



NAMA : Dika Arie Arrifky

NIM : 2341720232

KELAS : TI-2A

MATERI : JOBSHEET 07 Authentication dan Authorization di Laravel

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

```
61
62     'providers' => [
63         'users' => [
64             'driver' => 'eloquent',
65             'model' => App\Models\UserModel::class,
66         ],
67
68         // 'users' => [
69             //     'driver' => 'database',
70             //     'table' => 'users',
71             // ],
72     ],
73
```

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

```
24 //jobsheet 7
25 77 references | 0 implementations | Codeium: Refactor | Explain
26 class UserModel extends Authenticatable
27 {
28     use HasFactory;
29
30     0 references
31     protected $table = 'm_user';
32     0 references
33     protected $primaryKey = 'user_id';
34
35     0 references
36     protected $fillable = ['username', 'password', 'nama', 'level_id', 'created_at', 'updated_at'];
37
38     0 references
39     protected $hidden = ['password', '']; // jangan ditampilkan saat select
40
41     0 references
42     protected $casts = ['password' => 'hashed']; //casting password agar otomatis di hash
43
44     0 references | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | X
45     public function level(): BelongsTo
46     {
47         {
48             return $this->belongsTo(LevelModel::class, 'level_id', 'level_id');
49         }
50     }
51 }
```



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : JOBSHEET 07 Authentication dan Authorization di Laravel

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

```
auth.php U  UserModel.php 2, U  AuthController.php X
Minggu 7 > PWL_POS > app > Http > Controllers > AuthController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\Auth;
7
8  0 references | 0 implementations | Codeium: Refactor | Explain
9  class AuthController extends Controller
10 {
11     0 references | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | X
12     public function login(): Factory|Redirector|RedirectResponse|View
13     {
14         if (Auth::check()) { // jika sudah login, maka redirect ke halaman home
15             return redirect(to: '/');
16         }
17         return view(view: 'auth.login');
18     }
19     0 references | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | X
20     public function postlogin(Request $request): JsonResponse|mixed|Redirector|RedirectRes...
21     {
22         if ($request->ajax() || $request->wantsJson()) {
23             $credentials = $request->only('username', 'password');
24             if (Auth::attempt($credentials)) {
25                 return response()->json([
26                     'status' => true,
27                     'message' => 'Login Berhasil',
28                     'redirect' => url(path: '/')
29                 ]);
30             }
31             return response()->json([
32                 'status' => false,
33                 'message' => 'Login Gagal'
34             ]);
35         }
36         return redirect(to: 'login');
37     }
38     0 references | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | X
39     public function logout(Request $request): Redirector|RedirectResponse
40     {
41         Auth::logout();
42         $request->session()->invalidate();
43         $request->session()->regenerateToken();
44         return redirect(to: 'login');
45     }
46 }
```

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)



NAMA : Dika Arie Arrifky

NIM : 2341720232

KELAS : TI-2A

MATERI : JOBSHEET 07 Authentication dan Authorization di Laravel

```
Minggu 7 > PWL_POS > resources > views > auth > login.blade.php > html > body.hold-transition.login-page > script > ready() callback > unhighlight
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"
10     href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&display=fallback">
11   <!-- Font Awesome -->
12   <link rel="stylesheet" href="{{ asset('adminlte/plugins/fontawesome-free/css/all.min.css') }}">
13   <!-- icheck bootstrap -->
14   <link rel="stylesheet" href="{{ asset('adminlte/plugins/icheck-bootstrap/icheck-bootstrap.min.css') }}">
15   <!-- SweetAlert2 -->
16   <link rel="stylesheet" href="{{ asset('adminlte/plugins/sweetalert2-theme-bootstrap-4/bootstrap-4.min.css') }}">
17   <!-- Theme style -->
18   <link rel="stylesheet" href="{{ asset('adminlte/dist/css/adminlte.min.css') }}">
19 </head>
20
21 <body class="hold-transition login-page">
22   <div class="login-box">
23     <!-- /.login-logo -->
24     <div class="card card-outline card-primary">
25       <div class="card-header text-center"><a href="{{ url(path: '/') }}"
26         class="h1"><b>Admin</b></a></div>
27       <div class="card-body">
28         <p class="login-box-msg">Sign in to start your session</p>
29         <form action="{{ url(path: '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"
33               placeholder="Username">
34             <div class="input-group-append">
35               <div class="input-group-text">
36                 <span class="fas fa-envelope"></span>
37               </div>
38             </div>
39             <small id="error-username" class="error-text text-danger"></small>
40           </div>
41           <div class="input-group mb-3">
42             <input type="password" id="password" name="password" class="form-control"
43               placeholder="Password">
44             <div class="input-group-append">
45               <div class="input-group-text">
```



