi:

- *Guard*: Komponen yang mengatur bagaimana pengguna diautentikasi untuk setiap permintaan. *Guard* default menggunakan sesi dan cookie.



- *Provider*: Mengatur bagaimana pengguna diambil dari database atau sumber data lainnya. *Provider* default mengambil data pengguna dari database dengan menggunakan Eloquent ORM.
- *Session*: Laravel menggunakan sesi untuk menyimpan status autentikasi pengguna. Sesi memungkinkan sistem untuk mengingat pengguna yang sudah login di antara permintaan HTTP yang berbeda.

Alur umum dari autentikasi meliputi:

1. *Login*: Pengguna mengirimkan kredensial (biasanya berupa email dan password).
2. *Verifikasi Kredensial*: Sistem memeriksa apakah kredensial yang diberikan sesuai dengan data di database.
3. *Pembuatan Sesi*: Jika kredensial benar, sistem akan membuat sesi untuk pengguna yang akan disimpan di server.
4. *Akses ke Halaman yang Dilindungi*: Pengguna yang terautentikasi dapat mengakses halaman-halaman yang dilindungi oleh *middleware* auth.
5. *Logout*: Pengguna bisa keluar dari sistem dan sesi mereka akan dihapus.

### Middleware Autentikasi

Middleware auth di Laravel digunakan untuk melindungi rute atau halaman agar hanya dapat diakses oleh pengguna yang sudah terautentikasi. Jika pengguna mencoba mengakses rute yang memerlukan autentikasi tanpa login, mereka akan diarahkan ke halaman login.

- **Guard** bertanggung jawab untuk menangani proses autentikasi pengguna. Laravel secara default menggunakan *guard* berbasis sesi untuk autentikasi web, namun juga mendukung *guard* berbasis token (seperti API).
- **Provider** bertugas untuk mengambil pengguna dari database. Laravel menyediakan *provider* default yang menggunakan Eloquent, namun juga mendukung *provider* lain seperti Query Builder.

### Implementasi di Laravel 10

Kita akan menerapkan penggunaan authentication di Laravel. Dalam penerapan ini, kita akan mencoba membuat otentikasi secara di Laravel, agar kita paham langkah-langkah dalam membuat Authentication



## Praktikum 1 – Implementasi Authentication :

1. Kita buka project laravel **PWL\_POS** kita, dan kita modifikasi konfigurasi aplikasi kita di **config/auth.php**

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

Pada bagian ini kita sesuaikan dengan Model untuk tabel **m\_user** yang sudah kita buat

```
62     'providers' => [  
63         'users' => [  
64             'driver' => 'eloquent',  
65             'model' => App\Models\UserModel::class,  
66         ],  
67     ],
```

2. Selanjutnya kita modifikasi sedikit pada **UserModel.php** untuk bisa melakukan proses autentikasi

```
<?php  
namespace App\Models;  
  
use Illuminate\Database\Eloquent\Model;  
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Relations\BelongsTo;  
use Illuminate\Foundation\Auth\User as Authenticatable; // implementasi class Authenticatable  
  
class UserModel extends Authenticatable  
{  
    use HasFactory;  
  
    protected $table = 'm_user';  
    protected $primaryKey = 'user_id';  
    protected $fillable = ['username', 'password', 'nama', 'level_id', 'created_at', 'updated_at'];  
  
    protected $hidden = ['password']; // jangan di tampilkan saat select  
  
    protected $casts = ['password' => 'hashed']; // casting password agar otomatis di hash  
  
    /**  
     * Relasi ke tabel level  
     */  
    public function level(): BelongsTo  
    {  
        return $this->belongsTo(LevelModel::class, 'level_id', 'level_id');  
    }  
}
```



3. Selanjutnya kita buat `AuthController.php` untuk memproses login yang akan kita lakukan

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class AuthController extends Controller
{
    public function login()
    {
        if(Auth::check()){ // jika sudah login, maka redirect ke halaman home
            return redirect('/');
        }
        return view('auth.login');
    }

    public function postlogin(Request $request)
    {
        if($request->ajax() || $request->wantsJson()){
            $credentials = $request->only('username', 'password');

            if (Auth::attempt($credentials)) {
                return response()->json([
                    'status' => true,
                    'message' => 'Login Berhasil',
                    'redirect' => url('/')
                ]);
            }

            return response()->json([
                'status' => false,
                'message' => 'Login Gagal'
            ]);
        }

        return redirect('login');
    }

    public function logout(Request $request)
    {
        Auth::logout();

        $request->session()->invalidate();
        $request->session()->regenerateToken();
        return redirect('login');
    }
}
```

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=fallback">
<!-- Font Awesome -->
<link rel="stylesheet" href="{ asset('plugins/fontawesome-free/css/all.min.css') }}">
<!-- icheck bootstrap -->
<link rel="stylesheet" href="{ asset('plugins/icheck-bootstrap/icheck-bootstrap.min.css')
}}">
<!-- SweetAlert2 -->
<link rel="stylesheet" href="{ asset('plugins/sweetalert2-theme-bootstrap-4/bootstrap-
4.min.css') }}">
<!-- Theme style -->
<link rel="stylesheet" href="{ asset('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>
          <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>
          <!-- /.col -->
          <div class="col-4">
            <button type="submit" class="btn btn-primary btn-block">Sign In</button>
          </div>
          <!-- /.col -->
        </div>
      </form>
    </div>
    <!-- /.card-body -->
  </div>
  <!-- /.card -->
</div>
<!-- /.login-box -->

<!-- jQuery -->
```



```
<script src="{{ asset('plugins/jquery/jquery.min.js') }}"></script>
<!-- Bootstrap 4 -->
<script src="{{ asset('plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
<!-- jquery-validation -->
<script src="{{ asset('plugins/jquery-validation/jquery.validate.min.js') }}"></script>
<script src="{{ asset('plugins/jquery-validation/additional-methods.min.js') }}"></script>
<!-- SweetAlert2 -->
<script src="{{ asset('plugins/sweetalert2/sweetalert2.min.js') }}"></script>
<!-- AdminLTE App -->
<script src="{{ asset('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: 6, 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
            });
          }
        }
      });
    },
    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>
```



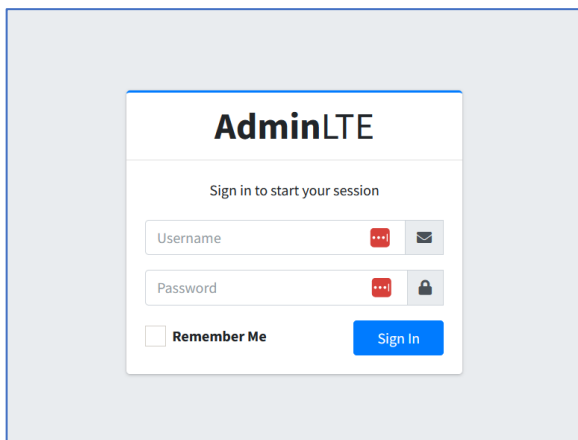


```
</body>  
</html>
```

5. Kemudian kita modifikasi [route/web.php](#) agar semua route masuk dalam auth

```
<?php  
  
use App\Http\Controllers\UserController;  
use App\Http\Controllers\SupplierController;  
use App\Http\Controllers\BarangController;  
use App\Http\Controllers\AuthController;  
use App\Http\Controllers\KategoriController;  
use App\Http\Controllers\LevelController;  
use App\Http\Controllers>WelcomeController;  
use Illuminate\Support\Facades\Route;  
  
Route::pattern('id','[0-9]+'); // artinya ketika ada parameter {id}, maka harus berupa angka  
  
Route::get('login', [AuthController::class,'login'])->name('login');  
Route::post('login', [AuthController::class,'postlogin']);  
Route::get('logout', [AuthController::class,'logout'])->middleware('auth');  
  
Route::middleware(['auth'])->group(function(){ // artinya semua route di dalam group ini harus login dulu  
    // masukkan semua route yang perlu autentikasi di sini  
});
```

6. Ketika kita coba mengakses halaman [localhost/PWL\\_POS/public](localhost/PWL_POS/public) maka akan tampil halaman awal untuk login ke aplikasi



