

# Modul Pertemuan 5

## Tujuan Pembelajaran

Tujuan pembelajaran yang harus dicapai mahasiswa dalam modul ini adalah :

- Memahami proses autentikasi pengguna
- Membuat halaman registrasi khusus untuk pasien
- Menerapkan proses login pasien menggunakan email dan password
- Mengatur hak akses pengguna berdasarkan role
- Memahami alur bisnis proses jadwal pemeriksaan pasien di sistem informasi klinik.
- Mengimplementasikan CRUD data jadwal pemeriksaan dengan Eloquent ORM Laravel.
- Mengelola jadwal pemeriksaan melalui antarmuka web terhubung database.
- Memahami alur bisnis proses pendaftaran pasien ke poli.
- Mampu mengimplementasikan relasi antar model dengan Eloquent ORM Laravel.
- Mampu menambahkan data ke database.

---

## 10. Register dan Login Pasien

### 10.1 Overview

Pada modul ini kita akan mempelajari bagaimana membangun sistem autentikasi yang aman dan terstruktur yang aman dan terstruktur dalam sebuah aplikasi. Mahasiswa akan diajak untuk memahami konsep dasar autentikasi pengguna, mulai dari proses registrasi hingga login. Fokus utama modul ini adalah membuat halaman registrasi khusus bagi pasien, sehingga data yang tersimpan lebih terorganisir sesuai kebutuhan sistem.

### 10.2 Analisis Model User

Langkah pertama yaitu kita membuat file bernama User.php di dalam folder `app/Models/` jadi `app/Models/User.php`

Model `User` ini berperan penting dalam proses autentikasi Laravel, khususnya untuk sistem multi-role yang terdiri dari admin, dokter, dan pasien. Model ini menyimpan atribut `role` untuk membedakan jenis pengguna, di mana pasien memiliki `no_rm` (nomor rekam medis) dan dokter memiliki `id_poli`, serta menggunakan proteksi password melalui `hashing` dan menyembunyikannya agar aman saat proses login.

```

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
10 class User extends Authenticatable
11 {
12     /** @use HasFactory<\Database\Factories\UserFactory> */
13     use HasFactory, Notifiable;
14
15     /**
16      * The attributes that are mass assignable.
17      *
18      * @var list<string>
19      */
20     protected $fillable = [
21         'nama',
22         'alamat',
23         'no_ktp',
24         'no_hp',
25         'no_rm', // role dokter tidak pakai ini
26         'role', // admin, pasien, dokter
27         'id_poli',
28         'email',
29         'password'
30     ];
31
32     /**
33      * The attributes that should be hidden for serialization.
34      *
35      * @var list<string>
36      */
37     protected $hidden = [
38         'password',
39         'remember_token',
40     ];
41
42     /**
43      * Get the attributes that should be cast.
44      *
45      * @return array<string, string>
46      */
47     protected function casts(): array
48     {
49         return [
50             'email_verified_at' => 'datetime',
51             'password' => 'hashed',
52         ];
53     }
54
55     public function poli(){
56         return $this->belongsTo(Poli::class, 'id_poli');
57     }
58
59     public function jadwalPeriksas(){
60         return $this->hasMany(JadwalPeriksa::class, 'id_dokter');
61     }
62
63
64
65 }
66

```

## 10.2 Auth Controller

Isi semua function `showLogin`, `login`, `showRegister`, dan `register` untuk menangani operasi Register dan Login di `AuthController`

`app/Http/Controllers/AuthController.php`

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Poli;
6  use App\Models\User;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Auth;
9  use Illuminate\Support\Facades\Hash;
10
11 class AuthController extends Controller
12 {
13     public function showLogin()
14     {
15         return view('auth.login');
16     }
17
18     public function login(Request $request)
19     {
20         $credentials = $request->only('email', 'password');
21
22         if (Auth::attempt($credentials)) {
23             $user = Auth::user();
24             // dd($user->role);
25             if ($user->role == 'admin') {
26                 return redirect()->route('admin.dashboard');
27             } elseif ($user->role == 'dokter') {
28                 return redirect()->route('dokter.dashboard');
29             } else {
30                 return redirect()->route('pasien.dashboard');
31             }
32             // dd($user);
33         }
34
35         // dd($user->role);
36         return back()->withErrors(['email' => 'Email atau Password Salah !']);
37     }
38
39     public function showRegister()
40     {
41         return view('auth.register');
42     }
43
44     public function register(Request $request)
45     {
46         $request->validate([
47             'nama' => ['required', 'string', 'max:255'],
48             'alamat' => ['required', 'string', 'max:255'],
49             'no_ktp' => ['required', 'string', 'max:30'],
50             'no_hp' => ['required', 'string', 'max:20'],
51             'email' => ['required', 'string', 'email', 'max:255', 'unique:users,email'],
52             'password' => ['required', 'confirmed'],
53         ]);
54
55         //cek apakah nomor KTP sudah terdaftar
56         if (User::where('no_ktp', $request->no_ktp)->exists()) {
57             return back()->withErrors(['no_ktp' => 'Nomor Ktp Sudah terdaftar']);
58         }
59
60         $no_rm = date('Ym') . '-' . str_pad(
61             User::where('no_rm', 'like', date('Ym') . '-%')->count() + 1,
62             3,
63             '0',
64             STR_PAD_LEFT
65         );
66
67         User::create([
68             'nama' => $request->nama,
69             'alamat' => $request->alamat,
70             'no_ktp' => $request->no_ktp,
71             'no_hp' => $request->no_hp,
72             'no_rm' => $no_rm,
73             'email' => $request->email,
74             'password' => Hash::make($request->password),
75             'role' => 'pasien',
76         ]);
77
78         return redirect()->route('login');
79     }
80
81     public function logout()
82     {
83         Auth::logout();
84         return redirect()->route('login');
85     }
86
87     public function dokter()
88     {
89         $data = Poli::with('dokters')->get();
90         return $data;
91     }
92 }
93
94
```

Link Code :

<https://drive.google.com/file/d/1w8nzdKT2agFtjrMnl-Zyo4QD4Re4CPB1/view?usp=sharing>

### 10.2.1 Public Function showLogin()