NAMA : Dika Arie Arrifky

NIM : 2341720232

KELAS : TI-2A

MATERI : JOBSHEET 07 Authentication dan Authorization di Laravel

```
Minggu 7 > PWL_POS > resources > views > auth > login.blade.php > html > body,hold-transition.login-page > script > ready() callback > unhighlight
2 <html lang="en">
21 <body class="hold-transition login-page">
22 <div class="login-box">
24 <div class="card card-outline card-primary">
27 <div class="card-body">
46 <div>
47 <div>
48 <div>
49 <small id="error-password" class="error-text text-danger"></small>
50 </div>
51 <div class="row">
52 <div class="col-8">
53 <div class="icheck-primary">
54 <input type="checkbox" id="remember"><label for="remember">Remember Me</label>
55 </div>
56 </div>
57 <!-- /.col -->
58 <div class="col-4">
59 <button type="submit" class="btn btn-primary btn-block">Sign In</button>
60 </div>
61 <!-- /.col -->
62 </div>
63 </form>
64 </div>
65 <!-- /.card-body -->
66 </div>
67 <!-- /.card -->
68 </div>
69 <!-- /.login-box -->
70 <!-- jQuery -->
71 <script src="{{ asset(path: 'adminlte/plugins/jquery/jquery.min.js') }}"></script>
72 <!-- Bootstrap 4 -->
73 <script src="{{ asset(path: 'adminlte/plugins/bootstrap/js/bootstrap.bundle.min.js') }}"></script>
74 <!-- jquery-validation -->
75 <script src="{{ asset(path: 'adminlte/plugins/jquery-validation/jquery.validate.min.js') }}"></script>
76 <script src="{{ asset(path: 'adminlte/plugins/jquery-validation/additional-methods.min.js') }}"></script>
77 <!-- SweetAlert2 -->
78 <script src="{{ asset(path: 'adminlte/plugins/sweetalert2/sweetalert2.min.js') }}"></script>
79 <!-- AdminLTE App -->
80 <script src="{{ asset(path: 'adminlte/dist/js/adminlte.min.js') }}"></script>
81 <script>
82 $.ajaxSetup({
83 headers: {
84 'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
85 }
```



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : JOBSHEET 07 Authentication dan Authorization di Laravel

```
Minggu 7 > PWL_POS > resources > views > auth > login.blade.php > html > body.hold-transition.login-page > script > ready() callback > unhighlight
2 <html lang="en">
21 <body class="hold-transition login-page">
81 <script>
82     $.ajaxSetup({
83         headers: {
84             'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
85         }
86     });
87     $(document).ready(function() {
88         $('#form-login').validate({
89             rules: {
90                 username: {
91                     required: true,
92                     minlength: 4,
93                     maxlength: 20
94                 },
95                 password: {
96                     required: true,
97                     minlength: 6,
98                     maxlength: 20
99                 }
100             },
101             submitHandler: function(form) { // ketika valid, maka bagian yg akan dijalankan
102                 $.ajax({
103                     url: form.action,
104                     type: form.method,
105                     data: $(form).serialize(),
106                     success: function(response) {
107                         if (response.status) { // jika sukses
108                             Swal.fire({
109                                 icon: 'success',
110                                 title: 'Berhasil',
111                                 text: response.message,
112                             }).then(function() {
113                                 window.location = response.redirect;
114                             });
115                         } else { // jika error
116                             $('.error-text').text('');
117                             $.each(response.msgField, function(prefix, val) {
118                                 $('#error-' + prefix).text(val[0]);
119                             });
120                             Swal.fire({
121                                 icon: 'error',
122                                 title: 'Terjadi Kesalahan',
123                                 text: response.message
124                             });
125                         }
126                     }
127                 });
128                 return false;
129             },
130             errorElement: 'span',
131             errorPlacement: function(error, element) {
132                 error.addClass('invalid-feedback');
133                 element.closest('.input-group').append(error);
134             },
135             highlight: function(element, errorClass, validClass) {
136                 $(element).addClass('is-invalid');
137             },
138             unhighlight: function(element, errorClass, validClass) {
139                 $(element).removeClass('is-invalid');
140             }
141         });
142     });
143 </script>
144 </body>
145
146 </html>
```

