

## **Tugas Pertemuan 7 Pemrograman Web Lanjut**



**Muhammad Farrel Caesarian**  
**TI-2D/16**  
**2341720053**  
**POLITEKNIK NEGERI MALANG**

## 1. **Praktikum 1** – Implementasi Authentication :

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

```
'providers' => [  
    'users' => [  
        'driver' => 'eloquent',  
        'model' => App\Models\UserModel::class,  
    ],  
],
```

Selanjutnya kita modifikasi sedikit pada UserModel.php untuk bisa melakukan proses otentikasi

```
app > Models > UserModel.php > PHP Intelephense > 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 use App\Models\Level;  
10 use App\Models\LevelModel;  
11  
12 class UserModel extends Authenticatable  
13 {  
14     use HasFactory;  
15  
16     protected $table = 'm_user'; // Mendefinisikan nama tabel yang digunakan oleh model ini  
17     protected $primaryKey = 'user_id'; // Mendefinisikan primary key dari tabel yang digunakan  
18  
19     protected $fillable = ['level_id', 'username', 'password', 'nama'];  
20  
21     protected $hidden = ['password'];  
22  
23     protected $casts = ['password' => 'hashed'];  
24  
25     public function level(): BelongsTo  
26     {  
27         return $this->belongsTo(LevelModel::class, 'level_id', 'level_id');  
28     }  
29 }
```

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

```
app > Http > Controllers > AuthController.php > PHP Intelephense > AuthController > postlogin  
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         if (Auth::check()) {  
13             return redirect('/');  
14         }  
15         return view('auth.login');  
16     }  
17  
18     public function postlogin(Request $request)  
19     {  
20         if ($request->ajax() || $request->wantsJson()) {  
21             $credentials = $request->only('username', 'password');  
22  
23             if (Auth::attempt($credentials)) {  
24                 return response()->json([  
25                     'status' => true,  
26                     'message' => 'Login Berhasil',  
27                     'redirect' => url('/'),  
28                 ]);  
29             }  
30  
31             return response()->json([  
32                 'status' => false,  
33                 'message' => 'Login Gagal',  
34             ]);  
35         }  
36  
37         return redirect('login');  
38     }  
39  
40     public function logout(Request $request)  
41     {  
42         Auth::logout();  
43         $request->session()->invalidate();  
44         $request->session()->regenerateToken();  
45  
46         return redirect('login');  
47     }  
48 }
```

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
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Login Pengguna</title>
7 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro">
8 <link rel="stylesheet" href="{{ asset('plugins/fontawesome-free/css/all.min.css') }}">
9 <link rel="stylesheet" href="{{ asset('plugins/bs4-extended/css/bs4-extended.min.css') }}">
10 <link rel="stylesheet" href="{{ asset('plugins/sweetalert2-theme-bootstrap-4/bootstrap-4/sweetalert2-theme-bootstrap-4.css') }}">
11 <link rel="stylesheet" href="{{ asset('dist/css/adminlte.min.css') }}">
12 </head>
13 <body class="hold-transition login-page">
14 <div class="login-box">
15 <div class="card card-outline card-primary">
16 <div class="card-header text-center">
17 <a href="{{ url('/') }}" class="h1"><b>Admin</b></a>
18 </div>
19 <div class="card-body">
20 <p class="login-box-msg">Sign in to start your session</p>
21 <form action="{{ url('login') }}" method="POST" id="form-login">
22 @csrf
23 <div class="input-group mb-3">
24 <input type="text" id="username" name="username" class="form-control">
25 <div class="input-group-append">
26 <div class="input-group-text">
27 <span class="fas fa-envelope"></span>
28 </div>
29 </div>
30 <small id="error-username" class="error-text text-danger"></small>
31 </div>
```

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

```
use App\Http\Controllers\AuthController;

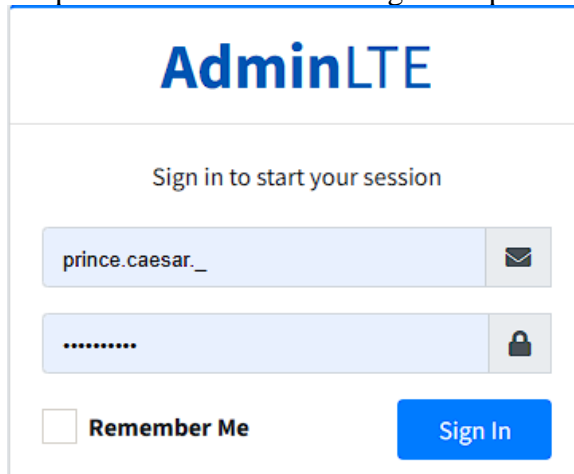
Route::pattern('id','[0-9]+');