```
1 public function showLogin()
2 {
3     return view('auth.login');
4 }
```

Menampilkan halaman form login (auth.login)

### 10.2.2 Public Function login (Request \$request)

```
1 public function login(Request $request)
2 {
3     $credentials = $request->only('email', 'password');
4
5     if (Auth::attempt($credentials)) {
6         $user = Auth::user();
7         // dd($user->role);
8         if ($user->role == 'admin') {
9             return redirect()->route('admin.dashboard');
10        } elseif ($user->role == 'dokter') {
11            return redirect()->route('dokter.dashboard');
12        } else {
13            return redirect()->route('pasien.dashboard');
14        }
15        // dd($user);
16    }
17
18    // dd($user->role);
19    return back()->withErrors(['email' => 'Email atau Password Salah !']);
20 }
21
```

Memproses login dengan mencocokkan email dan password. Jika berhasil, pengguna akan diarahkan ke dashboard sesuai **role** (admin, dokter, atau pasien).

### 10.3.2 Public Function showRegister()

```
1 public function showRegister()
2 {
3     return view('auth.register');
4 }
```

Menampilkan halaman form registrasi (auth.register).

### 10.2.4 Public Function register (Register \$request)

```
1 public function register(Request $request)
2 {
3     $request->validate([
4         'nama' => ['required', 'string', 'max:255'],
5         'alamat' => ['required', 'string', 'max:255'],
6         'no_ktp' => ['required', 'string', 'max:30'],
7         'no_hp' => ['required', 'string', 'max:20'],
8         'email' => ['required', 'string', 'email', 'max:255', 'unique:users,email'],
9         'password' => ['required', 'confirmed'],
10    ]);
11
12
13    //cek apakah nomor KTP sudah terdaftar
14    if(User::where('no_ktp', $request->no_ktp)->exists()) {
15        return back()->withErrors(['no_ktp' => 'Nomor Ktp Sudah terdaftar']);
16    }
17
18    $no_rm = date('Ym') . '-' . str_pad(
19        User::where('no_rm', 'like', date('Ym') . '-')->count() + 1,
20        3,
21        '0',
22        STR_PAD_LEFT
23    );
24
25    User::create([
26        'nama' => $request->nama,
27        'alamat' => $request->alamat,
28        'no_ktp' => $request->no_ktp,
29        'no_hp' => $request->no_hp,
30        'no_rm' => $no_rm,
31        'email' => $request->email,
32        'password' => Hash::make($request->password),
33        'role' => 'pasien',
34    ]);
35
36    return redirect()->route('login');
37 }
```

Menangani proses pendaftaran akun pasien dengan melakukan validasi data seperti nama, email, password, alamat, nomor HP, dan nomor KTP. Setelah data dinyatakan valid dan unik, sistem secara otomatis membuat nomor rekam medis (**no\_rm**) dengan format **YYYYMM-XXX**, di mana **YYYYMM** adalah **tahun dan bulan saat ini** dan **XXX** adalah **nomor urut tiga digit** berdasarkan jumlah pasien yang sudah terdaftar di bulan tersebut. Nomor rekam medis ini menjamin setiap pasien memiliki identitas unik dalam sistem sebelum data disimpan ke database dan diarahkan ke halaman login.

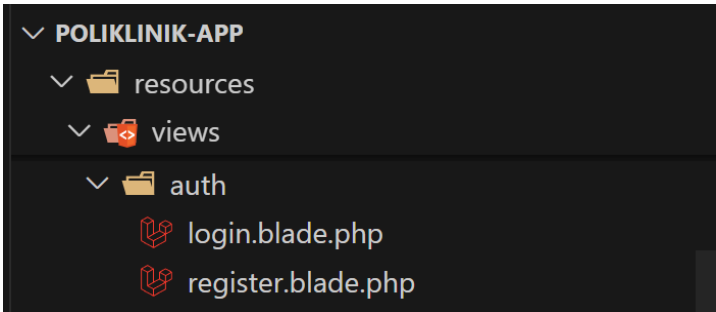
### 10.2.5 Public Function logout()

```
1 public function logout()
2 {
3     Auth::logout();
4     return redirect()->route('login');
5 }
```

Logout dari sesi pengguna dan mengarahkan kembali ke halaman login

## 10.4 Menyiapkan Tampilan Blade

Membuat folder auth di dalam views jadi **resources/views/auth**, kemudian membuat file di dalam auth yaitu **register.blade.php** dan **login.blade.php**



## 10.4.1 resources/views/auth/register.blade.php

```
1 <x-layouts.guest title="Register">
2     <div class="login-box d-flex flex-column justify-content-center align-items-center w-100 vh-100">
3         {{!-- login-logo --}}
4         <div class="card card-outline card-primary">
5             <div class="card-header text-center text-lg">
6                 <b>Poli</b>klinik
7             </div>
8
9             <div class="card-body">
10                <p class="login-box-msg">Register akun baru</p>
11                <form action="{{ route('register') }}" method="POST">
12                    @csrf
13
14                    <div class="input-group mb-3">
15                        <input type="text" class="form-control" required placeholder="Nama Lengkap" name="nama">
16                        <div class="input-group-append">
17                            <div class="input-group-text">
18                                <span class="fas fa-user"></span>
19                            </div>
20                        </div>
21                    </div>
22
23                    <div class="input-group mb-3">
24                        <input type="text" class="form-control" required placeholder="Email" name="email">
25                        <div class="input-group-append">
26                            <div class="input-group-text">
27                                <span class="fas fa-envelope"></span>
28                            </div>
29                        </div>
30                    </div>
31
32                    <div class="input-group mb-3">
33                        <input type="text" class="form-control" required placeholder="Alamat" name="alamat">
34                        <div class="input-group-append">
35                            <div class="input-group-text">
36                                <span class="fas fa-map-marker"></span>
37                            </div>
38                        </div>
39                    </div>
40
41                    <div class="input-group mb-3">
42                        <input type="number" class="form-control" required placeholder="No HP" name="no_hp">
43                        <div class="input-group-append">
44                            <div class="input-group-text">
45                                <span class="fas fa-phone-square"></span>
46                            </div>
47                        </div>
48                    </div>
49
50                    <div class="input-group mb-3">
51                        <input type="number" class="form-control" required placeholder="No KTP" name="no_ktp">
52                        <div class="input-group-append">
53                            <div class="input-group-text">
54                                <span class="fas fa-address-book"></span>
55                            </div>
56                        </div>
57                    </div>
58
59                    <div class="input-group mb-3">
60                        <input type="password" class="form-control" required placeholder="Password" name="password">
61                        <div class="input-group-append">
62                            <div class="input-group-text">
63                                <span class="fas fa-lock"></span>
64                            </div>
65                        </div>
66                    </div>
67
68                    <div class="input-group mb-3">
69                        <input type="password" class="form-control" required placeholder="Confirm Password"
70                            name="password_confirmation">
71                        <div class="input-group-append">
72                            <div class="input-group-text">
73                                <span class="fas fa-lock"></span>
74                            </div>
75                        </div>
76                    </div>
77
78                    @if ($errors->any())
79                        <div class="alert alert-danger">
80                            <ul class="mb-0">
81                                @foreach ($errors->all() as $error)
82                                    <li>{{ $error }}</li>
83                                @endforeach
84                            </ul>
85                        </div>
86                    @endif
87
88                    <div class="row">
89                        <div class="col-12 text-right">
90                            <button type="submit" class="btn btn-primary btn-block">Register</button>
91                        </div>
92                    </div>
93                </form>
94
95                <div class="row justify-content-center mt-3">
96                    <div class="col-12 text-center">
97                        <p>Sudah punya akun? <a href="{{ route('login') }}">Login</a></p>
98                    </div>
99                </div>
100            </div>
101        </div>
102    </div>
103 </x-layouts.guest>
```

untuk code lebih detailnya : [register.blade.php](#)

## 10.4.1 resources/views/auth/login.blade.php