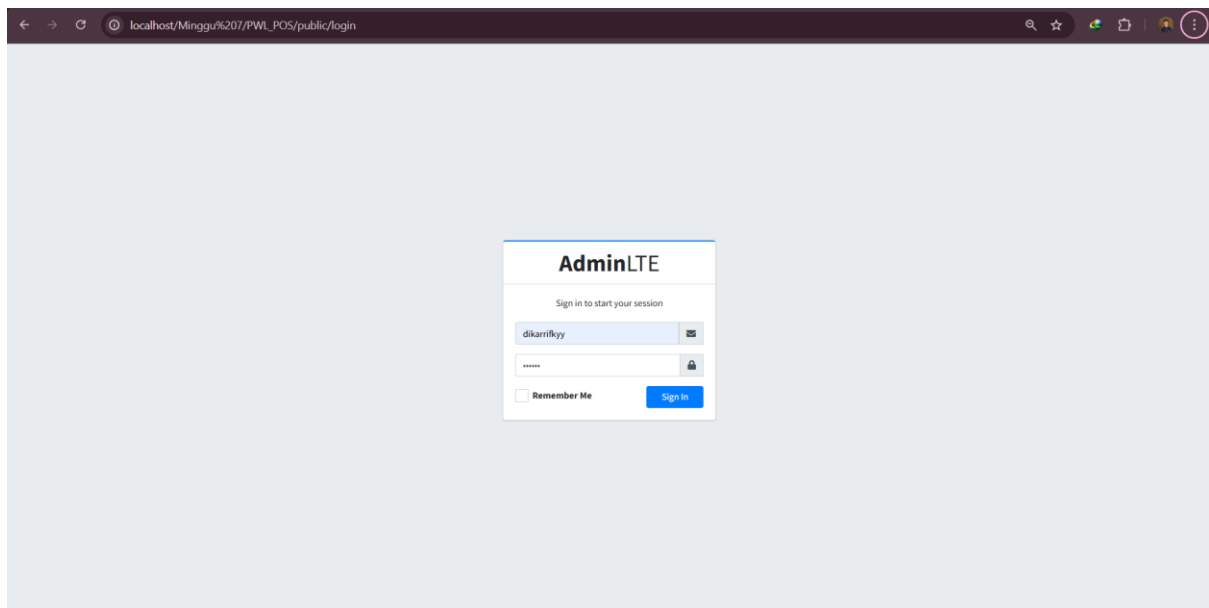
5. Kemudian kita modifikasi route/web.php agar semua route masuk dalam auth



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : JOBSHEET 07 Authentication dan Authorization di Laravel

```
Minggu 7 > PWL_POS > routes > web.php > ...
1  <?php
2
3  use App\Http\Controllers\WelcomeController;
4  use Illuminate\Support\Facades\Route;
5  use App\Http\Controllers\LevelController;
6  use App\Http\Controllers\KategoriController;
7  use App\Http\Controllers\UserController;
8  use App\Http\Controllers\BarangController;
9  use App\Http\Controllers\SupplierController;
10 use App\Http\Controllers\AuthController;
11
12 Route::pattern('id', '[0-9]+'); //ketika ada parameter id , maka harus berupa angka
13
14
15 Route::get('login', [AuthController::class, 'login'])->name('login');
16 Route::post('login', [AuthController::class, 'postlogin']);
17 Route::get('logout', [AuthController::class, 'logout'])->middleware('auth');
18
19 Route::middleware(['auth'])->group(function () { // artinya semua route di dalam group ini harus login dulu
20     //masukkan semua route yang membutuhkan auth di sini
21     // Welcome
22
23 });
24 /*
25
```

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



Tugas 1 – Implementasi Authentication :

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



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : JOBSHEET 07 Authentication dan Authorization di Laravel

Sudah

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

Sudah

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

Sudah saya commit

Praktikum 2 – Implementasi Authorizaton 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

```
43  
44 //praktikum 2 Implementasi Authorizaton di Laravel dengan Middleware  
0 references | 0 overrides | Codeium: Refactor | Explain | ×  
45 public function getRoleName(): string  
46 {  
47     return $this->level->level_name;  
48 }  
49  
0 references | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | ×  
50 public function hasRole($role): bool  
51 {  
52     return $this->level->level_kode == $role;  
53 }  
54 }  
55
```