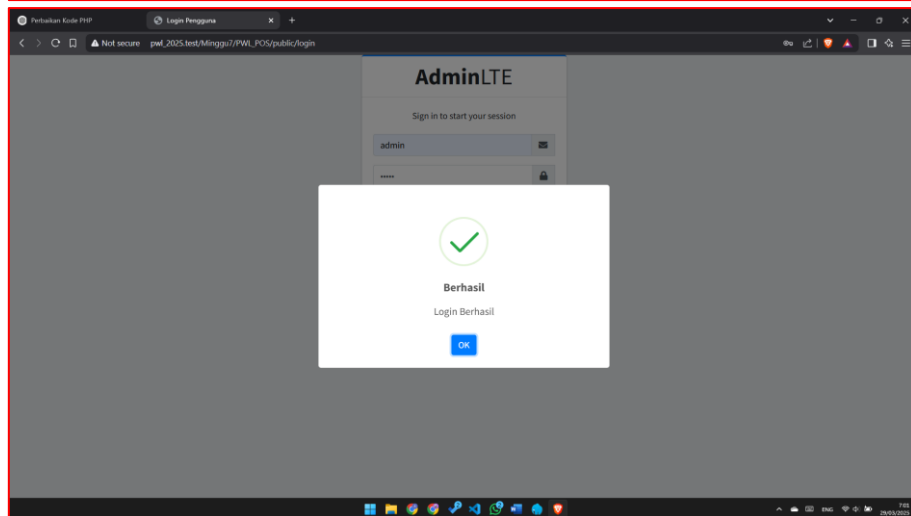
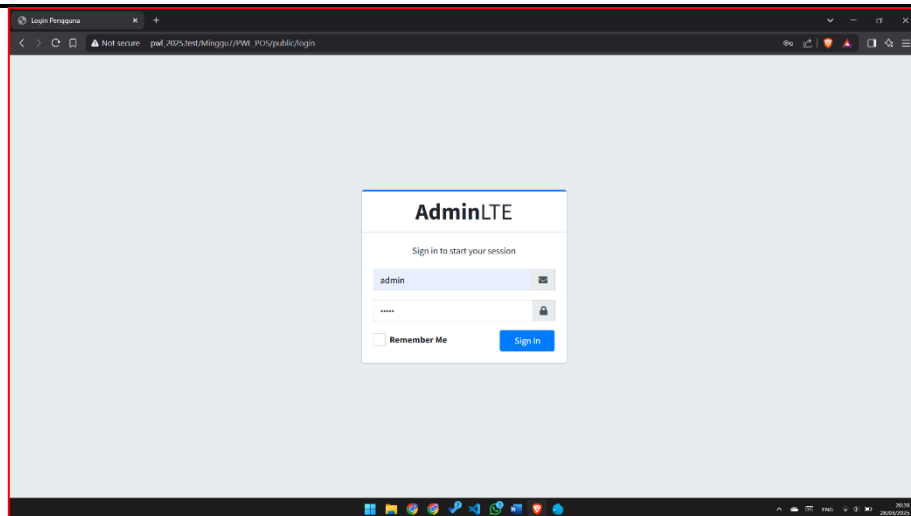
## Tugas 1 – Implementasi Authentication :

1. Silahkan implementasikan proses login pada project kalian masing-masing
  - Berhasil

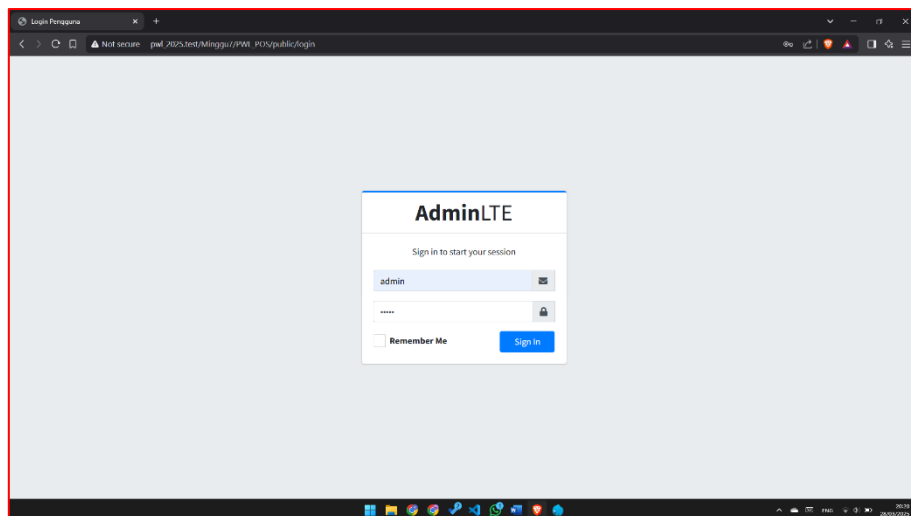


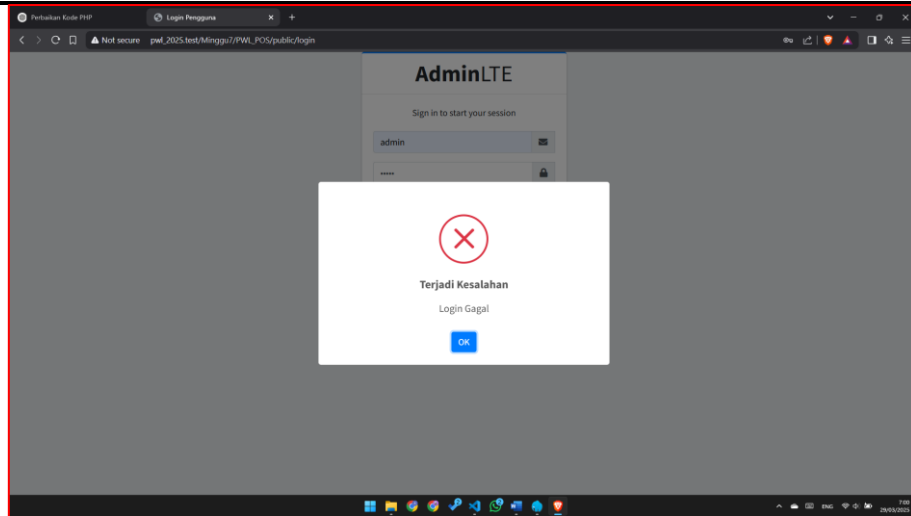


KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141  
Telp. (0341) 404424 – 404425, Fax (0341) 404420  
<http://www.polinema.ac.id>

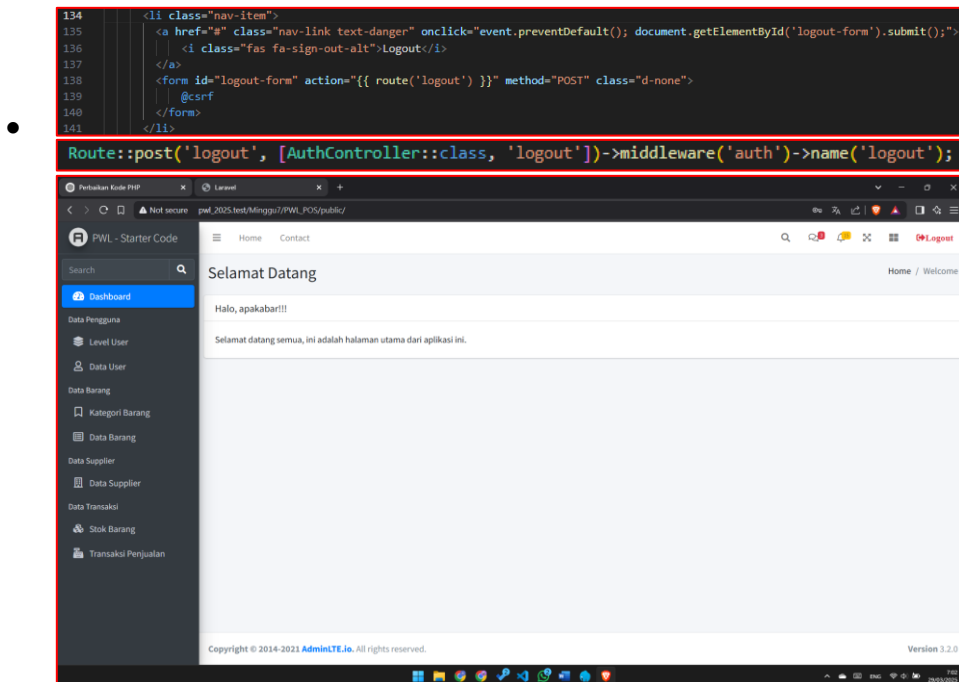


- Gagal





2. Silahkan implementasi proses logout pada halaman web yang kalian buat



3. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
4. Submit kode untuk impementasi Authentication pada repository github kalian.



## B. Implementasi *Authorization* di Laravel

*Authorization* merupakan proses setelah *authentication* berhasil dilakukan (dalam kata lain, kita berhasil login ke sistem). *Authorization* berkenaan dengan hak akses pengguna dalam menggunakan sistem. *Authorization* memberikan/memastikan hak akses (ijin akses) kita, sesuai dengan aturan (role) yang ada di sistem. *Authorization* sangat penting untuk membatasi akses pengguna sesuai dengan peruntukannya.

**Contoh** ketika kita mengakses LMS dengan akun (*username* dan *password*) yang bertipe **Mahasiswa**. Saat berhasil melakukan *authentication*, maka hak akses kita juga akan diberikan selayaknya mahasiswa. Seperti melihat kursus (course), melihat materi, men-download file materi, mengerjakan/meng-upload tugas, mengikuti ujian, dll. Kita tidak akan diberikan hak akses oleh sistem untuk membuat materi, membuat soal ujian, membuat tugas, memberikan nilai tugas karena hak akses tersebut masuk ke ranah akun tipe **Dosen/Pengajar**.

Selain menyediakan layanan otentikasi bawaan, Laravel juga menyediakan cara sederhana untuk mengotorisasi tindakan pengguna terhadap sumber daya tertentu. Misalnya, meskipun pengguna diautentikasi, mereka mungkin tidak berwenang untuk memperbarui atau menghapus model Eloquent atau rekaman database tertentu yang dikelola oleh aplikasi Anda. Fitur otorisasi Laravel menyediakan cara yang mudah dan terorganisir untuk mengelola jenis pemeriksaan otorisasi ini.