```
1  {{-- <h2>Login</h2> --}}
2  {{--
3  @if ($errors->any())
4      <p style="color:red">{{ $errors->first() }}</p>
5  @endif --}}
6
7  {{-- <form method="POST" action="{{ route('login') }}">
8      @csrf
9      <label>Email:</label><br>
10     <input type="email" name="email" required><br>
11
12     <label>Password:</label><br>
13     <input type="password" name="password" required><br><br>
14
15     <button type="submit">Login</button>
16 </form> --}}
17
18 <x-layouts.guest title="Login">
19     <div class="login-box d-flex flex-column justify-content-center align-items-center w-100 vh-100">
20         <div class="card card-outline card-primary">
21             <div class="card-header text-center text-lg">
22                 <b>Poli</b>klinik
23             </div>
24             <div class="card-body">
25                 <p class="login-box-msg">Login ke akun anda</p>
26                 <form action="{{ route('login') }}" method="POST">
27                     @csrf
28                     <div class="input-group mb-3">
29                         <input type="text" class="form-control" placeholder="Email" name="email">
30                         <div class="input-group-append">
31                             <div class="input-group-text">
32                                 <span class="fas fa-envelope"></span>
33                             </div>
34                         </div>
35                     </div>
36                     <div class="input-group mb-3">
37                         <input type="password" class="form-control" placeholder="Password" name="password">
38                         <div class="input-group-append">
39                             <div class="input-group-text">
40                                 <span class="fas fa-lock"></span>
41                             </div>
42                         </div>
43                     </div>
44
45                     @if ($errors->any())
46                         <p class="alert alert-danger">{{ $errors->first() }}</p>
47                     @endif
48
49                     <div class="row">
50                         <div class="col-12">
51                             <button type="submit" class="btn btn-primary btn-block" name="submit">Login</button>
52                         </div>
53                     </div>
54                 </form>
55
56                 <div class="row mt-3 justify-content-center">
57                     <div class="col-12 text-center">
58                         <span>Belum punya akun? <a href="{{ route('register') }}">Register</a></span>
59                     </div>
60                 </div>
61             </div>
62         </div>
63     </div>
64 </x-layouts.guest>
65
```

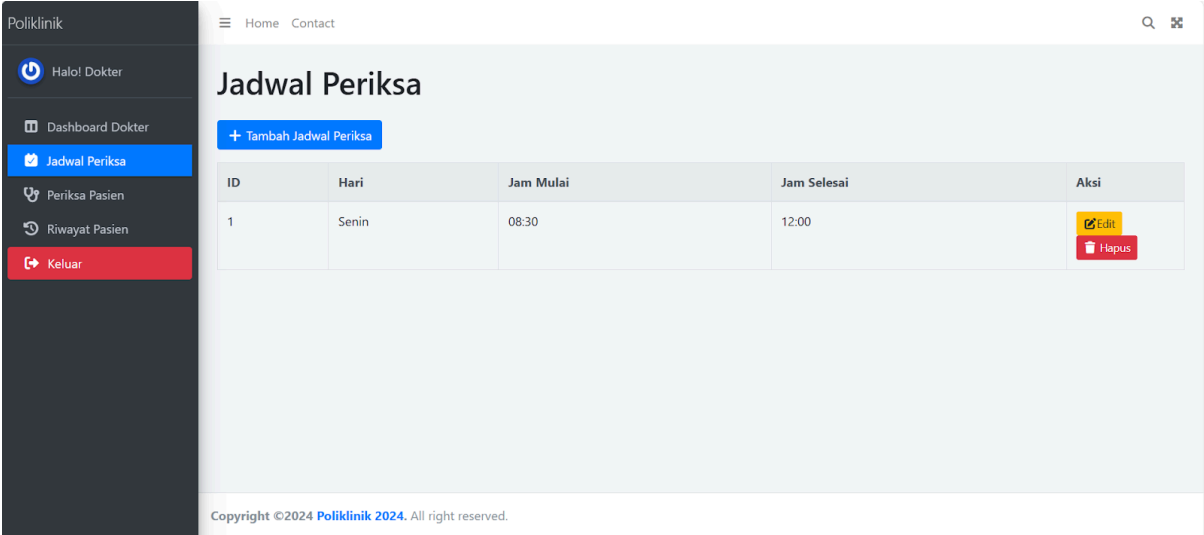
Untuk code lebih detailnya ada pada link berikut ini: [login.blade.php](#)

---

# 11 CRUD Jadwal Periksa

## 11.1 Overview

Pada modul ini kita akan membangun fitur **CRUD Jadwal Periksa**. Prosesnya dimulai dengan menampilkan daftar jadwal periksa yang sudah tersimpan di database. Setelah itu, kita membuat form input untuk menambahkan jadwal baru. Selanjutnya, kita akan mempelajari cara mengedit data jadwal jika ada perubahan, serta menghapus jadwal yang tidak diperlukan. Untuk menjaga kerapian dan konsistensi, juga ditambahkan validasi agar data yang masuk sesuai aturan. Dengan begitu, kita dapat memahami alur lengkap pengelolaan jadwal periksa melalui fitur CRUD.



- Tabel daftar jadwal periksa dengan kolom: **ID, Hari, Jam mulai, Jam selesai, dan Aksi.**
- Pada aksi Berisi tombol **Edit** untuk mengubah data jadwal yang sudah ada, dan tombol **Hapus** untuk menghapus jadwal dari sistem.

## 11.2 Implementasi CRUD dengan Controller

Untuk membuat file **PoliController.php** di dalam folder **pasien**, kita bisa menggunakan perintah Artisan dengan opsi **--resource**. Dengan opsi ini, Laravel otomatis membuat controller berisi **7 method bawaan** yang siap dipakai untuk fitur CRUD.