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 () {

    Route::group(['prefix' => 'user'], function () {
        Route::get('/', [UserController::class, 'index']); // menampilkan halaman
        Route::get('/create_ajax', [UserController::class, 'create_ajax']); // me
        Route::post('/ajax', [UserController::class, 'store_ajax']); // menyimpan
        Route::get('/{id}/edit_ajax', [UserController::class, 'edit_ajax']); // m
        Route::put('/{id}/update_ajax', [UserController::class, 'update_ajax']);
        Route::get('/{id}/delete_ajax', [UserController::class, 'confirm_ajax']);
        Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']);
        Route::post('/list', [UserController::class, 'list']); // menampilkan dat
        Route::get('/create', [UserController::class, 'create']); // menampilkan
        Route::post('/', [UserController::class, 'store']); // menyimpan data use
        Route::get('/{id}', [UserController::class, 'show']); // menampilkan deta
        Route::get('/{id}/edit', [UserController::class, 'edit']); // menampilkan
        Route::put('/{id}', [UserController::class, 'update']); // menyimpan peru
```

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



Jika username atau password salah



**Terjadi Kesalahan**

Login Gagal

OK

Jika username dan password benar

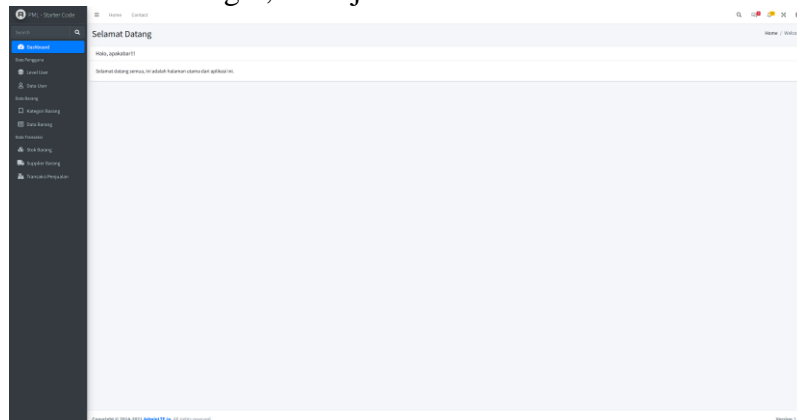


**Berhasil**

Login Berhasil

OK

Setelah admin login, menuju halaman dashboard



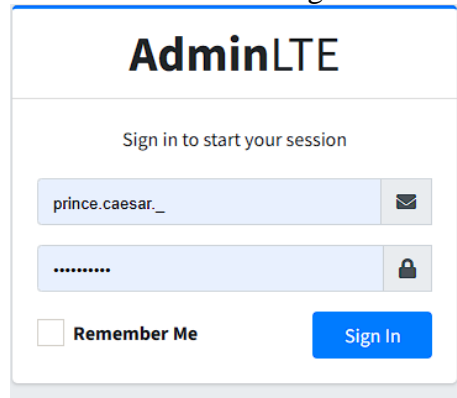
Menambahkan button logout pada navbar

Home Contact Logout

Selamat Datang

```
</li>
<li class="nav-item d-none d-sm-inline-block">
  <a href="../../index3.html" class="nav-link">Home</a>
</li>
<li class="nav-item d-none d-sm-inline-block">
  <a href="#" class="nav-link">Contact</a>
</li>
<li class="nav-item d-none d-sm-inline-block">
  <a href="{{ url('logout') }}" class="nav-link">Logout</a>
</li>
</ul>
```

Kembali ke halaman login

A screenshot of the AdminLTE login page. It features a white card with a blue border. At the top, it says "AdminLTE" in a large, bold, black font. Below that, it says "Sign in to start your session". There are two input fields: one for the username "prince.caesar\_" and one for the password ".....". Both fields have icons (an envelope for email and a lock for password) on the right. Below the password field, there is a checkbox labeled "Remember Me" and a blue button labeled "Sign In".

## 2. **Praktikum 2** – Implementasi *Authorizaton* di Laravel dengan Middleware

Kita modifikasi UserModel.php dengan menambahkan kode berikut

```
public function getRoleName(): String{
    return $this->level->level_nama;
}

public function hasRole ($role): bool{
    return $this->level->level_kode == $role;
}
```

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

