

PRAKTIKUM PEMROGRAMAN WEB

Laporan Praktikum Jobsheet 6 Dan 7



Dosen Pengampu: Dr. Ika Parma Dewi, M.Pd.T.

Oleh :

Rian Septiawan 23076052

PRODI PENDIDIKAN TEKNIK INFORMATIKA

DEPARTEMEN ELEKTRONIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI PADANG

2025

FT UNP Padang	Lembaran : JOBSHEET 6 - 7
Jurusan : Pendidikan Teknik Informatika	Matakuliah: Pratikum Pemrograman Web
Waktu : 3 x 50 Menit	Topik : Data User
Kode : TIK1.61.4318	Judul : Data User

Fakultas	Proram Studi	Kode MK	Waktu
Teknik	Pendidikan Teknik Informatika	TIK1.61.4318	2 x 170 Menit

A. LEARNING OUTCOMES

Setelah mengikuti perkuliahan ini diharapkan mahasiswa mampu :

1. Dapat menampilkan tabel dari database
2. Dapat mengelola database
3. Dapat membuat projek dengan Laravel.

B. ALAT DAN BAHAN

1. PC/Laptop
2. Browser Internet (Internet Explorer/Mozilla Firefox/Google Chrome)
3. XAMPP/Laragon
4. Text Editor (Visual Studio Code)

C. TEORI SINGKAT

Auth (Authentication) di Laravel adalah fitur yang digunakan untuk mengelola autentikasi pengguna, termasuk login, logout, dan perlindungan akses ke halaman tertentu.

1. Fungsi Utama Auth

1. Memeriksa Status Login

Mengecek apakah pengguna sudah login atau belum.

2. Mengambil Data Pengguna yang Sedang Login

Memungkinkan sistem untuk mengakses informasi pengguna yang sedang aktif.

3. Autentikasi dengan Middleware

Melindungi rute atau halaman agar hanya bisa diakses oleh pengguna yang sudah login.

4. Logout Pengguna

Menghapus sesi pengguna dan mengarahkannya kembali ke halaman login.

5. Login Manual dengan Kredensial

Memproses login secara manual dengan mencocokkan data pengguna di database.

2. Cara Kerja Auth di Laravel

1. Menggunakan Middleware auth

Laravel memiliki middleware auth yang memastikan halaman tertentu hanya bisa diakses oleh pengguna yang sudah login.

2. Mengelola Akses Berdasarkan Peran (Roles)

Bisa digunakan untuk membatasi akses pengguna berdasarkan peran tertentu, misalnya **admin**, **petugas**, atau **pasien**.

3. Pembuatan dan Pengelolaan Sesi Pengguna

Setelah pengguna berhasil login, Laravel akan menyimpan informasi pengguna dalam sesi untuk menjaga status login selama sesi aktif.

4. Penyimpanan dan Keamanan Password

Laravel menggunakan hashing untuk menyimpan password di database, sehingga lebih aman.

3. Kesimpulan

- **Auth** di Laravel membantu mengelola proses login, logout, dan otorisasi pengguna.
- **Middleware auth** memastikan halaman tertentu hanya bisa diakses oleh pengguna yang sudah login.
- **Mendukung sistem role-based access control (RBAC)** untuk mengelola akses berdasarkan peran pengguna.