```
php artisan make:controller Dokter/JadwalPeriksaController --resource
```



app/Http/Controllers/Dokter/JadwalPeriksaController.php

```
1  <?php
2
3  namespace App\Http\Controllers\Dokter;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\JadwalPeriksa;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Auth;
9
10 class JadwalPeriksaController extends Controller
11 {
12     public function index()
13     {
14         // 1. Ambil user dari auth
15         $dokter = Auth::user();
16         // 2. ambil id_dokter ambil hanya hari
17         $jadwalPeriksas = JadwalPeriksa::where('id_dokter', $dokter->id)->orderBy('hari')->get();
18         // 3. return
19         return view('dokter.jadwal-periksa.index', compact('jadwalPeriksas'));
20     }
21
22     public function create()
23     {
24         return view('dokter.jadwal-periksa.create');
25     }
26
27     public function store(Request $request)
28     {
29         $request->validate([
30             'hari' => 'required',
31             'jam_mulai' => 'required',
32             'jam_selesai' => 'required'
33         ]);
34
35         JadwalPeriksa::create([
36             'id_dokter' => Auth::id(),
37             // 'id_dokter' => Auth::id(),
38             'hari' => $request->hari,
39             'jam_mulai' => $request->jam_mulai,
40             'jam_selesai' => $request->jam_selesai
41         ]);
42
43         return redirect()->route('jadwal-periksa.index')
44             ->with('message', 'Data Berhasil di Simpan')
45             ->with('type', 'success');
46     }
47
48     public function edit($id)
49     {
50         $jadwalPeriksa = JadwalPeriksa::findOrFail($id);
51         return view('dokter.jadwal-periksa.edit', compact('jadwalPeriksa'));
52     }
53
54     public function update(Request $request, string $id)
55     {
56         $request->validate([
57             'hari' => 'required',
58             'jam_mulai' => 'required',
59             'jam_selesai' => 'required'
60         ]);
61
62         $jadwalPeriksa = JadwalPeriksa::findOrFail($id);
63         $jadwalPeriksa->update([
64             'hari' => $request->hari,
65             'jam_mulai' => $request->jam_mulai,
66             'jam_selesai' => $request->jam_selesai
67         ]);
68
69         return redirect()->route('jadwal-periksa.index')
70             ->with('message', 'Berhasil Melakukan Update Data')
71             ->with('type', 'success');
72     }
73
74     public function destroy(string $id)
75     {
76         $jadwalPeriksa = JadwalPeriksa::findOrFail($id);
77         $jadwalPeriksa->delete();
78
79         return redirect()->route('jadwal-periksa.index')
80             ->with('message', 'Berhasil Melakukan Hapus Data')
81             ->with('type', 'success');
82     }
83
84 }
85
```

Link Code :

[JadwalPeriksaController.php](#)

Controller ini nantinya digunakan untuk:

- Menampilkan data pasien yang sedang login, data poli yang tersedia, dan data jadwal periksa.
- Menambahkan data baru ke database agar jadwal periksa bisa dikelola sesuai kebutuhan aplikasi.

Selain itu, ada juga **JadwalPeriksaController** yang khusus menangani proses CRUD jadwal periksa untuk dokter. Pada bagian awal, controller ini memakai **namespace** dan **use** untuk memanggil class penting seperti **Request**, **JadwalPeriksa**, dan **Controller** bawaan Laravel. Hal ini membuat controller bisa berkomunikasi dengan model dan menerima input dari form.

## 1. Function Index

method **index()** digunakan untuk mengambil semua jadwal periksa milik dokter yang sedang login dengan bantuan **Eloquent ORM**. Data tersebut kemudian dikirim ke view, sehingga dapat ditampilkan sebagai daftar jadwal periksa

```
1 public function index()
2 {
3     // 1. Ambil user dari auth
4     $dokter = Auth::user();
5     // 2. ambil id_dokter ambil hanya hari
6     $jadwalPeriksas = JadwalPeriksa::where('id_dokter', $dokter->id)->orderBy('hari')->get();
7     // 3. return
8     return view('dokter.jadwal-periksa.index', compact('jadwalPeriksas'));
9 }
```

## 2. Function Create

Selanjutnya ada **proses menambah data baru**

- Method **Create()** digunakan untuk menampilkan form input jadwal.
- Method **store()** bertugas memproses data yang dikirim dari form tersebut.

Pada tahap ini juga dilakukan **validasi** agar data yang dimasukkan sesuai aturan, seperti hari, jam mulai, dan jam selesai. Jika validasi berhasil, data akan langsung disimpan ke dalam tabel **jadwal\_periksas**.