```
PS C:\laragon\www\PMU-Muhammad-Farrel-Caesarian_2025\Minggu 7\POS_Jobsheet 7> php artisan make:middleware AuthorizeUser
INFO: Middleware [C:\laragon\www\PMU-Muhammad-Farrel-Caesarian_2025\Minggu 7\POS_Jobsheet 7\App\Http\Middleware\AuthorizeUser.php] created successfully.
```

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

```
public function handle(Request $request, Closure $next, $role = ''): Response
{
    $user = $request->user();

    if ($user->hasRole($role)){
        return $next($request);
    }

    abort(403, 'Forbidden, kamu tidak punya akses ke halaman ini');
}
```

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,
    'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
```

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
lete	1 ADM	Administrator	NULL	NULL
lete	2 MNG	Manager	NULL	NULL
lete	3 STF	Staff/Kasir	NULL	NULL
lete	6 CUS	Pelanggan	2025-03-06 13:35:44	NULL

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(['authorize:ADM'])->group(function () {
    Route::get('/dashboard', [WelcomeController::class, 'dashboard']);


    Route::group(['prefix' => 'level'], function () {
        Route::get('/', [LevelController::class, 'index']);
        Route::get('/create_ajax', [LevelController::class, 'create_ajax']); // menampilkan halaman form tambah user Ajax
        Route::post('/ajax', [LevelController::class, 'store_ajax']); // menyimpan data user baru Ajax
        Route::get("/{id}/edit_ajax", [LevelController::class, 'edit_ajax']); // menampilkan halaman form edit user Ajax
        Route::put("/{id}/update_ajax", [LevelController::class, 'update_ajax']); // menyimpan perubahan data user Ajax
        Route::get("/{id}/delete_ajax", [LevelController::class, 'confirm_ajax']); // menampilkan halaman form Delete user Ajax
        Route::delete("/{id}/delete_ajax", [LevelController::class, 'delete_ajax']); // menghapus data user Ajax
        Route::post('/list', [LevelController::class, 'list']);
        Route::get('/create', [LevelController::class, 'create']);
        Route::post('/', [LevelController::class, 'store']);
        Route::get("/{id}", [LevelController::class, 'show']);
        Route::get("/{id}/edit", [LevelController::class, 'edit']);
        Route::put("/{id}", [LevelController::class, 'update']);
        Route::delete("/{id}", [LevelController::class, 'destroy']);
    });
});
```


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.

Untuk membuktikannya, sekarang kita coba login menggunakan akun selain level administrator, dan kita akses route menu level tersebut  