2. Kemudian kita buat middleware dengan nama AuthorizeUser.php. Kita bisa buat middleware dengan mengetikkan perintah pada terminal/CMD
php artisan make:middleware AuthorizeUser



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : JOBSHEET 07 Authentication dan Authorization di Laravel

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS
PS C:\laragon\www\P1> cd '.\Minggu 7\'
PS C:\laragon\www\P1\Minggu 7> cd '.\PWL_POS\'
PS C:\laragon\www\P1\Minggu 7\PWL_POS> php artisan make:middleware AuthorizeUser

Warning: Module "pdo_mysql" is already loaded in Unknown on line 0

INFO Middleware [C:\laragon\www\P1\Minggu 7\PWL_POS\app\Http\Middleware\AuthorizeUser.php] created successfully.
PS C:\laragon\www\P1\Minggu 7\PWL_POS>
```

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

```
Minggu 7 > PWL_POS > app > Http > Middleware > AuthorizeUser.php > PHP > AuthorizeUser > handle()
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         $user = $request->user(); // ambil data user yg login
19         //fungsi user() diambil dari UserModel.php
20         if ($user->hasRole($role)) { //cek apakah user punya role yang diinginkan
21             return $next($request);
22         }
23
24         //jika tidak punya role, maka tampilkan error 403
25         abort(403, 'forbidden, kamu tidak punya akses ke halaman ini');
26     }
27 }
28
```

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

```
58
59
60 'auth' => \App\Http\Middleware\Authenticate::class,
61 'authorize' => \App\Http\Middleware\AuthorizeUser::class, // middle ware yg kita buat
62 'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
63 'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,
64
65 }
```




NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : JOBSHEET 07 Authentication dan Authorization di Laravel

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

| Extra options | | | | | | |
|--------------------------|------------------|----------|------------|---------------|---------------------|------------|
| ← T → | | | | | | |
| | | level_id | level_kode | level_nama | created_at | updated_at |
| <input type="checkbox"/> | Edit Copy Delete | 1 | ADM | Administrator | NULL | NULL |
| <input type="checkbox"/> | Edit Copy Delete | 2 | MNG | Manager | NULL | NULL |
| <input type="checkbox"/> | Edit Copy Delete | 3 | STF | Staff/Kasir | NULL | NULL |
| <input type="checkbox"/> | Edit Copy Delete | 4 | CUS | Customer | 2025-03-01 10:42:39 | 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. 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.

```
19
20 Route::middleware(['auth'])->group(function (): void { // artinya semua route di dalam group ini harus login dulu
21     //masukkan semua route yang membutuhkan auth di sini
22     Route::get('/', [WelcomeController::class, 'index']);
23     //route level
24
25     //artinya semua route di dalam group ini harus punya role ADM (Administrator)
26     Route::middleware(['authorize:ADM'])->group(function (): void {
27         Route::get('/level', [LevelController::class, 'index']);
28         Route::post('/level/list', [LevelController::class, 'list']);
29         Route::get('/level/create', [LevelController::class, 'create']);
30         Route::post('/level', [LevelController::class, 'store']);
31         Route::get('/level/{id}/edit', [LevelController::class, 'edit']);
32         Route::put('/level/{id}', [LevelController::class, 'update']);
33         Route::delete('/level/{id}', [LevelController::class, 'destroy']);
34     });
35
36 });
37
```

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

Tidak bisa login untuk selain role ADM(Administrator)



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : JOBSHEET 07 Authentication dan Authorization di Laravel





NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : JOBSHEET 07 Authentication dan Authorization di Laravel

Tugas 2 – Implementasi Authoriization :

1. Apa yang kalian pahami pada praktikum 2 ini?

Membuat system login dan crud untuk role administrator

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

Sudah

3. Submit kode untuk impementasi Authorization pada repository github kalian.

Sudah saya commit



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : JOBSHEET 07 Authentication dan Authorization di Laravel

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()

```
55 //Praktikum 3 – Implementasi Multi-Level Authorizaton di Laravel dengan Middleware
56
0 references | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | X
57 public function getRole(): mixed
58 {
59     return $this->level->level_kode;
60 }
61
62
```

2. Selanjutnya, Kita modifikasi middleware AuthorizeUser.php dengan kode berikut