```
1 public function create()
2 {
3     return view('dokter.jadwal-periksa.create');
4 }
5
6 public function store(Request $request)
7 {
8     $request->validate([
9         'hari' => 'required',
10        'jam_mulai' => 'required',
11        'jam_selesai' => 'required'
12    ]);
13
14    JadwalPeriksa::create([
15        'id_dokter' => Auth::id(),
16        // 'id_dokter' => Auth::id(),
17        'hari' => $request->hari,
18        'jam_mulai' => $request->jam_mulai,
19        'jam_selesai' => $request->jam_selesai
20    ]);
21
22    return redirect()->route('jadwal-periksa.index')
23        ->with('message', 'Data Berhasil di Simpan')
24        ->with('type', 'success');
25 }
```

### 3. Function Edit

Untuk proses **mengubah data**, digunakan method **edit()** dan **update()**. Method **edit()** berfungsi mengambil data jadwal berdasarkan ID lalu menampilkannya di form edit. Setelah pengguna melakukan perubahan, method **update()** akan melakukan validasi terhadap data baru dan menyimpannya kembali ke database. Dengan cara ini, konsistensi data jadwal pemeriksaan tetap terjaga.

```
1 public function edit($id)
2 {
3     $jadwalPeriksa = JadwalPeriksa::findOrFail($id);
4     return view('dokter.jadwal-periksa.edit', compact('jadwalPeriksa'));
5 }
6
7 public function update(Request $request, string $id)
8 {
9     $request->validate([
10         'hari' => 'required',
11         'jam_mulai' => 'required',
12         'jam_selesai' => 'required'
13     ]);
14
15     $jadwalPeriksa = JadwalPeriksa::findOrFail($id);
16     $jadwalPeriksa->update([
17         'hari' => $request->hari,
18         'jam_mulai' => $request->jam_mulai,
19         'jam_selesai' => $request->jam_selesai
20     ]);
21
22     return redirect()->route('jadwal-periksa.index')
23         ->with('message', 'Berhasil Melakukan Update Data')
24         ->with('type', 'success');
25 }
26
```

### 4. Function Destroy (Delete)

Terakhir, ada proses **menghapus data** menggunakan method **destroy()**. Method ini akan mencari jadwal pemeriksaan berdasarkan ID, kemudian menghapusnya dari database. Setelah berhasil, pengguna akan diarahkan kembali ke halaman daftar jadwal dengan notifikasi sebagai konfirmasi bahwa data telah dihapus.

```
1 public function destroy(string $id)
2 {
3     $jadwalPeriksa = JadwalPeriksa::findOrFail($id);
4     $jadwalPeriksa->delete();
5
6     return redirect()->route('jadwal-periksa.index')
7         ->with('message', 'Berhasil Melakukan Hapus Data')
8         ->with('type', 'success');
9 }
```

## 11.3 Menambahkan Routes

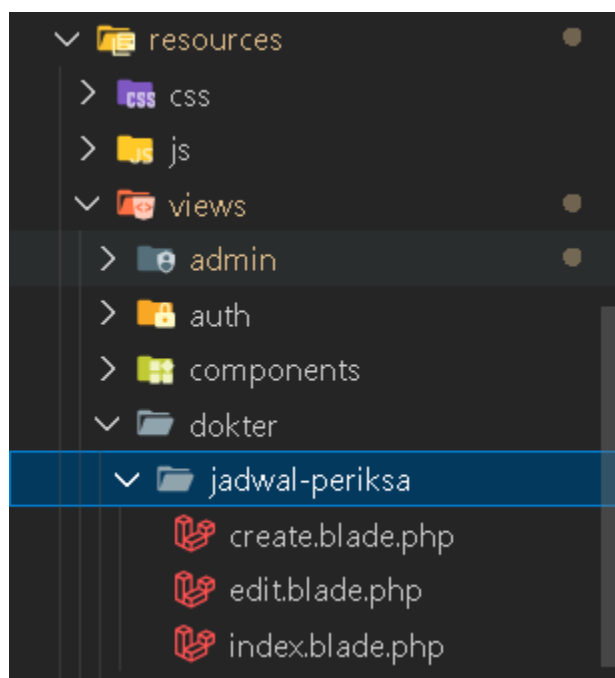
Pada `routes/web.php`, tambahkan code routes jadwal-periksa di bagian role Dokter seperti ini:

```
1 Route::middleware(['auth', 'role:dokter'])->prefix('dokter')->group(function () {
2     Route::get('/dashboard', function () {
3         return view('dokter.dashboard');
4     })->name('dokter.dashboard');
5     Route::resource('jadwal-periksa', DokterJadwalPeriksaController::class);
6
7 });
```

Baris ini digunakan untuk mendefinisikan **resource route** pada `JadwalPeriksaController`. Dengan `Route::resource`, Laravel otomatis membuat semua route yang dibutuhkan untuk operasi CRUD, seperti: menampilkan daftar jadwal (**index**), menampilkan form tambah (**create**), menyimpan data (**store**), menampilkan form edit (**edit**), memperbarui data (**update**), dan menghapus data (**destroy**).

## 11.4 Create View

Setelah membuat controller dan route, langkah selanjutnya adalah menyiapkan **file view** yang digunakan untuk menampilkan data jadwal pemeriksaan. Seluruh file view ditempatkan di dalam folder `resources/views/dokter/jadwal-periksa`. Pastikan penamaan folder maupun file sesuai dengan yang dipanggil di controller, agar aplikasi dapat berjalan tanpa error.



Pada tahap ini akan dibuat tiga file utama, yaitu **index.blade.php** untuk menampilkan daftar jadwal, **create.blade.php** untuk menambahkan jadwal baru, serta **edit.blade.php** untuk mengubah jadwal yang sudah tersedia.

resource/views/dokter/jadwal-periksa/index.blade.php