Jika role bukan admin:

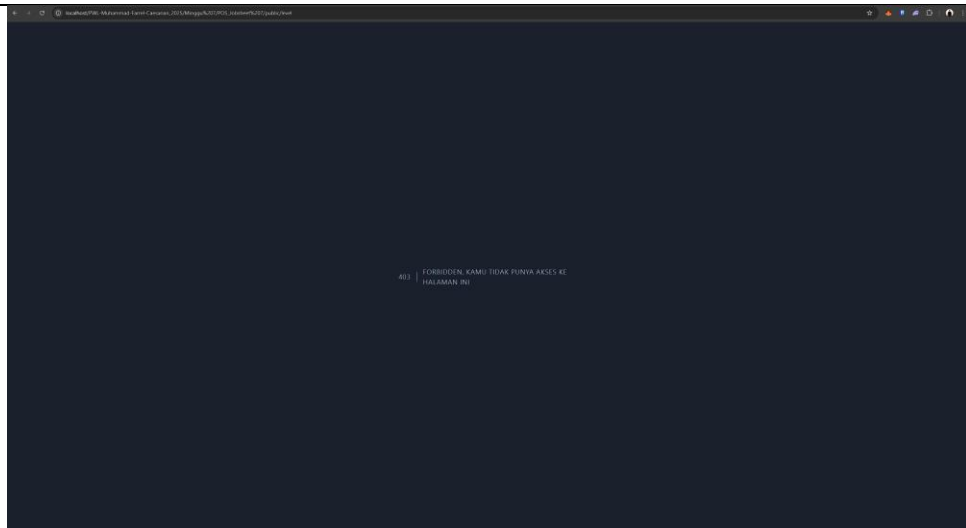
# AdminLTE

Sign in to start your session







☐ Remember Me



Jika role admin:

A screenshot of the AdminLTE login page. It features a white background with a blue header containing the "AdminLTE" logo. Below the header, the text "Sign in to start your session" is displayed. There are two input fields: one for the username "FarrelAdmin" and another for the password, represented by dots. To the right of each field is an icon (an envelope for email and a padlock for password). Below the password field is a "Remember Me" checkbox and a blue "Sign In" button.A screenshot of the AdminLTE dashboard. The left sidebar contains a menu with options like "Dashboard", "Data Pengguna", "Level User", "Data Barang", "Kategori Barang", "Data Barang", "Data Transaksi", "Data Barang", "Supplier Barang", and "Transaksi Pengiriman". The main content area is titled "Daftar Level" and shows a table of user levels. The table has columns for "ID", "Kode Level", "Nama Level", and "Aksi". There are four rows of data. Below the table, it says "Showing 1 to 4 of 4 entries". At the bottom right, there is a "Print" button and a "Close" button.

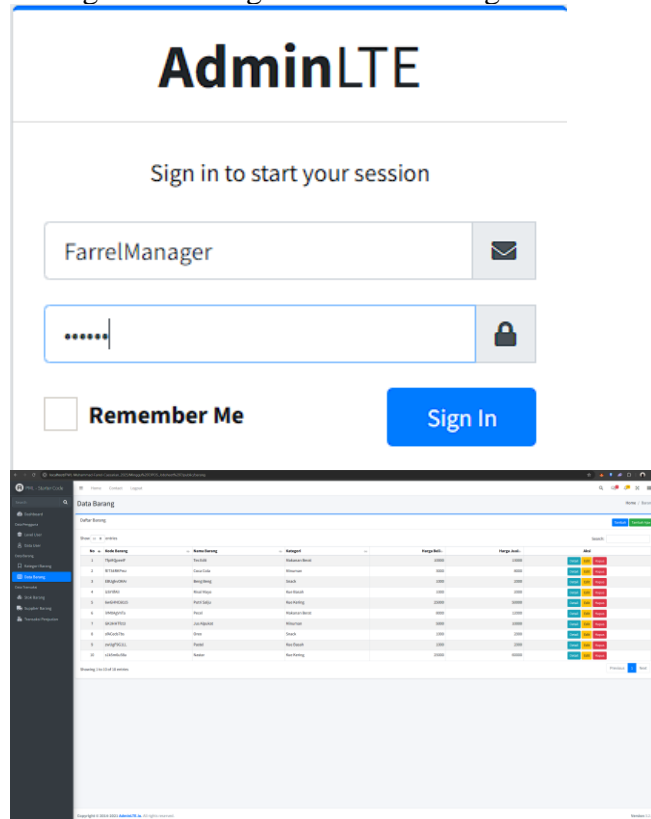
ID	Kode Level	Nama Level	Aksi
1	ADM	Administrator	<a href="#">Edit</a> <a href="#">Hapus</a>
2	MNG	Manager	<a href="#">Edit</a> <a href="#">Hapus</a>
3	STF	Staff/Kasir	<a href="#">Edit</a> <a href="#">Hapus</a>
4	CLS	Pelanggan	<a href="#">Edit</a> <a href="#">Hapus</a>

3.	<p><b>Tugas 2 – Implementasi Authoriization :</b></p> <p>1. Apa yang kalian pahami pada praktikum 2 ini?  Jawab: Tiap pengguna dengan berbagai role dapat diatur untuk mempunyai lingkup kekuasaan yang berbeda-beda, seperti role staff tidak bisa mengakses halaman level karena halaman level hanya diperuntukkan untuk administrator.</p> <p>2. Amati dan jelaskan tiap tahapan yang kalian kerjakan, dan jabarkan dalam laporan</p> <p>3. Submit kode untuk impementasi Authorization pada repository github kalian.</p> <div data-bbox="322 604 730 779"> <p><b>Implementasi Authorization</b>   FarrelCaesarian • just now</p> <hr/> <p><b>Implementasi authentication</b>   FarrelCaesarian • 21 minutes ago</p> </div>