D. LANGKAH KERJA

4.1. Membuat Tabel User

1. Database user

```
PS C:\Users\ryans\Documents\PROJECT DEVELOP\A Github\Rumah-Sakit-PHP> php artisan make:migration create_user_table
<!-- web.php -->

INFO Migration [C:\Users\ryans\Documents\PROJECT DEVELOP\A Github\Rumah-Sakit-PHP\database\Migrations\2025_03_04_064545_create_user_table.php] created successfully.

database > migrations > 2025_03_04_064545_create_user_table.php > class > up
7 return new class extends Migration
14 public function up()
15 {
16     Schema::create('user', function (Blueprint $table) {
17         $table->id(); // Menambahkan kolom id (primary key)
18         $table->string('nama_user'); // Menambahkan kolom nama_user
19         $table->string('username')->unique(); // Menambahkan kolom username dengan constraint unique
20         $table->string('password'); // Menambahkan kolom password
21         $table->string('no_telepon'); // Menambahkan kolom no_telepon
22         $table->string('foto_user')->nullable(); // Menambahkan kolom foto_user, nullable jika tidak di
23         $table->enum('roles', ['admin', 'petugas', 'pasien']); // Menambahkan kolom roles dengan nilai
24         $table->timestamps(); // Menambahkan kolom created_at dan updated_at
25     });
26 }
```

```
PS C:\Users\ryans\Documents\PROJECT DEVELOP\A Github\Rumah-Sakit-PHP> php artisan migrate
<!-- web.php -->

INFO Running migrations.

2025_03_04_064545_create_user_table ..... 78ms DONE

PS C:\Users\ryans\Documents\PROJECT DEVELOP\A Github\Rumah-Sakit-PHP>
```

2. Models User.php

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Database\Eloquent\Factories\HasFactory;
7 use Illuminate\Foundation\Auth\User as Authenticatable;
8 use Illuminate\Notifications\Notifiable;
9 use Laravel\Sanctum\HasApiTokens;
10
11 class User extends Authenticatable
12 {
13     use HasApiTokens, HasFactory, Notifiable;
14
15     /**
16      * The attributes that are mass assignable.
17      *
18      * @var array<int, string>
19      */
```

```
20     protected $fillable = [
21         'nama_user',
22         'username',
23         'password',
24         'no_telepon',
25         'foto_user',
26         'roles',
27     ];
28
29     /**
30      * The attributes that should be hidden for serialization.
31      *
32      * @var array<int, string>
33      */
34     protected $hidden = [
35         'password',
36         'remember_token',
37     ];
38
39     /**
40      * The attributes that should be cast.
41      *
42      * @var array<string, string>
43      */
44     protected $casts = [
45         'email_verified_at' => 'datetime',
46     ];
47 }
```

3. Menyesuaikan Auth untuk Menggunakan username

Di dalam file **config/auth.php**, ubah **providers** untuk menggunakan **username** sebagai autentikasi:

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

```

4. Kemudian buat Controllers dengan nama **LoginController.php** letakkan dalam folder **Auth**, **app/Http/Controllers/Auth/LoginController.php**

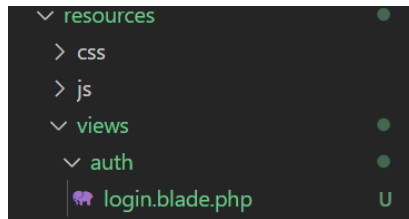
```
PS C:\Users\ryans\Documents\PROJECT DEVELOP\A Github\Rumah-Sakit-PHP> php artisan make:controller Auth/LoginController
<!-- web.php -->
INFO Controller [C:\Users\ryans\Documents\PROJECT DEVELOP\A Github\Rumah-Sakit-PHP\app\Http\Controllers\Auth/LoginController.php] created successfully.
PS C:\Users\ryans\Documents\PROJECT DEVELOP\A Github\Rumah-Sakit-PHP>
```

```
1  <?php
2
3  namespace App\Http\Controllers\Auth;
4
5  use App\Http\Controllers\Controller;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Auth;
8
9  class LoginController extends Controller
10 {
11     public function username()
12     {
13         return 'username';
14     }
15
16     public function showLoginForm()
17     {
18         return view('auth.login');
19     }
20
21     public function login(Request $request)
22     {
23         $credentials = $request->only('username', 'password');
24
25         if (Auth::attempt($credentials)) {
```

```
26         $user = Auth::user();
27         return $this->authenticated($request, $user);
28     }
29
30     return redirect()->back()->withErrors([
31         'username' => 'The provided credentials do not match our
records.',
32     ]);
33 }
34
35 protected function authenticated(Request $request, $user)
36 {
37     if ($user->roles == 'admin') {
38         return redirect('/dashboard-admin');
39     } elseif ($user->roles == 'petugas') {
40         return redirect('/dashboard-petugas');
41     } elseif ($user->roles == 'pasien') {
42         return redirect('/dashboard-pasien');
43     }
44 }
45 }
```