```
1 <x-layouts.app title="Jadwal Periksa">
2   <div class="container-fluid px-4 mt-4">
3     <div class="row">
4       <div class="col-lg-12">
5
6         {{-- Alert flash message --}}
7         @if (session('message'))
8           <div class="alert alert-{{ session('type', 'success') }} alert-dismissible fade show" role="alert">
9             {{ session('message') }}
10            <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
11          </div>
12        @endif
13
14        <h1 class="mb-4">Jadwal Periksa</h1>
15
16        <a href="{{ route('jadwal-periksa.create') }}" class="btn btn-primary mb-3">
17          <i class="fas fa-plus"></i> Tambah Jadwal Periksa
18        </a>
19
20        <div class="table-responsive">
21          <table class="table table-bordered">
22            <thead class="thead-light">
23              <tr>
24                <th>ID</th>
25                <th>Hari</th>
26                <th>Jam Mulai</th>
27                <th>Jam Selesai</th>
28                <th style="width: 150px;">Aksi</th>
29              </tr>
30            </thead>
31            <tbody>
32              @forelse ($jadwalPeriksas as $jadwalPeriksa)
33                <tr>
34                  <td>{{ $jadwalPeriksa->id }}</td>
35                  <td>{{ $jadwalPeriksa->hari }}</td>
36                  <td>{{ format('H:i') akan menampilkan waktu dalam format 24 jam, misalnya 14:30. --}}
37                  <td>{{ \Carbon\Carbon::parse($jadwalPeriksa->jam_mulai)->format('H:i') }}</td>
38                  <td>{{ \Carbon\Carbon::parse($jadwalPeriksa->jam_selesai)->format('H:i') }}</td>
39                  <td>
40                    <a href="{{ route('jadwal-periksa.edit', $jadwalPeriksa->id) }}"
41                      class="btn btn-sm btn-warning">
42                      <i class="fas fa-edit"></i> Edit
43                    </a>
44                    <form action="{{ route('jadwal-periksa.destroy', $jadwalPeriksa->id) }}"
45                      method="POST" style="display: inline-block;">
46                      @csrf
47                      @method('DELETE')
48                      <button class="btn btn-sm btn-danger"
49                        onclick="return confirm('Yakin ingin menghapus Data Jadwal Periksa ini ?')">
50                        <i class="fas fa-trash"></i> Hapus
51                      </button>
52                    </form>
53                  </td>
54                </tr>
55              @empty
56                <tr>
57                  <td class="text-center" colspan="7">
58                    Belum ada Jadwal Periksa
59                  </td>
60                </tr>
61              @endforelse
62            </tbody>
63          </table>
64        </div>
65      </div>
66    </div>
67  </x-layouts.app>
68
69  <script>
70    setTimeout(() => {
71      const alert = document.querySelector('.alert');
72      if (alert) {
73        alert.classList.remove('show');
74        alert.classList.add('fade');
75        setTimeout(() => alert.remove(), 500);
76      }
77    }, 2000);
78  </script>
79
80 </x-layouts.app>
```

Code dapat dilihat pada file berikut: [index.blade.php](#)

resource/views/dokter/jadwal-periksa/create.blade.php

```
1 <x-layouts.app title="Tambah Jadwal Periksa Dokter">
2 <!-- ALERT FLASH MESSAGE -->
3 <div class="container-fluid px-4 mt-4">
4     <div class="row">
5         <div class="col-lg-8 offset-lg-2">
6             <h1 class="mb-4">Tambah Jadwal Periksa</h1>
7             <div class="card">
8                 <div class="card-body">
9                     <form action="{{ route('jadwal-periksa.store') }}" method="POST">
10                        @csrf
11                        <div class="form-group mb-3">
12                            <label for="hari" class="form-label">Hari</label>
13                            <select name="hari" id="hari" class="form-select @error('hari') is-invalid @enderror"
14                                required>
15                                <option value="">Pilih Hari</option>
16                                @foreach (['Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu', 'Minggu'] as $day)
17                                    <option value="{{ $day }}" {{ old('hari') == $day ? 'selected' : '' }}>{{ $day }}
18                                </option>
19                                @endforeach
20                            </select>
21                            @error('hari')
22                                <div class="invalid-feedback">{{ $message }}</div>
23                            @enderror
24                        </div>
25
26                        <div class="form-group mb-3">
27                            <label for="jam_mulai" class="form-label">Jam Mulai</label>
28                            <input type="time" name="jam_mulai" id="jam_mulai"
29                                class="form-control @error('jam_mulai') is-invalid @enderror"
30                                value="{{ old('jam_mulai') }}" required>
31                            @error('jam_mulai')
32                                <div class="invalid-feedback">{{ $message }}</div>
33                            @enderror
34                        </div>
35
36                        <div class="form-group mb-3">
37                            <label for="jam_selesai" class="form-label">Jam Selesai</label>
38                            <input type="time" name="jam_selesai" id="jam_selesai"
39                                class="form-control @error('jam_selesai') is-invalid @enderror"
40                                value="{{ old('jam_selesai') }}" required>
41                            @error('jam_selesai')
42                                <div class="invalid-feedback">{{ $message }}</div>
43                            @enderror
44                        </div>
45
46                        <div class="form-group mt-4">
47                            <button type="submit" class="btn btn-success">
48                                <i class="fas fa-save"></i> Simpan
49                            </button>
50                            <a href="{{ route('jadwal-periksa.index') }}" class="btn btn-secondary">
51                                <i class="fas fa-arrow-left"></i> Kembali
52                            </a>
53                        </div>
54                    </form>
55                </div>
56            </div>
57        </div>
58    </div>
59 </x-layouts.app>
60
```

Code dapat dilihat pada file berikut: [create.blade.php](#)

resource/views/dokter/jadwal-periksa/edit.blade.php