4.	<p><b>Praktikum 3 – Implementasi Multi-Level Authorizaton di Laravel dengan Middleware</b></p> <p>Kita modifikasi UserModel.php untuk mendapatkan level_kode dari user yang sudah login. Jadi kita buat fungsi dengan nama getRole()</p> <div data-bbox="322 1019 906 1169"> <pre>public function getRole () {     return \$this-&gt;level-&gt;level_kode; }</pre> </div> <p>Selanjutnya, Kita modifikasi middleware AuthorizeUser.php dengan kode berikut</p> <div data-bbox="322 1281 1114 1563"> <pre>public function handle(Request \$request, Closure \$next, ... \$roles ): Response {     \$user_role = \$request-&gt;user()-&gt;getRole();      if (in_array(\$user_role, \$roles)){         return \$next(\$request);     }     abort(403, 'Forbidden, kamu tidak punya akses ke halaman ini'); }</pre> </div> <p>Setelah itu tinggal kita perbaiki route/web.php sesuaikan dengan role/level yang diinginkan. Contoh</p> <div data-bbox="322 1675 1279 2033"> <pre>//Admin dan Manager Route::middleware(['authorize:ADM,MNG'])-&gt;group(function () {     Route::group(['prefix' =&gt; 'barang'], function () {         Route::get('/dashboard', [WelcomeController::class, 'dashboard']);         Route::get('/', [BarangController::class, 'index']);         Route::get('/create_ajax', [BarangController::class, 'create_ajax']); // menampilkan halaman form tambah user Ajax         Route::post('/ajax', [BarangController::class, 'store_ajax']); // menyimpan data user baru Ajax         Route::get('/{id}/edit_ajax', [BarangController::class, 'edit_ajax']); // menampilkan halaman form edit user Ajax         Route::put('/{id}/update_ajax', [BarangController::class, 'update_ajax']); // menyimpan perubahan data user Ajax         Route::get('/{id}/delete_ajax', [BarangController::class, 'confirm_ajax']); // menampilkan halaman form Delete user Ajax         Route::delete('/{id}/delete_ajax', [BarangController::class, 'delete_ajax']); // menghapus data user Ajax         Route::post('/list', [BarangController::class, 'list']);         Route::get('/create', [BarangController::class, 'create']);         Route::post('/', [BarangController::class, 'store']);         Route::get('/{id}', [BarangController::class, 'show']);         Route::get('/{id}/edit', [BarangController::class, 'edit']);         Route::put('/{id}', [BarangController::class, 'update']);         Route::delete('/{id}', [BarangController::class, 'destroy']);     }); });</pre> </div>

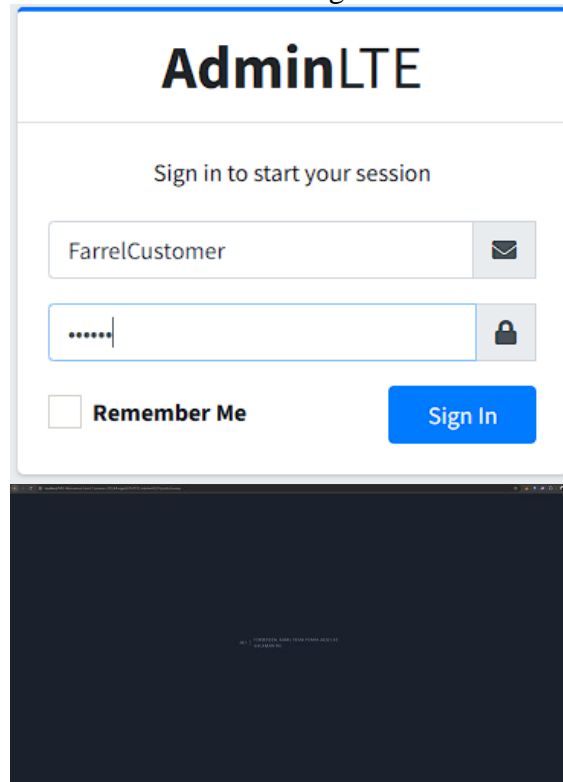


Sekarang kita sudah bisa memberikan hak akses menu/route ke beberapa level user

Manager bisa mengakses Data Barang



Customer tidak bisa mengakses Data Barang



5. **Tugas 3 – Implementasi Multi-Level Authorization :**

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

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

3. Implementasikan multi-level authorization untuk semua Level/Jenis User dan Menumenu yang sesuai dengan Level/Jenis User