```
16 0 references | 0 overrides | Codeium: Refactor | Explain | X
17 public function handle(Request $request, Closure $next, ... $roles): Response
18 {
19     //praktikum 2
20     // $user = $request->user(); // ambil data user yg login
21     // //fungsi user() diambil dari UserModel.php
22     // if ($user->hasRole($role)) { //cek apakah user punya role yang diinginkan
23     //     return $next($request);
24     // }
25
26     //praktikum 3
27     $user_role = $request->user()->getRole(); //ambil data level_kode dari user yg login
28     if(in_array(needle: $user_role, haystack: $roles)) // cek apakah level_kode user ada di array $roles
29     {
30         return $next($request); // jika ada maka lanjutkan request
31     }
32
33     //jika tidak punya role, maka tampilkan eror 403
34     abort(code: 403,message: 'Forbidden, kamu tidak punya akses ke halaman ini');
}
```



NAMA : Dika Arie Arrifky

NIM : 2341720232

KELAS : TI-2A

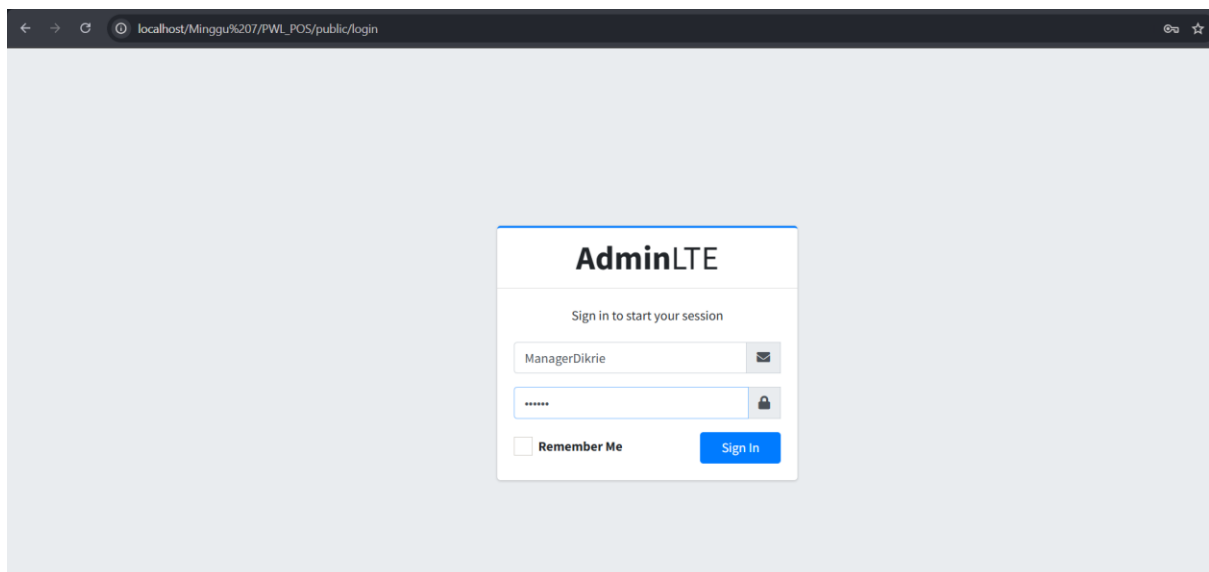
MATERI : JOBSHEET 07 Authentication dan Authorization di Laravel

3. Setelah itu tinggal kita perbaiki route/web.php sesuaikan dengan role/level yang diinginkan. Contoh