### Praktikum 2 – Implementasi *Authorization* di Laravel dengan Middleware

Kita akan menerapkan *authorization* pada project Laravel dengan menggunakan Middleware sebagai pengecekan akses. Langkah-langkah yang kita kerjakan sebagai berikut:

1. Kita modifikasi **UserModel.php** dengan menambahkan kode berikut

```
/**
 * Relasi ke tabel level
 */
public function level(): BelongsTo
{
    return $this->belongsTo(LevelModel::class, 'level_id', 'level_id');
}

/**
 * Mendapatkan nama role
 */
public function getRoleName(): string
{
    return $this->level->level_nama;
}

/**
 * Cek apakah user memiliki role tertentu
 */
public function hasRole($role): bool
{
    return $this->level->level_kode == $role;
}
```



2. Kemudian kita buat *middleware* dengan nama `AuthorizeUser.php`. Kita bisa buat *middleware* dengan mengetikkan perintah pada terminal/CMD

```
php artisan make:middleware AuthorizeUser
```

File *middleware* akan dibuat di `app/Http/Middleware/AuthorizeUser.php`

3. Kemudian kita edit *middleware* `AuthorizeUser.php` untuk bisa mengecek apakah pengguna yang mengakses memiliki Level/Role/Group yang sesuai

```
1  <?php
2  namespace App\Http\Middleware;
3
4  use Closure;
5  use Illuminate\Http\Request;
6  use Symfony\Component\HttpFoundation\Response;
7
8  class AuthorizeUser
9  {
10     /**
11      * Handle an incoming request.
12      *
13      * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)
14      */
15     public function handle(Request $request, Closure $next, $role = ''): Response
16     {
17         $user = $request->user(); // ambil data user yg login
18         // fungsi user() diambil dari UserModel.php
19         if($user->hasRole($role)){ // cek apakah user punya role yg diinginkan
20             return $next($request);
21         }
22         // jika tidak punya role, maka tampilkan error 403
23         abort(403, 'Forbidden. Kamu tidak punya akses ke halaman ini');
24     }
25 }
```

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

```
protected $middlewareAliases = [
    'auth' => \App\Http\Middleware\Authenticate::class,
    'authorize' => \App\Http\Middleware\AuthorizeUser::class, // middleware yg kita buat
    'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,
```

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

level_id	level_kode	level_nama	created_at	updated_at	deleted_at
1	ADM	Administrator	NULL	NULL	NULL
2	MNG	Manager	NULL	NULL	NULL
3	STF	Staf	NULL	2024-08-16 01:49:20	NULL
4	KSR	Kasir	NULL	NULL	NULL



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

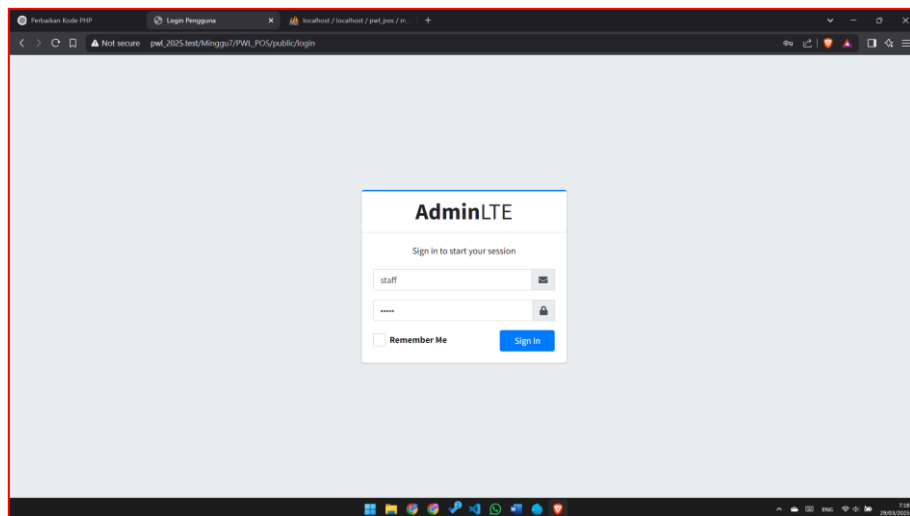
```
Route::middleware(['auth'])->group(function(){ // artinya semua route di dalam group ini harus login dulu
    Route::get('/', [WelcomeController::class,'index']);
    // route Level

    // artinya semua route di dalam group ini harus punya role ADM (Administrator)
    Route::middleware(['authorize:ADM'])->group(function(){
        Route::get('/level',[LevelController::class,'index']);
        Route::post('/level/list',[LevelController::class,'list']); // untuk list json datatables
        Route::get('/level/create',[LevelController::class,'create']);
        Route::post('/level',[LevelController::class,'store']);
        Route::get('/level/{id}/edit',[LevelController::class,'edit']); // untuk tampilkan form edit
        Route::put('/level/{id}',[LevelController::class,'update']); // untuk proses update data
        Route::delete('/level/{id}',[LevelController::class,'destroy']); // untuk proses hapus data
    });

    // route Kategori
```

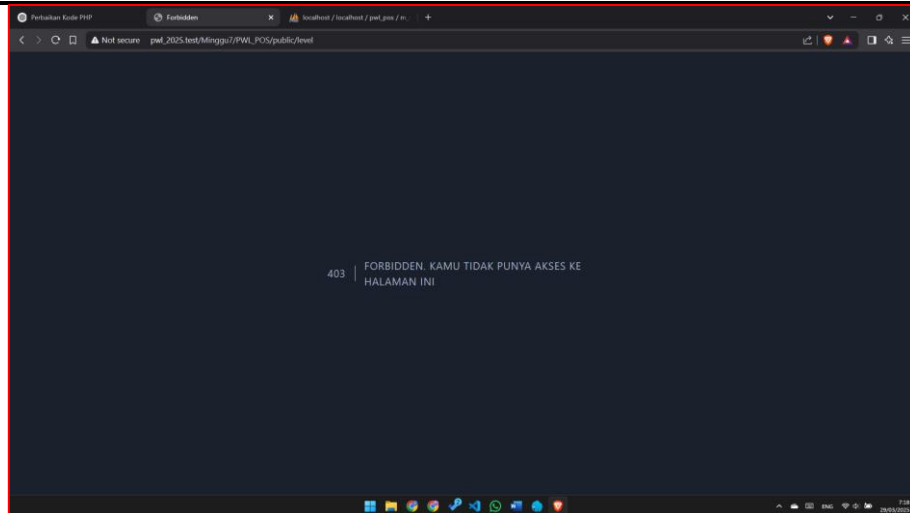
Pada kode yang ditandai merah, terdapat `authorize:ADM`. Kode `ADM` adalah nilai dari `level_kode` pada tabel `m_level`. Yang artinya, user yang bisa mengakses route untuk manage data level, adalah user yang memiliki level sebagai Administrator.

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

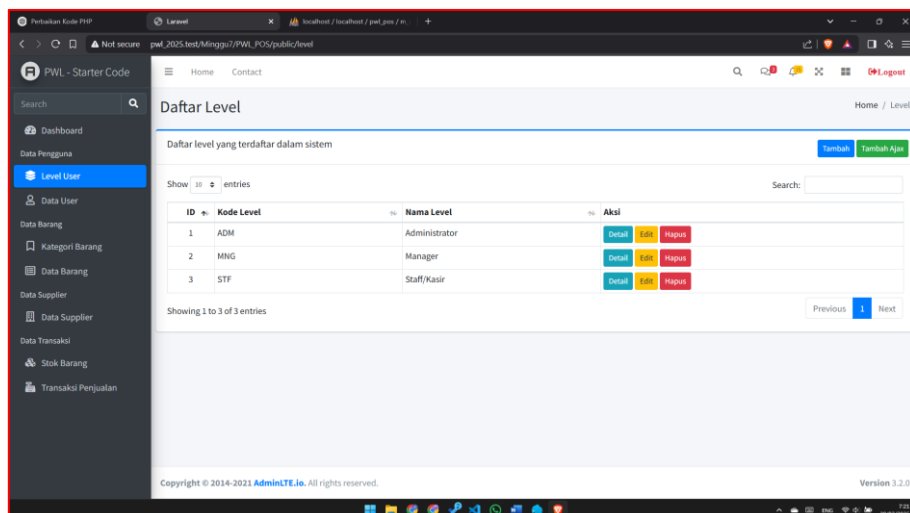
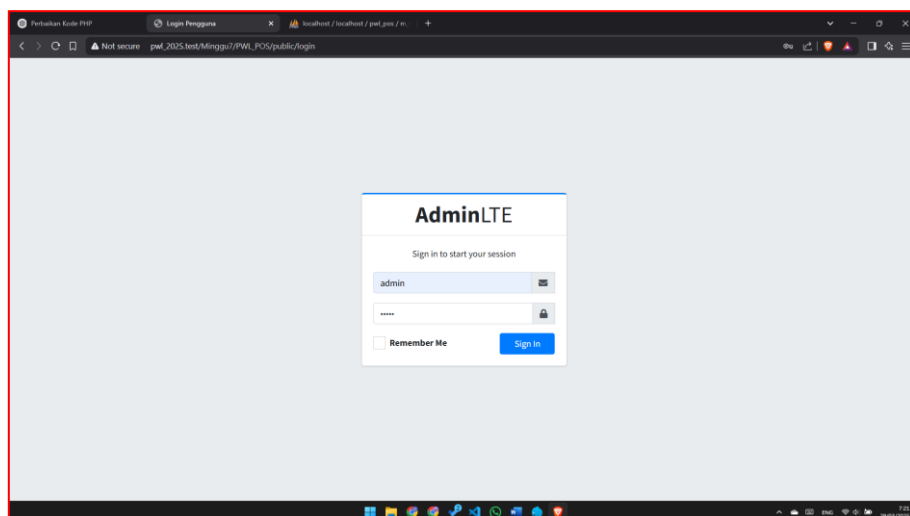




KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141  
Telp. (0341) 404424 – 404425, Fax (0341) 404420  
<http://www.polinema.ac.id>



- Admin





## **Tugas 2** – Implementasi Authoriization :

1. Apa yang kalian pahami pada praktikum 2 ini?
  - Dengan menambahkan authorize:... pada suatu route, akses ke route tersebut terbatas hanya untuk pengguna dengan role tertentu. Pengguna dengan role lain tidak dapat mengaksesnya dan akan diarahkan ke halaman error 404.
2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
3. Submit kode untuk impementasi Authorization pada repository github kalian.





## C. Multi-Level Authorization di Laravel

Bagaimana seandainya jika terdapat level/group/role satu dengan yang lain memiliki hak akses yang sama. Contoh sederhana, user level Admin dan Manager bisa sama-sama mengakses menu Barang pada aplikasi yang kita buat. Maka tidak mungkin kalau kita buat route untuk masing-masing level user. Hal ini akan memakan banyak waktu, dan proses yang lama.

```
// artinya semua route di dalam group ini harus punya role ADM (Administrator)
Route::middleware(['authorize:ADM'])->group(function(){
    Route::get('/barang',[BarangController::class,'index']);
    Route::post('/barang/list',[BarangController::class,'list']);
    Route::get('/barang/create_ajax',[BarangController::class,'create_ajax']); // ajax form create
    Route::post('/barang_ajax',[BarangController::class,'store_ajax']); // ajax store
    Route::get('/barang/{id}/edit_ajax',[BarangController::class,'edit_ajax']); // ajax form edit
    Route::put('/barang/{id}/update_ajax',[BarangController::class,'update_ajax']); // ajax update
    Route::get('/barang/{id}/delete_ajax',[BarangController::class,'confirm_ajax']); // ajax form confirm delete
    Route::delete('/barang/{id}/delete_ajax',[BarangController::class,'delete_ajax']); // ajax delete
});

// artinya semua route di dalam group ini harus punya role MNG (Manager)
Route::middleware(['authorize:MNG'])->group(function(){
    Route::get('/barang',[BarangController::class,'index']);
    Route::post('/barang/list',[BarangController::class,'list']);
    Route::get('/barang/create_ajax',[BarangController::class,'create_ajax']); // ajax form create
    Route::post('/barang_ajax',[BarangController::class,'store_ajax']); // ajax store
    Route::get('/barang/{id}/edit_ajax',[BarangController::class,'edit_ajax']); // ajax form edit
    Route::put('/barang/{id}/update_ajax',[BarangController::class,'update_ajax']); // ajax update
    Route::get('/barang/{id}/delete_ajax',[BarangController::class,'confirm_ajax']); // ajax form confirm delete
    Route::delete('/barang/{id}/delete_ajax',[BarangController::class,'delete_ajax']); // ajax delete
});
```

Hal ini jadi kendala ketika kita mau mengganti hak akses, maka kita akan mengganti sebagian besar route yang sudah kita tulis. Untuk itu, kita perlu mengelola middleware agar bisa mendukung penambahan hak akses secara dinamis.

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

Kita akan menerapkan multi-level authorization pada project Laravel dengan menggunakan Middleware sebagai pengecekan akses. Langkah-langkah yang kita kerjakan sebagai berikut:

1. Kita modifikasi `UserModel.php` untuk mendapatkan `level_kode` dari user yang sudah login. Jadi kita buat fungsi dengan nama `getRole()`



```
/**
 * Mendapatkan nama role
 */
public function getRoleName(): string
{
    return $this->level->level_nama;
}

/**
 * Cek apakah user memiliki role tertentu
 */
public function hasRole($role): bool
{
    return $this->level->level_kode == $role;
}

/**
 * Mendapatkan kode role
 */
public function getRole()
{
    return $this->level->level_kode;
}
```

2. Selanjutnya, Kita modifikasi middleware [AuthorizeUser.php](#) dengan kode berikut

```
1  <?php
2  namespace App\Http\Middleware;
3
4  use Closure;
5  use Illuminate\Http\Request;
6  use Symfony\Component\HttpFoundation\Response;
7
8  class AuthorizeUser
9  {
10     /**
11      * Handle an incoming request.
12      *
13      * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)
14      */
15     public function handle(Request $request, Closure $next, ... $roles): Response
16     {
17         $user_role = $request->user()->getRole(); // ambil data level_kode dari user yg login
18         if(in_array($user_role, $roles)){ // cek apakah level_kode user ada di dalam array roles
19             return $next($request); // jika ada, maka lanjutkan request
20         }
21         // jika tidak punya role, maka tampilkan error 403
22         abort(403, 'Forbidden. Kamu tidak punya akses ke halaman ini');
23     }
24 }
```

3. Setelah itu tinggal kita perbaiki [route/web.php](#) sesuaikan dengan role/level yang diinginkan. Contoh



```
// artinya semua route di dalam group ini harus punya role ADM (Administrator) dan MNG (Manager)
Route::middleware(['authorize:ADM,MNG'])->group(function(){
    Route::get('/barang',[BarangController::class,'index']);
    Route::post('/barang/list',[BarangController::class,'list']);
    Route::get('/barang/create_ajax',[BarangController::class,'create_ajax']); // ajax form create
    Route::post('/barang_ajax',[BarangController::class,'store_ajax']); // ajax store
    Route::get('/barang/{id}/edit_ajax',[BarangController::class,'edit_ajax']); // ajax form edit
    Route::put('/barang/{id}/update_ajax',[BarangController::class,'update_ajax']); // ajax update
    Route::get('/barang/{id}/delete_ajax',[BarangController::class,'confirm_ajax']); // ajax form confirm
    Route::delete('/barang/{id}/delete_ajax',[BarangController::class,'delete_ajax']); // ajax delete
});
```

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



### Tugas 3 – Implementasi Multi-Level Authorization :

1. Silahkan implementasikan multi-level authorization pada project kalian masing-masing
  - Admin

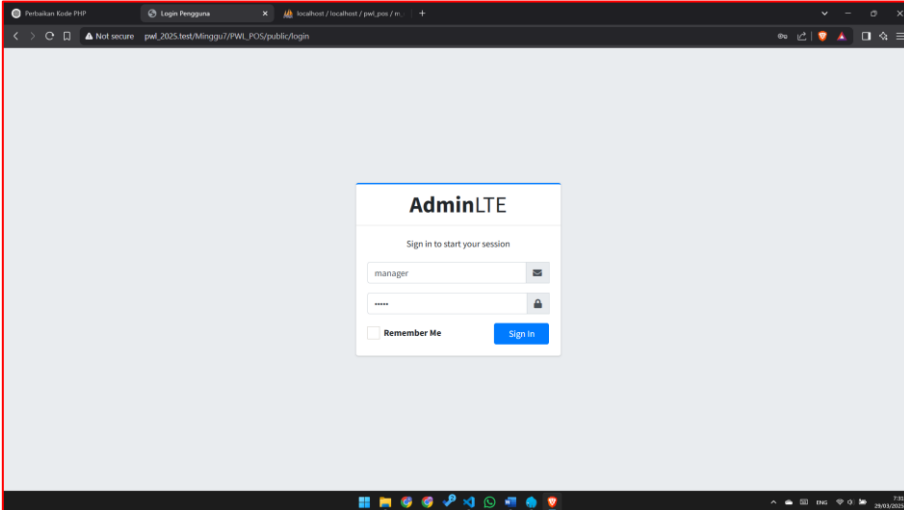
The top screenshot shows the AdminLTE login page with a sign-in form. The bottom screenshot shows a sample data table for 'Daftar Barang' (List of Goods) with columns for ID, Code, Name, Buy Price, Sell Price, Category, and Actions.

ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori Barang	Aksi
1	MKN001	Nasi Goreng	10000	15000	Makanan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	MKN002	Mie Goreng	5000	10000	Makanan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	MIN001	Teh Botol	2000	4000	Minuman	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	MIN002	Aqua	1000	2000	Minuman	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	PKN001	Kemeja	50000	70000	Pakaian	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
6	PKN002	Celana	70000	100000	Pakaian	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
7	ATK001	Pensil	1000	2000	Alat Tulis	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
8	ATK002	Buku Tulis	5000	10000	Alat Tulis	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
9	MIN001	Lego	20000	25000	Mainan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
10	MIN002	Boneka	30000	40000	Mainan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

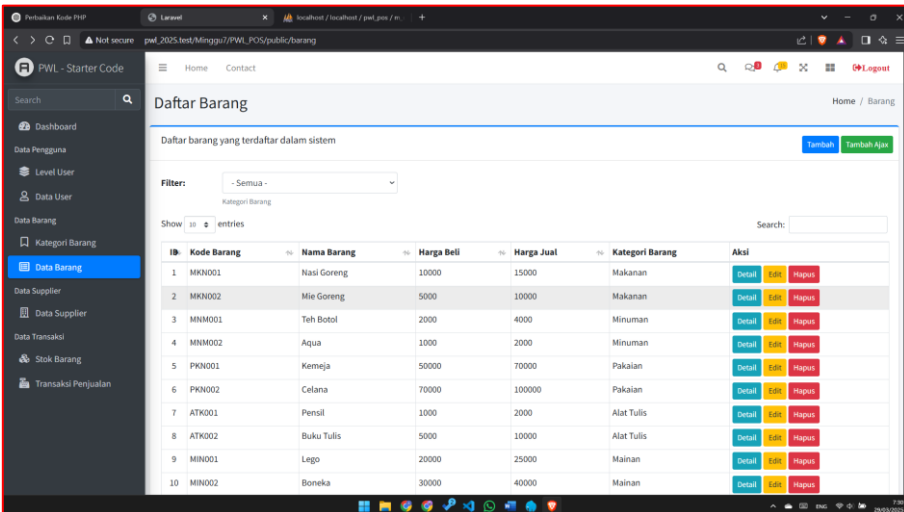


KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141  
Telp. (0341) 404424 – 404425, Fax (0341) 404420  
<http://www.polinema.ac.id>

- **Manager**



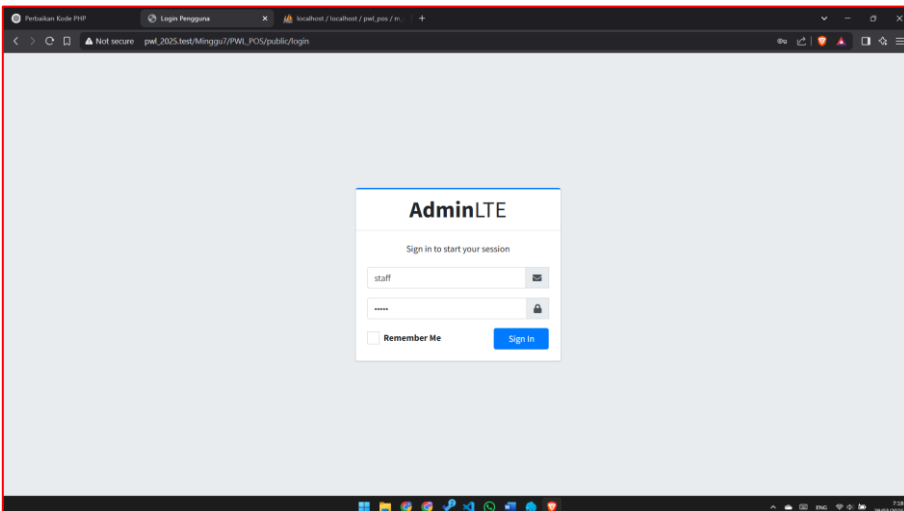
The screenshot shows the AdminLTE login page for a Manager user. The login form is centered on a light gray background. It includes a username field with 'manager' entered, a password field with '\*\*\*\*\*', a 'Remember Me' checkbox, and a 'Sign In' button.



The screenshot shows the AdminLTE dashboard for a Manager user. The dashboard has a dark sidebar with a search bar and a list of menu items: Dashboard, Data Pengguna, Level User, Data User, Data Barang, Kategori Barang, Data Barang (highlighted), Data Supplier, Data Supplier, Data Transaksi, Stok Barang, and Transaksi Penjualan. The main content area is titled 'Daftar Barang' and displays a table of goods. The table has columns for ID, Kode Barang, Nama Barang, Harga Beli, Harga Jual, Kategori Barang, and Aksi. The table contains 10 rows of data.

ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori Barang	Aksi
1	MKN001	Nasi Goreng	10000	15000	Makanan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	MKN002	Mie Goreng	5000	10000	Makanan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	MIN001	Teh Botol	2000	4000	Minuman	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	MIN002	Aqua	1000	2000	Minuman	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	PKN001	Kemeja	50000	70000	Pakaian	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
6	PKN002	Celana	70000	100000	Pakaian	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
7	ATK001	Pensil	1000	2000	Alat Tulis	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
8	ATK002	Buku Tulis	5000	10000	Alat Tulis	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
9	MIN001	Lego	20000	25000	Mainan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
10	MIN002	Boneka	30000	40000	Mainan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

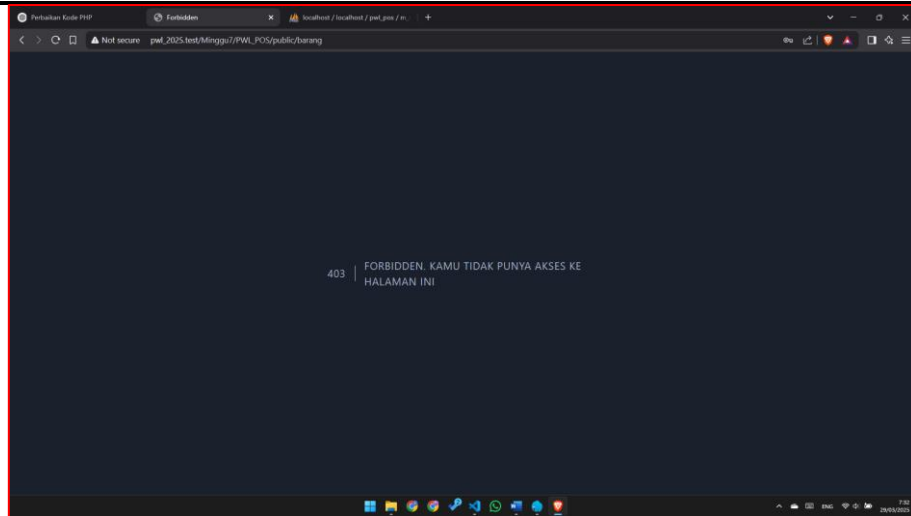
- **Staff**



The screenshot shows the AdminLTE login page for a Staff user. The login form is centered on a light gray background. It includes a username field with 'staff' entered, a password field with '\*\*\*\*\*', a 'Remember Me' checkbox, and a 'Sign In' button.



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  
<http://www.polinema.ac.id>



2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan
3. Implementasikan multi-level authorization untuk semua Level/Jenis User dan Menu-menu yang sesuai dengan Level/Jenis User