```
//Hanya Admin yang bisa melihat Data Level
Route::middleware(['authorize:ADM'])->group(function () {
    Route::get('/dashboard', [WelcomeController::class, 'dashboard']);

    Route::group(['prefix' => 'level'], function () {
        Route::get('/', [LevelController::class, 'index']);
        Route::get('/create_ajax', [LevelController::class, 'create_ajax']); // menampilkan halaman form tambah user Ajax
        Route::post('/ajax', [LevelController::class, 'store_ajax']); // menyimpan data user baru Ajax
        Route::get('/{id}/edit_ajax', [LevelController::class, 'edit_ajax']); // menampilkan halaman form edit user Ajax
        Route::put('/{id}/update_ajax', [LevelController::class, 'update_ajax']); // menyimpan perubahan data user Ajax
        Route::get('/{id}/delete_ajax', [LevelController::class, 'confirm_ajax']); // menampilkan halaman form Delete user Ajax
        Route::delete('/{id}/delete_ajax', [LevelController::class, 'delete_ajax']); // menghapus data user Ajax
        Route::post('/list', [LevelController::class, 'list']);
        Route::get('/create', [LevelController::class, 'create']);
        Route::post('/', [LevelController::class, 'store']);
        Route::get('/{id}', [LevelController::class, 'show']);
        Route::get('/{id}/edit', [LevelController::class, 'edit']);
        Route::put('/{id}', [LevelController::class, 'update']);
        Route::delete('/{id}', [LevelController::class, 'destroy']);
    });
});
```

```
//Admin, Manager, Staff dan Kasir dapat melihat Data Barang
Route::middleware(['authorize:ADM,MNG,STF,KSR'])->group(function () {
    Route::group(['prefix' => 'barang'], function () {
        Route::get('/dashboard', [WelcomeController::class, 'dashboard']);
        Route::get('/', [BarangController::class, 'index']);
        Route::get('/create_ajax', [BarangController::class, 'create_ajax']); // menampilkan halaman form tambah user Ajax
        Route::post('/ajax', [BarangController::class, 'store_ajax']); // menyimpan data user baru Ajax
        Route::get('/{id}/edit_ajax', [BarangController::class, 'edit_ajax']); // menampilkan halaman form edit user Ajax
        Route::put('/{id}/update_ajax', [BarangController::class, 'update_ajax']); // menyimpan perubahan data user Ajax
        Route::get('/{id}/delete_ajax', [BarangController::class, 'confirm_ajax']); // menampilkan halaman form Delete user Ajax
        Route::delete('/{id}/delete_ajax', [BarangController::class, 'delete_ajax']); // menghapus data user Ajax
        Route::post('/list', [BarangController::class, 'list']);
        Route::get('/create', [BarangController::class, 'create']);
        Route::post('/', [BarangController::class, 'store']);
        Route::get('/{id}', [BarangController::class, 'show']);
        Route::get('/{id}/edit', [BarangController::class, 'edit']);
        Route::put('/{id}', [BarangController::class, 'update']);
        Route::delete('/{id}', [BarangController::class, 'destroy']);
    });
});
```

```
//Admin, Manager dan Staff dapat melihat Data Supplier
Route::middleware(['authorize:ADM,MNG,STF'])->group(function () {
    Route::group(['prefix' => 'supplier'], function () {
        Route::get('/', [SupplierController::class, 'index']);
        Route::get('/create_ajax', [SupplierController::class, 'create_ajax']); // menampilkan halaman form tambah user Ajax
        Route::post('/ajax', [SupplierController::class, 'store_ajax']); // menyimpan data user baru Ajax
        Route::get('/{id}/edit_ajax', [SupplierController::class, 'edit_ajax']); // menampilkan halaman form edit user Ajax
        Route::put('/{id}/update_ajax', [SupplierController::class, 'update_ajax']); // menyimpan perubahan data user Ajax
        Route::get('/{id}/delete_ajax', [SupplierController::class, 'confirm_ajax']); // menampilkan halaman form Delete user Ajax
        Route::delete('/{id}/delete_ajax', [SupplierController::class, 'delete_ajax']); // menghapus data user Ajax
        Route::post('/list', [SupplierController::class, 'list']);
        Route::get('/create', [SupplierController::class, 'create']);
        Route::post('/', [SupplierController::class, 'store']);
        Route::get('/{id}', [SupplierController::class, 'show']);
        Route::get('/{id}/edit', [SupplierController::class, 'edit']);
        Route::put('/{id}', [SupplierController::class, 'update']);
        Route::delete('/{id}', [SupplierController::class, 'destroy']);
    });
});
```