```
1 <x-layouts.app title="Edit Jadwal Periksa Dokter">
2 <!-- ALERT FLASH MESSAGE -->
3 <div class="container-fluid px-4 mt-4">
4   <div class="row">
5     <div class="col-lg-8 offset-lg-2">
6       <h1 class="mb-4">Edit Jadwal Periksa</h1>
7       <div class="card">
8         <div class="card-body">
9           <form action="{{ route('jadwal-periksa.update', $jadwalPeriksa->id) }}" method="POST">
10             @csrf
11             @method('PUT')
12             <div class="form-group mb-3">
13               <label for="hari" class="form-label">Hari</label>
14               <select name="hari" id="hari" class="form-select @error('hari') is-invalid @enderror"
15                 required>
16                 <option value="">Pilih Hari</option>
17                 @foreach (['Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu', 'Minggu'] as $day)
18                   <option value="{{ $day }}" {{ (old('hari') ?? $jadwalPeriksa->hari) == $day ? 'selected' : '' }}>{{ $day }}</option>
19                 @endforeach
20               </select>
21               @error('hari')
22               <div class="invalid-feedback">{{ $message }}</div>
23             @enderror
24             </div>
25
26             <div class="form-group mb-3">
27               <label for="jam_mulai" class="form-label">Jam Mulai</label>
28               <input type="time" name="jam_mulai" id="jam_mulai"
29                 class="form-control @error('jam_mulai') is-invalid @enderror"
30                 value="{{ old('jam_mulai') ?? $jadwalPeriksa->jam_mulai }}" required>
31               @error('jam_mulai')
32               <div class="invalid-feedback">{{ $message }}</div>
33             @enderror
34             </div>
35
36             <div class="form-group mb-3">
37               <label for="jam_selesai" class="form-label">Jam Selesai</label>
38               <input type="time" name="jam_selesai" id="jam_selesai"
39                 class="form-control @error('jam_selesai') is-invalid @enderror"
40                 value="{{ old('jam_selesai') ?? $jadwalPeriksa->jam_selesai }}" required>
41               @error('jam_selesai')
42               <div class="invalid-feedback">{{ $message }}</div>
43             @enderror
44             </div>
45
46             <div class="form-group mt-4">
47               <button type="submit" class="btn btn-success">
48                 <i class="fas fa-save"></i> Update
49               </button>
50               <a href="{{ route('jadwal-periksa.index') }}" class="btn btn-secondary">
51                 <i class="fas fa-arrow-left"></i> Kembali
52               </a>
53             </div>
54           </form>
55         </div>
56       </div>
57     </div>
58   </div>
59 </x-layouts.app>
```

Code dapat dilihat pada file berikut : [edit.blade.php](#)

## 11.4 Menambahkan Sidebar

Supaya fitur mudah diakses, kita tambahkan menu baru untuk “**Jadwal Periksa**” ke dalam sidebar layout. Tempatkan sdi Role Dokter setelah “Dashboard Dokter”.

```
1 <li class="nav-item">
2   <a href="{{ route('jadwal-periksa.index') }}"
3     class="nav-link {{ request()->routeIs('jadwal-periksa.*') ? 'active' : '' }}">
4     <i class="nav-icon fas fa-calendar-check"></i>
5     <p>
6       Jadwal Periksa
7     </p>
8   </a>
9 </li>
```

## 11.4 Testing Fitur

Tahap terakhir adalah mencoba semua fungsi CRUD.

### 11.4.1 Login Sebagai Admin


Email : dokter@gmail.com


Password : dokter


### 11.4.2 Uji Semua Fitur


- Tambah Jadwal Periksa
- Edit Jadwal Periksa
- Hapus Jadwal Periksa


Poliklinik

 Halo! Dokter

 Dashboard Dokter

 Jadwal Periksa

 Keluar







 Home Contact

Q

✕

## Jadwal Periksa

+ Tambah Jadwal Periksa

ID	Hari	Jam Mulai	Jam Selesai	Aksi
1	Senin	09:00	14:00	<div> Edit</div> <div> Hapus</div>
3	Selasa	08:00	14:00	<div> Edit</div> <div> Hapus</div>
4	Rabu	09:00	15:00	<div> Edit</div> <div> Hapus</div>

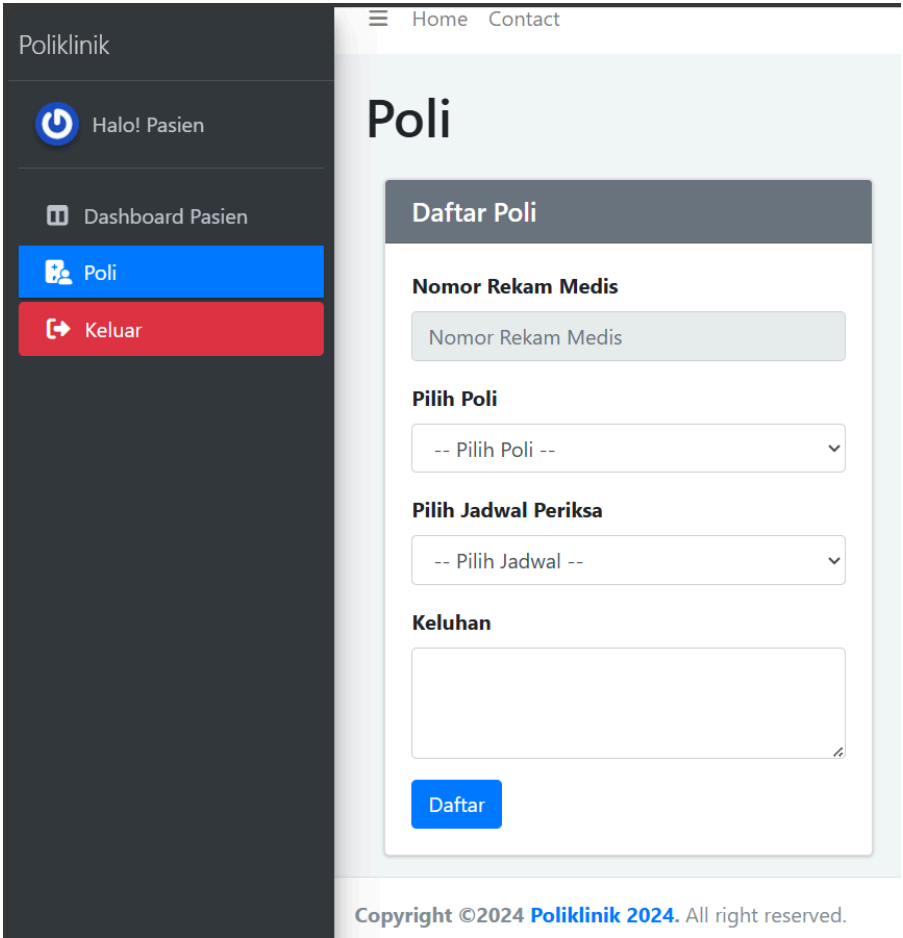
Copyright %copy;2025 Poliklinik 2025. All right reserved.



# Modul 12: Pasien Mendaftar ke Poli

## 12.0 Overview

Pada modul ini, kita akan membuat fitur **Daftar Poli**, yaitu fitur yang berfungsi agar pasien bisa mendaftar ke **Poli** untuk melakukan periksa ke dokter sesuai dengan poli dokter dan jadwal periksanya. Di halaman daftar poli ini terdapat form pendaftaran yang berisi kolom nomor rekam medis dimana nomor rekam medis didapat ketika pasien melakukan registrasi akun, kemudian akan terisi secara otomatis pada kolom. Kemudian kolom untuk memilih poli dan juga jadwal poli tersebut. Kemudian kolom keluhan untuk menjelaskan kepada dokter tentang keluhan yang dialami oleh pasien. Terakhir ada Tombol "Daftar" digunakan untuk mengirimkan data pendaftaran yang telah diisi pada form. Setelah tombol ditekan, sistem akan memproses data dan menyimpannya ke database.



## 12.1 Membuat Controller PoliController

Kita akan membuat file PoliController.php di dalam Folder Pasien di dalam Folder Controller, kita bisa menggunakan perintah `php artisan make:controller Pasien/PoliController`.