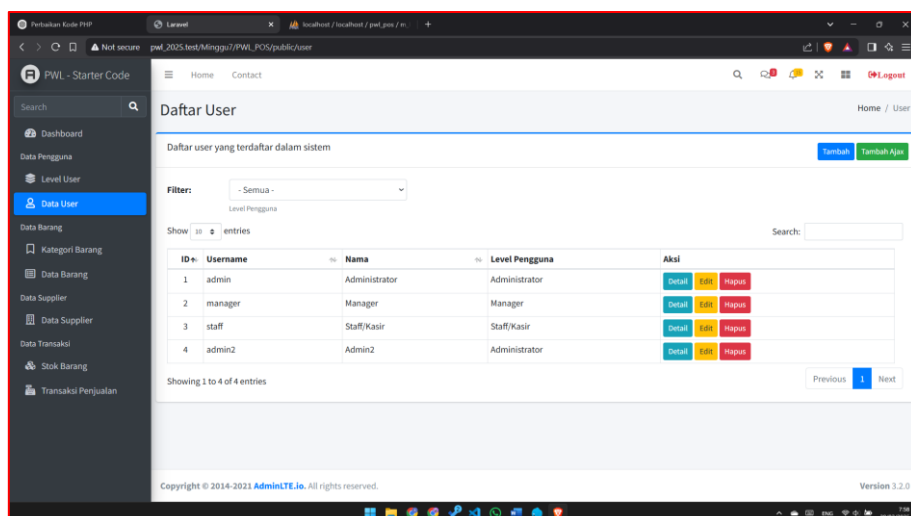
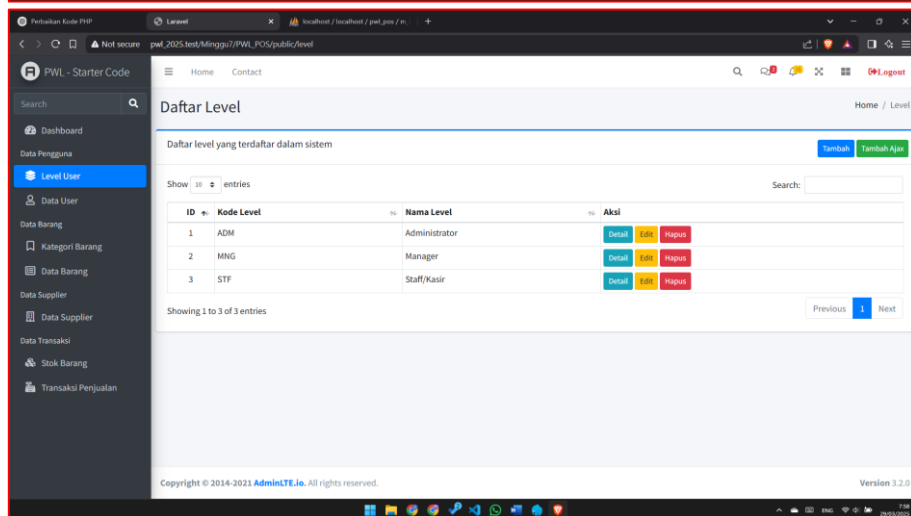
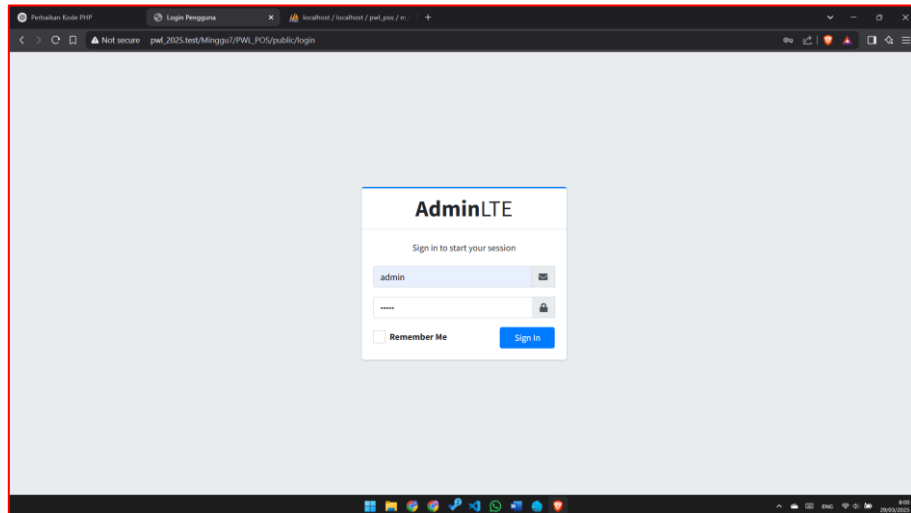
```
1 <?php
2
3 use App\Http\Controllers\UserController;
4 use App\Http\Controllers\SupplierController;
5 use App\Http\Controllers\BarangController;
6 use App\Http\Controllers\AuthController;
7 use App\Http\Controllers\KategoriController;
8 use App\Http\Controllers\LevelController;
9 use App\Http\Controllers\WelcomeController;
10 use Illuminate\Support\Facades\Route;
11
12 /*
13 -----
14 | Web Routes
15 -----
16 |
17 | Here is where you can register web routes for your application. These
18 | routes are loaded by the RouteServiceProvider and all of them will
19 | be assigned to the 'web' middleware group. Make something great!
20 |
21 */
22
23 Route::pattern('id', '[0-9]+');
24
25 Route::get('login', [AuthController::class, 'login'])->name('login');
26 Route::post('login', [AuthController::class, 'postlogin']);
27 Route::post('logout', [AuthController::class, 'logout'])->middleware('auth')->name('logout');
28
29 Route::middleware('auth')->group(function () {
30     Route::get('/', [WelcomeController::class, 'index']);
31
32     Route::middleware(['authorize:ADM'])->group(function () {
33         Route::group(['prefix' => 'barang'], function () {
34             Route::get('/create', [BarangController::class, 'create']);
35             Route::post('/', [BarangController::class, 'store']);
36             Route::get('/create_ajax', [BarangController::class, 'create_ajax']);
37             Route::get('/{id}/edit', [BarangController::class, 'edit']);
38             Route::put('/{id}', [BarangController::class, 'update']);
39             Route::get('/{id}/edit_ajax', [BarangController::class, 'edit_ajax']);
40             Route::put('/{id}/edit_ajax', [BarangController::class, 'update_ajax']);
41             Route::delete('/{id}', [BarangController::class, 'destroy']);
42             Route::get('/{id}/delete_ajax', [BarangController::class, 'confirm_ajax']);
43             Route::delete('/{id}/delete_ajax', [BarangController::class, 'delete_ajax']);
44         });
45
46         Route::group(['prefix' => 'user'], function () {
47             Route::get('/', [UserController::class, 'index']);
48             Route::post('/list', [UserController::class, 'list']);
49             Route::get('/create', [UserController::class, 'create']);
50             Route::post('/', [UserController::class, 'store']);
51             Route::get('/create_ajax', [UserController::class, 'create_ajax']);
52             Route::post('/ajax', [UserController::class, 'store_ajax']);
53             Route::get('/{id}', [UserController::class, 'show']);
54             Route::get('/{id}/edit', [UserController::class, 'edit']);
55             Route::put('/{id}', [UserController::class, 'update']);
56             Route::get('/{id}/edit_ajax', [UserController::class, 'edit_ajax']);
57             Route::put('/{id}/edit_ajax', [UserController::class, 'update_ajax']);
58             Route::delete('/{id}', [UserController::class, 'destroy']);
59             Route::get('/{id}/delete_ajax', [UserController::class, 'confirm_ajax']);
60             Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']);
61         });
62
63         Route::group(['prefix' => 'level'], function () {
64             Route::get('/', [LevelController::class, 'index']);
65             Route::post('/list', [LevelController::class, 'list']);
66             Route::get('/create', [LevelController::class, 'create']);
67             Route::post('/', [LevelController::class, 'store']);
68             Route::get('/create_ajax', [LevelController::class, 'create_ajax']);
69             Route::post('/ajax', [LevelController::class, 'store_ajax']);
70             Route::get('/{id}', [LevelController::class, 'show']);
71             Route::get('/{id}/edit', [LevelController::class, 'edit']);
72             Route::put('/{id}', [LevelController::class, 'update']);
73             Route::get('/{id}/edit_ajax', [LevelController::class, 'edit_ajax']);
74             Route::put('/{id}/edit_ajax', [LevelController::class, 'update_ajax']);
75             Route::delete('/{id}', [LevelController::class, 'destroy']);
76             Route::get('/{id}/delete_ajax', [LevelController::class, 'confirm_ajax']);
77             Route::delete('/{id}/delete_ajax', [LevelController::class, 'delete_ajax']);
78         });
79
80         Route::group(['prefix' => 'supplier'], function () {
81             Route::get('/create', [SupplierController::class, 'create']);
82             Route::post('/', [SupplierController::class, 'store']);
83             Route::get('/create_ajax', [SupplierController::class, 'create_ajax']);
84             Route::post('/ajax', [SupplierController::class, 'store_ajax']);
85             Route::get('/{id}/edit', [SupplierController::class, 'edit']);
86             Route::put('/{id}', [SupplierController::class, 'update']);
87             Route::get('/{id}/edit_ajax', [SupplierController::class, 'edit_ajax']);
88             Route::put('/{id}/edit_ajax', [SupplierController::class, 'update_ajax']);
89             Route::delete('/{id}', [SupplierController::class, 'destroy']);
90             Route::get('/{id}/delete_ajax', [SupplierController::class, 'confirm_ajax']);
91             Route::delete('/{id}/delete_ajax', [SupplierController::class, 'delete_ajax']);
92         });
93
94         Route::group(['prefix' => 'kategori'], function () {
95             Route::get('/create', [KategoriController::class, 'create']);
96             Route::post('/', [KategoriController::class, 'store']);
97             Route::get('/create_ajax', [KategoriController::class, 'create_ajax']);
98             Route::post('/ajax', [KategoriController::class, 'store_ajax']);
99             Route::get('/{id}/edit', [KategoriController::class, 'edit']);
100            Route::put('/{id}', [KategoriController::class, 'update']);
101            Route::get('/{id}/edit_ajax', [KategoriController::class, 'edit_ajax']);
102            Route::put('/{id}/edit_ajax', [KategoriController::class, 'update_ajax']);
103            Route::delete('/{id}', [KategoriController::class, 'destroy']);
104            Route::get('/{id}/delete_ajax', [KategoriController::class, 'confirm_ajax']);
105            Route::delete('/{id}/delete_ajax', [KategoriController::class, 'delete_ajax']);
106        });
107    });
108
109    Route::middleware(['authorize:ADM,PMG'])->group(function () {
110        Route::group(['prefix' => 'supplier'], function () {
111            Route::get('/', [SupplierController::class, 'index']);
112            Route::post('/list', [SupplierController::class, 'list']);
113            Route::get('/{id}', [SupplierController::class, 'show']);
114        });
115    });
116
117    Route::middleware(['authorize:ADM,PMG,STF'])->group(function () {
118        Route::group(['prefix' => 'barang'], function () {
119            Route::get('/', [BarangController::class, 'index']);
120            Route::post('/list', [BarangController::class, 'list']);
121            Route::get('/{id}', [BarangController::class, 'show']);
122        });
123    });
124
125    Route::group(['prefix' => 'kategori'], function () {
126        Route::get('/', [KategoriController::class, 'index']);
127        Route::post('/list', [KategoriController::class, 'list']);
128        Route::get('/{id}', [KategoriController::class, 'show']);
129    });
130
131 });
```