```
37
38 //praktikum 3
39 Route::get('/', [BarangController::class, 'index']);
40 Route::post('/list', [BarangController::class, 'list']);
41 Route::get('/create_ajax', [BarangController::class, 'create_ajax']);
42 Route::post('/ajax', [BarangController::class, 'store_ajax']);
43 Route::get('/{id}/show_ajax', [BarangController::class, 'show_ajax']);
44 Route::get('/{id}/edit_ajax', [BarangController::class, 'edit_ajax']);
45 Route::put('/{id}/update_ajax', [BarangController::class, 'update_ajax']);
46 Route::get('/{id}/delete_ajax', [BarangController::class, 'confirm_ajax']);
47 Route::delete('/{id}/delete_ajax', [BarangController::class, 'delete_ajax']);
48
49 });
50
```

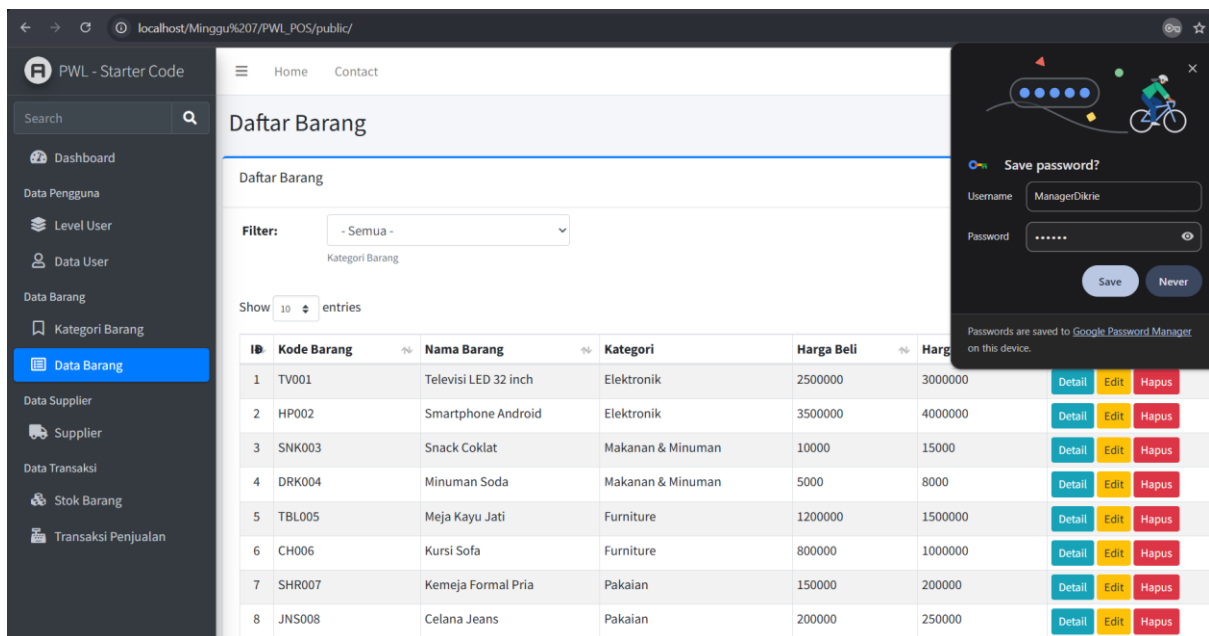
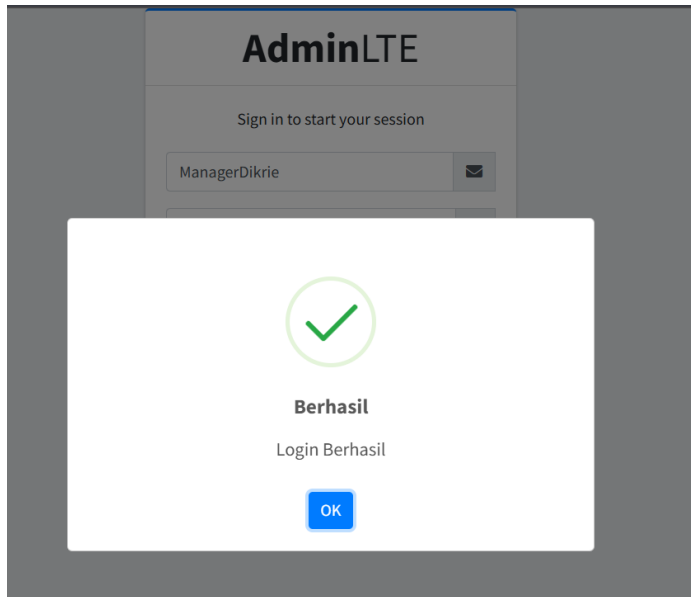
4. Sekarang kita sudah bisa memberikan hak akses menu/route ke beberapa level user

Bisa login menggunakan role Manager





NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : JOBSHEET 07 Authentication dan Authorization di Laravel





NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : JOBSHEET 07 Authentication dan Authorization di Laravel

Tugas 3 – Implementasi Multi-Level Authorization :

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

Sudah

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

Jawab : pengguna dengan role ADM login dengan menggunakan username dan password sesuai dengan di database

3. Implementasikan multi-level authorization untuk semua Level/Jenis User dan Menu-menu yang sesuai dengan Level/Jenis User
4. Submit kode untuk impementasi Authorization pada repository github kalian.

Sudah

Tugas 4 – Implementasi Form Registrasi :

1. Silahkan implementasikan form untuk registrasi user.

Sudah

2. Screenshot hasil yang kalian kerjakan

Sudah

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

Sudah