4.2. Membuat Halaman Login

1. Buat file view untuk login di **resources/views/auth/login.blade.php**



2. Buat source code ini untuk tampilan halaman login

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-
6         scale=1.0">
7     <title>Login</title>
8     <link rel="stylesheet"
9         href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.m
10         in.css">
11     <link rel="stylesheet"
12         href="https://cdnjs.cloudflare.com/ajax/libs/font-
13         awesome/5.15.4/css/all.min.css">
14     <style>
15         body {
16             background-image: url('/rsaule.jpg');
17             background-size: cover;
18             background-repeat: no-repeat;
19             background-position: center;
20             height: 100vh;
21             display: flex;
22             align-items: center;
23             justify-content: center;
24             color: #333;
25         }
26         .card {
27             background-color: rgba(255, 255, 255, 0.9);
28             border: none;
29             border-radius: 10px;
30             box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
31             padding: 20px;
32         }
33         .card-header {
34             background-color: transparent;
```



```
30         color: #4f6b69;
31         font-size: 24px;
32         text-align: center;
33         border-bottom: none;
34         margin-bottom: 20px;
35     }
36     .form-control {
37         border: 1px solid #d1d3d2;
38         border-radius: 5px;
39     }
40     .password-container {
41         position: relative;
42     }
43     .password-toggle {
44         position: absolute;
45         right: 10px;
46         top: 70%;
47         transform: translate(-50%);
48         cursor: pointer;
49         color: #6c757d;
50         font-size: 18px;
51         padding: 0;
52         line-height: 1;
53     }
54     .btn-primary {
55         background-color: #4e73df;
56         border: none;
57         border-radius: 5px;
58         width: 100%;
59         padding: 10px;
60         font-size: 16px;
61         color: #ffffff;
62     }
63     .btn-info {
64         background-color: #2caec8;
65         border: none;
66         border-radius: 5px;
67         width: 100%;
68         padding: 10px;
69         font-size: 16px;
70         color: #ffffff;
71     }
72     .btn-primary:hover {
73         background-color: #375a7f;
74     }
```

```

75     .text-muted {
76         text-align: center;
77         display: block;
78         margin-top: 10px;
79         color: #858796;
80     }
81     .text-muted a {
82         color: #4e73df;
83         text-decoration: none;
84     }
85     .text-muted a:hover {
86         text-decoration: underline;
87     }
88 </style>
89 </head>
90 <body>
91     <div class="card col-md-4 col-sm-6">
92         <div class="card-header">LOGIN</div>
93         <div class="card-body">
94             @if($errors->any())
95                 <div class="alert alert-danger">
96                     <ul>
97                         @foreach($errors->all() as $error)
98                             <li>{{ $error }}</li>
99                         @endforeach
100                     </ul>
101                 </div>
102             @endif
103             <form method="POST" action="{{ route('login') }}">
104                 @csrf
105                 <div class="form-group">
106                     <label for="username">Username</label>
107                     <input type="text" class="form-control"
108 id="username" name="username" placeholder="Enter Username" required
109 autofocus>
110                 </div>
111                 <div class="form-group password-container">
112                     <label for="password">Password</label>
113                     <input type="password" class="form-control"
114 id="password" name="password" placeholder="Enter Password" required>
115                     <span class="password-toggle" id="toggle-icon"
116 onclick="togglePassword()">
117                         <i class="fa fa-eye"></i>
118                     </span>
119                 </div>

```