```
//Admin dan Manager dapat melihat Data User
Route::middleware(['authorize:ADM,MNG'])->group(function () {
    Route::group(['prefix' => 'user'], function () {
        Route::get('/', [UserController::class, 'index']); // menampilkan halaman awal user
        Route::get('/create_ajax', [UserController::class, 'create_ajax']); // menampilkan halaman form tambah user Ajax
        Route::post('/ajax', [UserController::class, 'store_ajax']); // menyimpan data user baru Ajax
        Route::get('/{id}/edit_ajax', [UserController::class, 'edit_ajax']); // menampilkan halaman form edit user Ajax
        Route::put('/{id}/update_ajax', [UserController::class, 'update_ajax']); // menyimpan perubahan data user Ajax
        Route::get('/{id}/delete_ajax', [UserController::class, 'confirm_ajax']); // menampilkan halaman form Delete user Ajax
        Route::delete('/{id}/delete_ajax', [UserController::class, 'delete_ajax']); // menghapus data user Ajax
        Route::post('/list', [UserController::class, 'list']); // menampilkan data user dalam bentuk json untuk datatables
        Route::get('/create', [UserController::class, 'create']); // menampilkan halaman form tambah user
        Route::post('/', [UserController::class, 'store']); // menyimpan data user baru
        Route::get('/{id}', [UserController::class, 'show']); // menampilkan detail user
        Route::get('/{id}/edit', [UserController::class, 'edit']); // menampilkan halaman form edit user
        Route::put('/{id}', [UserController::class, 'update']); // menyimpan perubahan data user
        Route::delete('/{id}', [UserController::class, 'destroy']); // menghapus data user
    });
});
```

```
//Admin, Manager, Staff dan Kasir dapat melihat Data Kategori
Route::middleware(['authorize:ADM,MNG,STF,KSR'])->group(function () {
Route::group(['prefix' => 'kategori'], function () {
Route::get('/', [KategoriController::class, 'index']);
Route::get('/create_ajax', [KategoriController::class, 'create_ajax']); // menampilkan halaman form tambah user Ajax
Route::post('/ajax', [KategoriController::class, 'store_ajax']); // menyimpan data user baru Ajax
Route::get('/{id}/edit_ajax', [KategoriController::class, 'edit_ajax']); // menampilkan halaman form edit user Ajax
Route::put('/{id}/update_ajax', [KategoriController::class, 'update_ajax']); // menyimpan perubahan data user Ajax
Route::get('/{id}/delete_ajax', [KategoriController::class, 'confirm_ajax']); // menampilkan halaman form Delete user Ajax
Route::delete('/{id}/delete_ajax', [KategoriController::class, 'delete_ajax']); // menghapus data user Ajax
Route::post('/list', [KategoriController::class, 'list']);
Route::get('/create', [KategoriController::class, 'create']);
Route::post('/', [KategoriController::class, 'store']);
Route::get('/{id}', [KategoriController::class, 'show']);
Route::get('/{id}/edit', [KategoriController::class, 'edit']);
Route::put("/{id}", [KategoriController::class, 'update']);
Route::delete("/{id}", [KategoriController::class, 'destroy']);
});
});
```