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  
<http://www.polinema.ac.id>

- Admin





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  
<http://www.polinema.ac.id>

Daftar Kategori

Daftar kategori yang terdaftar dalam sistem

Show 5 entries

ID	Kode Kategori	Nama Kategori	Aksi
1	MKN	Makanan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	MNM	Minuman	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	PKN	Pakaian	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	ATK	Alat Tulis	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	MIN	Mainan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Showing 1 to 5 of 5 entries

Previous 1 Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

Daftar Barang

Daftar barang yang terdaftar dalam sistem

Filter: - Semua -

Show 10 entries

ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori Barang	Aksi
1	MKN001	Nasi Goreng	10000	15000	Makanan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	MKN002	Mie Goreng	5000	10000	Makanan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	MNM001	Teh Botol	2000	4000	Minuman	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	MNM002	Aqua	1000	2000	Minuman	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	PKN001	Kemeja	50000	70000	Pakaian	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
6	PKN002	Celana	70000	100000	Pakaian	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
7	ATK001	Pensil	1000	2000	Alat Tulis	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
8	ATK002	Buku Tulis	5000	10000	Alat Tulis	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
9	MIN001	Lego	20000	25000	Mainan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
10	MIN002	Boneka	30000	40000	Mainan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Showing 1 to 10 of 10 entries

Previous 1 Next

Daftar Supplier

Daftar supplier yang terdaftar dalam sistem

Show 10 entries

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	S001	PT. ABC	Jl. Raya No. 1	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	S002	PT. DEF	Jl. Raya No. 2	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	S003	PT. GHI	Jl. Raya No. 3	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	S004	PT. JKL	Jl. Raya No. 4	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	S005	PT. MNO	Jl. Raya No. 5	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
6	S006	PT. PQR	Jl. Raya No. 6	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
7	S007	PT. STU	Jl. Raya No. 7	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
8	S008	PT. VWX	Jl. Raya No. 8	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
9	S009	PT. YZA	Jl. Raya No. 9	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
10	S010	PT. BCD	Jl. Raya No. 10	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

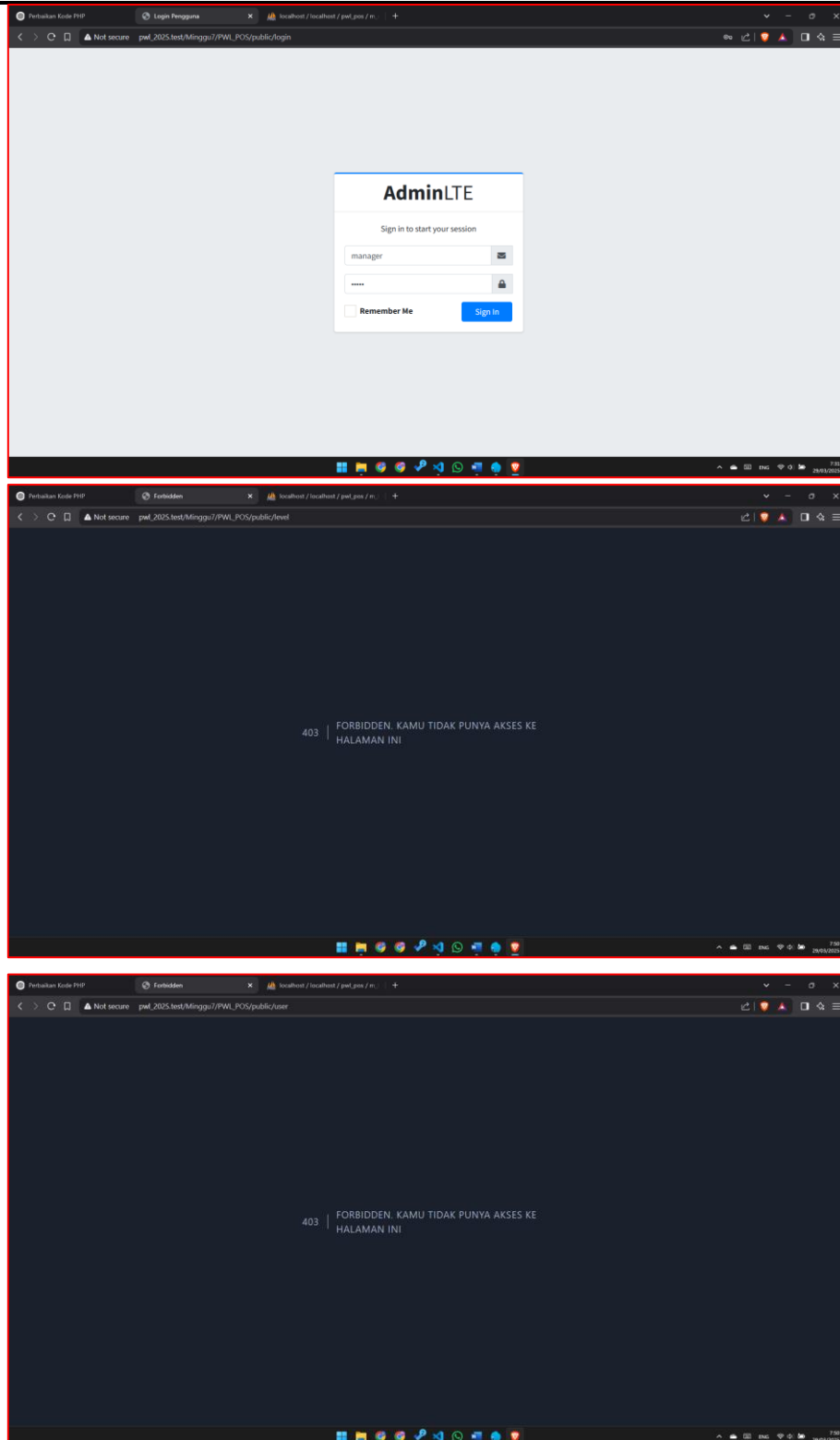
Showing 1 to 10 of 10 entries

Previous 1 Next

- Manager



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  
<http://www.polinema.ac.id>





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  
<http://www.polinema.ac.id>

Daftar Kategori

Daftar kategori yang terdaftar dalam sistem

Show 5 entries

ID	Kode Kategori	Nama Kategori	Aksi
1	MKN	Makanan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	MNM	Minuman	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	PKN	Pakaian	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	ATK	Alat Tulis	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	MIN	Mainan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Showing 1 to 5 of 5 entries

Previous 1 Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

Daftar Barang

Daftar barang yang terdaftar dalam sistem

Filter: - Semua -

Show 10 entries

ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori Barang	Aksi
1	MKN001	Nasi Goreng	10000	15000	Makanan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	MKN002	Mie Goreng	5000	10000	Makanan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	MNM001	Teh Botol	2000	4000	Minuman	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	MNM002	Aqua	1000	2000	Minuman	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	PKN001	Kemeja	50000	70000	Pakaian	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
6	PKN002	Celana	70000	100000	Pakaian	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
7	ATK001	Pensil	1000	2000	Alat Tulis	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
8	ATK002	Buku Tulis	5000	10000	Alat Tulis	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
9	MIN001	Lego	20000	25000	Mainan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
10	MIN002	Boneka	30000	40000	Mainan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Showing 1 to 10 of 10 entries

Previous 1 Next

Daftar Supplier

Daftar supplier yang terdaftar dalam sistem

Show 10 entries

ID	Kode Supplier	Nama Supplier	Alamat Supplier	Aksi
1	S001	PT. ABC	Jl. Raya No. 1	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	S002	PT. DEF	Jl. Raya No. 2	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	S003	PT. GHI	Jl. Raya No. 3	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	S004	PT. JKL	Jl. Raya No. 4	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	S005	PT. MNO	Jl. Raya No. 5	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
6	S006	PT. PQR	Jl. Raya No. 6	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
7	S007	PT. STU	Jl. Raya No. 7	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
8	S008	PT. VWX	Jl. Raya No. 8	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
9	S009	PT. YZA	Jl. Raya No. 9	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
10	S010	PT. BCD	Jl. Raya No. 10	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