```
116         <button type="submit" class="btn btn-  
primary">Login</button>  
117         <a href="#" class="btn btn-info">Forgot Password?</a>  
118     </form>  
119 </div>  
120 </div>  
121  
122 <script>  
123     function togglePassword() {  
124         var passwordField = document.getElementById("password");  
125         var toggleIcon = document.getElementById("toggle-icon");  
126         if (passwordField.type === "password") {  
127             passwordField.type = "text";  
128             toggleIcon.classList.remove("fa-eye");  
129             toggleIcon.classList.add("fa-eye-slash");  
130         } else {  
131             passwordField.type = "password";  
132             toggleIcon.classList.remove("fa-eye-slash");  
133             toggleIcon.classList.add("fa-eye");  
134         }  
135     }  
136 </script>  
137 </body>  
138 </html>
```

4.3. Menambah Middleware untuk Role

1. Buat middleware baru dengan perintah: **php artisan make:middleware RoleMiddleware**

```
PS C:\Users\ryans\Documents\PROJECT DEVELOP\A Github\Rumah-Sakit-PHP> php artisan make:middleware RoleMiddleware
<!-- web.php -->

INFO Middleware [C:\Users\ryans\Documents\PROJECT DEVELOP\A Github\Rumah-Sakit-PHP\app\Http\Middleware\RoleMiddleware.php] created successfully.
PS C:\Users\ryans\Documents\PROJECT DEVELOP\A Github\Rumah-Sakit-PHP>
```

2. Daftarkan middleware ini di **app/Http/Kernel.php**

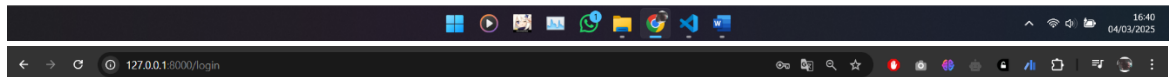
```
protected $routeMiddleware = [
    // ...
    'role' => \App\Http\Middleware\RoleMiddleware::class,
];
```

```
57     protected $routeMiddleware = [
58         'auth' => \App\Http\Middleware\Authenticate::class,
59         'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
60         'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
61         'can' => \Illuminate\Auth\Middleware\Authorize::class,
62         'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
63         'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
64         'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
65         'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
66         'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
67         'role' => \App\Http\Middleware\RoleMiddleware::class,
68     ];
```

4.4. Web.php

```
57 // Route untuk login
58 Route::get('login', [LoginController::class, 'showLoginForm'])->name('login');
59 Route::post('login', [LoginController::class, 'login']);
60
61 Route::middleware(['redirect.if.authenticated'])->group(function () {
62     Route::get('/login', [LoginController::class, 'showLoginForm'])->name('login');
63     // Tambahkan rute lain yang ingin Anda lindungi
64 });
```

Buka link <http://127.0.0.1:8000/login> setelah menjalankan perintah **php artisan serve**

A login form for "Rumah Sakit Dr. Rian". It features a title "Rumah Sakit" in blue and "Dr. Rian" below it. The form has two input fields: "Username" and "Password" (with a toggle icon). Below the inputs are two buttons: a blue "Login" button and a teal "Forgot Password?" button.The same login form as above, but with an error message displayed in a pink box: "The provided credentials do not match our records." The form fields and buttons remain the same.

E. LEMBAR KERJA

1. Lakukan praktikum sesuai langkah kerja yang diberikan
2. Buat laporannya

F. DAFTAR PUSTAKA

- [1] Deitel. 2001 Internet & WWW How to Programing, Prentice Hall
- [2]. Budi Raharjo. 2012. Modul Prmograman WEB. Modula: Bandung.
- [3]. Bunafit Nugroho. 2004. PHP & MySQL Dengan Editor Dreamweaver MX. Andi Yogyakarta
- [4] Sklar, David, and Adam Trachtenberg. PHP Cookbook. O'Reilly, 2008.
- [5] Welling, Luke. PHP and MySQL Web Development. Developers Library, 2012
- [6] Yank, Kevin, and Cameron Adams. Simply Javascript. Sitepoint, 2010
- [7] Manual PHP. 2014
- [8] Niederst, Jennifer. Learning Web Design (4th edition). O'Reilly. 2013.