```
//Semua bisa ke halaman dashboard
Route::middleware(['authorize:ADM,MNG,STF,KSR,CUS'])->group(function () {
Route::get('/dashboard', [WelcomeController::class, 'dashboard']);
});
```

4. Submit kode untuk impementasi Authorization pada repository github kalian

Implementasi Multi-Level Auth...

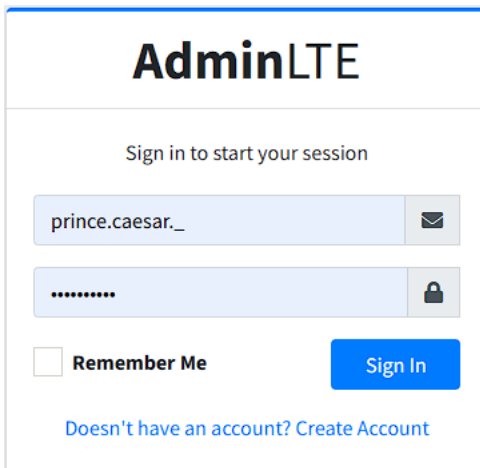
 FarrelCaesarian • just now



#### 6. Tugas 4 – Implementasi Form Registrasi :

1. Silahkan implementasikan form untuk registrasi user.
2. Screenshot hasil yang kalian kerjakan
3. Commit dan push hasil tugas kalian ke masing-masing repo github kalian

Jawab:



# AdminLTE

Create Account to start your session

Create Account

Already have an account? [Login here](#)

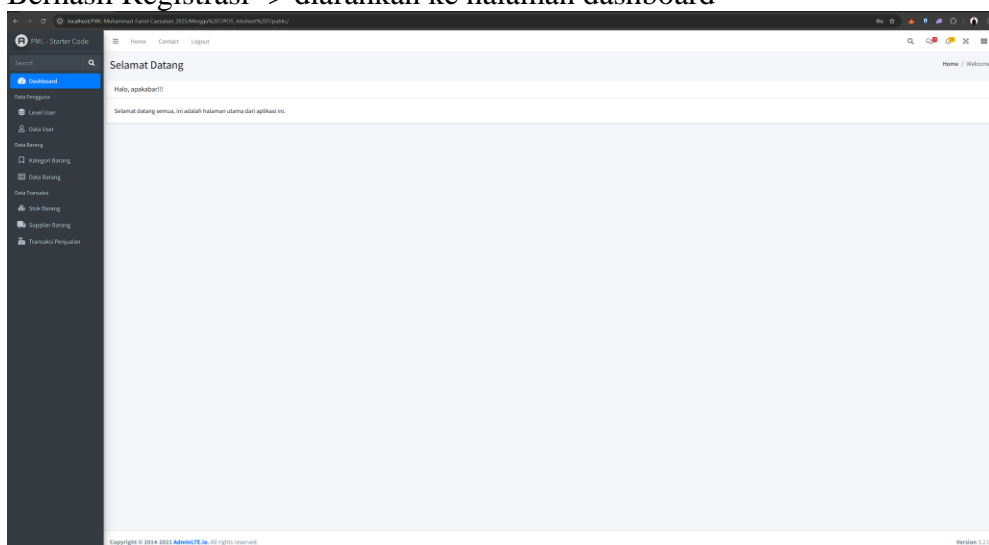
# AdminLTE

Create Account to start your session

Create Account

Already have an account? [Login here](#)

Berhasil Registrasi -> diarahkan ke halaman dashboard



Data terekam di table m\_user

<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	37	1	FarrelRegister	FarrelRegister	\$2y\$12\$cRaBhE5sa4QDS6Q/aJceuEuMvysS5AamBzrNJXlm9...	2025-03-30 10:03:33	2025-03-30 10:03:33
--------------------------	----------------------	----------------------	------------------------	----	---	----------------	----------------	--	---------------------	---------------------

Source code:

## Auth Controller

```
public function showRegistrationForm()
{
    return view('auth.register');
}

public function register(Request $request)
{
    $validator = Validator::make($request->all(), [
        'username' => 'required|string|max:255|unique:m_user',
        'nama' => 'required|string|max:255',
        'password' => 'required|string|min:6|confirmed',
    ]);

    if ($validator->fails()) {
        return redirect()->back()->withErrors($validator)->withInput();
    }

    $user = UserModel::create([
        'username' => $request->username,
        'nama' => $request->nama,
        'password' => Hash::make($request->password),
        'level_id' => 1, // Level ID Admin
    ]);

    Auth::login($user);

    return redirect('/')->with('success', 'Registrasi berhasil!');
}
```

## Register.blade.php (lebih lengkapnya di GitHub)

```
resources > views > auth > register.blade.php > html > body.hold-transition.login-page > div.login-box >
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Login Pengguna</title>
7 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:400,700,900">
8 <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}">
9 <link rel="stylesheet" href="{{ asset('adminlte/plugins/icheck-bootstrap/icheck-bootstrap.min.css') }}">
10 <link rel="stylesheet" href="{{ asset('adminlte/plugins/sweetalert2-theme-bootstrap/sweetalert2-theme-bootstrap.min.css') }}">
11 <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}">
12 </head>
13 <body class="hold-transition login-page">
14 <div class="login-box">
15 <div class="card card-outline card-primary">
16 <div class="card-header text-center">
17 <a href="{{ url('/') }}" class="h1"><b>Admin</b>LTE</a>
18 </div>
19 <div class="card-body">
20 <p class="login-box-msg">Create Account to start your session</p>
21 <form action="{{ url('register') }}" method="POST" id="form-register">
22 @csrf
23 <div class="input-group mb-3">
24 <input type="text" id="username" name="username" class="form-control">
25 <div class="input-group-append">
26 <div class="input-group-text">
27 <span class="fas fa-envelope"></span>
28 </div>
29 </div>
30 <small id="error-username" class="error-text text-danger"></small>
```

## Route

```
//Registrasi
Route::get('/register', [AuthController::class, 'showRegistrationForm'])->name('register');
Route::post('/register', [AuthController::class, 'register']);
```