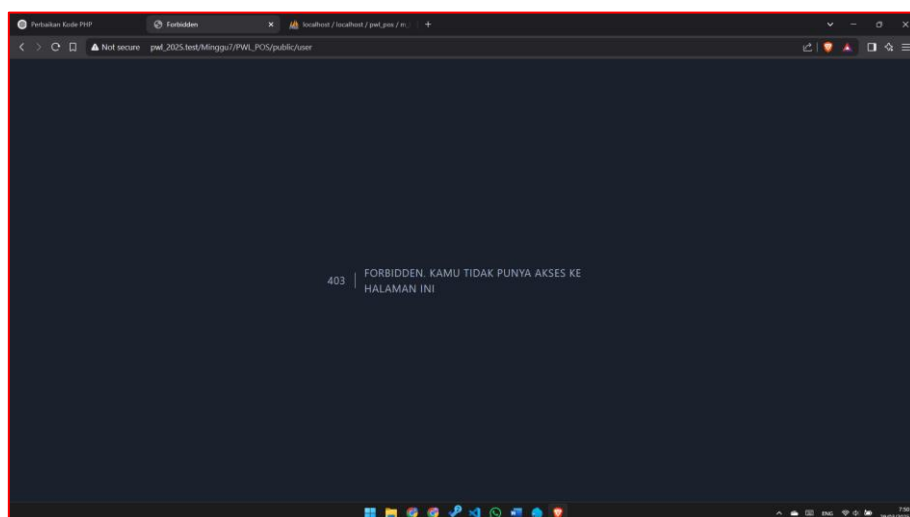
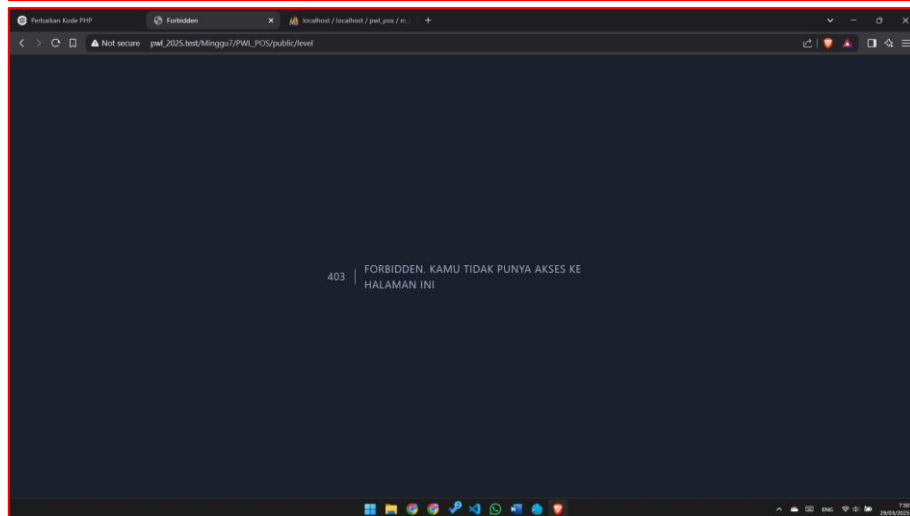
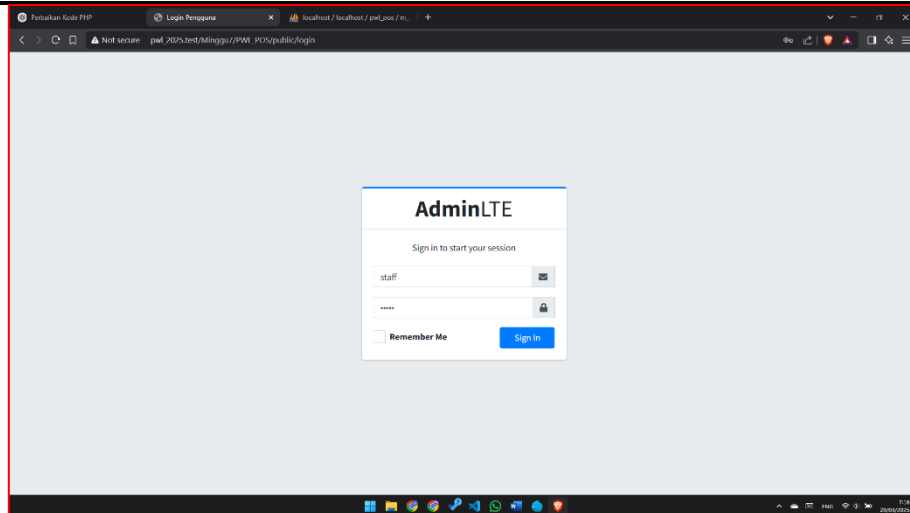
Showing 1 to 10 of 10 entries

Previous 1 Next

- Staff



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  
<http://www.polinema.ac.id>





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  
<http://www.polinema.ac.id>

Daftar Kategori

Daftar kategori yang terdaftar dalam sistem

Show 5 entries

ID	Kode Kategori	Nama Kategori	Aksi
1	MKN	Makanan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	MNM	Minuman	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	PKN	Pakaian	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	ATK	Alat Tulis	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	MIN	Mainan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

Showing 1 to 5 of 5 entries

Previous 1 Next

Copyright © 2014-2021 AdminLTE.io. All rights reserved. Version 3.2.0

Daftar Barang

Daftar barang yang terdaftar dalam sistem

Filter: - Semua -

Show 10 entries

ID	Kode Barang	Nama Barang	Harga Beli	Harga Jual	Kategori Barang	Aksi
1	MKN001	Nasi Goreng	10000	15000	Makanan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
2	MKN002	Mie Goreng	5000	10000	Makanan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
3	MNM001	Teh Botol	2000	4000	Minuman	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
4	MNM002	Aqua	1000	2000	Minuman	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
5	PKN001	Kemeja	50000	70000	Pakaian	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
6	PKN002	Celana	70000	100000	Pakaian	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
7	ATK001	Pensil	1000	2000	Alat Tulis	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
8	ATK002	Buku Tulis	5000	10000	Alat Tulis	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
9	MIN001	Lego	20000	25000	Mainan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>
10	MIN002	Boneka	30000	40000	Mainan	<a href="#">Detail</a> <a href="#">Edit</a> <a href="#">Hapus</a>

403 | FORBIDDEN. KAMU TIDAK PUNYA AKSES KE HALAMAN INI

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



#### Tugas 4 – Implementasi Form Registrasi :

1. Silahkan implementasikan form untuk registrasi user.
2. Screenshot hasil yang kalian kerjakan

The image displays two screenshots of a web application interface, both titled "AdminLTE".

The top screenshot shows the "Sign in to start your session" form. It includes fields for "Username" and "Password", a "Remember Me" checkbox, a "Register?" link, and a "Sign in" button.

The bottom screenshot shows the "Register" form. It includes a "Select Level" dropdown menu, fields for "Username", "Name", "Password", and "Confirm Password", a "Have an account? Login" link, and a "Register" button.





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  
<http://www.polinema.ac.id>

The image displays three sequential screenshots of a web application interface for registration and login, titled "AdminLTE".

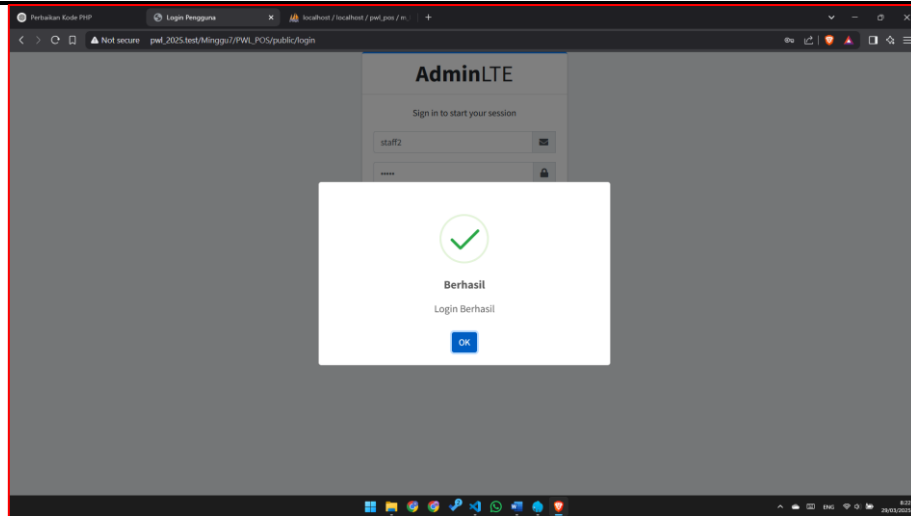
**Top Screenshot (Register Page):** The browser address bar shows "localhost / localhost / pol\_goa / m...". The page title is "Register Pengguna". The URL is "pol\_2025.test/Minggu7/PWL\_POS/public/register". The form includes a dropdown menu for "Staff/Kasir" (set to "staff2"), a text input for "staff2", a text input for "Kasir 2" (set to "Kasir 2"), and two password input fields. A "Register" button is at the bottom. A link "Have an account? Login" is also present.

**Middle Screenshot (Registration Success):** The browser address bar shows "localhost / localhost / pol\_goa / m...". The page title is "Login Pengguna". The URL is "pol\_2025.test/Minggu7/PWL\_POS/public/login". A modal dialog box is displayed with a green checkmark icon and the text "Registration Successful" and "Registration Success.". An "OK" button is at the bottom.

**Bottom Screenshot (Login Page):** The browser address bar shows "localhost / localhost / pol\_goa / m...". The page title is "Login Pengguna". The URL is "pol\_2025.test/Minggu7/PWL\_POS/public/login". The form includes a text input for "staff2", a password input field, a "Remember Me" checkbox, and a "Sign In" button. A link "Register?" is also present.



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  
<http://www.polinema.ac.id>



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

*\*\*\* Sekian, dan selamat belajar \*\*\**