```
php artisan make:controller Pasien/PoliController --resource
```

Di dalam file ini kita akan membuat fungsi untuk menampilkan data milik pasien yang sedang login, data poli yang ada, dan data jadwal pemeriksaan. Kemudian kita akan membuat fungsi untuk memasukkan data ke dalam database.



`app/Http/Controllers/Pasien/PoliController.php`

```
1 <?php
2
3 namespace App\Http\Controllers\Pasien;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\DaftarPoli;
7 use App\Models\JadwalPeriksa;
8 use App\Models\Poli;
9 use Illuminate\Http\Request;
10 use Illuminate\Support\Facades\Auth;
11
12 class PoliController extends Controller
13 {
14     public function get()
15     {
16         $user = Auth::user();
17         $polis = Poli::all();
18         $jadwal = JadwalPeriksa::with('dokter', 'dokter.poli')->get();
19
20         return view('pasien.daftar', [
21             'user' => $user,
22             'polis' => $polis,
23             'jadwals' => $jadwal,
24         ]);
25     }
26
27     public function submit(Request $request)
28     {
29         $request->validate([
30             // 'id_poli' => 'required|exists:poli,id',
31             'id_jadwal' => 'required|exists:jadwal_periksa,id',
32             'keluhan' => 'nullable|string',
33             'id_pasien' => 'required|exists:users,id',
34         ]);
35
36         $jumlahSudahDaftar = DaftarPoli::where('id_jadwal', $request->id_jadwal)->count();
37         $daftar = DaftarPoli::create([
38             'id_pasien' => $request->id_pasiens, //Auth::id(),
39             'id_jadwal' => $request->id_jadwals,
40             'keluhan' => $request->keluhan,
41             'no_antrian' => $jumlahSudahDaftar + 1,
42         ]);
43
44         // dd($daftar);
45
46         return redirect()->back()->with('message', 'Berhasil Mendaftar ke Poli')->with('type', 'success');
47     }
48 }
49
```

Link Code : [PoliController.php](#)

### 12.1.1 Penjelasan function get

Pada fungsi ini, kita mengambil data user yang sedang login dan menyimpannya di variabel `$user`, kemudian mengambil semua daftar poli dan menyimpannya di variabel `$polis`. Selanjutnya, kita mengambil semua jadwal pemeriksaan menggunakan `JadwalPeriksa::with('dokter', 'dokter.poli')->get()`, yang berarti kita memanfaatkan **eager loading** untuk sekaligus mengambil data relasi dokter dan poli dalam satu kali pengambilan data. Eager loading digunakan agar Laravel tidak perlu melakukan query tambahan setiap kali kita mengakses relasi (seperti `$jadwal->dokter` atau `$jadwal->dokter->poli`), sehingga aplikasi menjadi lebih cepat dan efisien. Setelah semua data terkumpul, kita kirimkan ke view `pasien.daftar` agar bisa ditampilkan di halaman pendaftaran.

### 12.1.2 Penjelasan function submit

Pada fungsi `submit`, pertama-tama dilakukan **validasi data** dari form pendaftaran, seperti memastikan `id_jadwal` dan `id_pasien` ada di database serta `keluhan` berupa teks. Setelah validasi berhasil, kita menghitung jumlah pendaftar pada jadwal tersebut menggunakan `DaftarPoli::where('id_jadwal',$request->id_jadwal)->count ()` untuk menentukan **nomor antrian** berikutnya. Kemudian dibuat data pendaftaran baru (`DaftarPoli::create`) dengan menyimpan ID pasien, ID jadwal, keluhan, dan nomor antrian yang sudah dihitung. Terakhir, pengguna diarahkan kembali ke halaman sebelumnya dengan pesan sukses agar mereka tahu bahwa pendaftaran berhasil dilakukan.

---

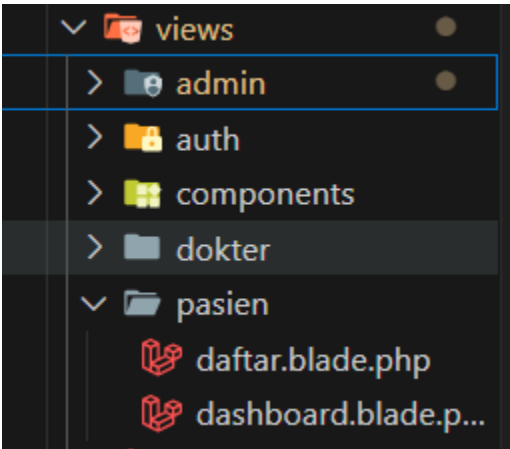
## 12.2 Menambahkan Route

```
1 use App\Http\Controllers\Pasien\PoliController as PasienPoliController;
2
3 Route::middleware(['auth','role:pasien'])->prefix('/pasien')->group( function() {
4     Route::get('/dashboard', function(){
5         return view('pasien.dashboard');
6     })->name('pasien.dashboard');
7     Route::get('/daftar', [PasienPoliController::class,'get'])->name('pasien.daftar');
8     Route::post('/daftar', [PasienPoliController::class,'submit'])->name('pasien.daftar.submit');
9 });
```

Pada file `routes/web.php` Import controller `PoliController` didalam Folder `Pasien`. Karena sebelumnya kita sudah pernah import controller dengan nama `PoliController`, kita bisa mengaliaskan dengan nama `PasienPoliController` agar nantinya tidak bertabrakan. Kemudian kita tambahkan 2 route baru yaitu Route daftar dengan method `get` yang mengarah ke function `get` di dalam `PasienPoliController` dan Route daftar dengan method `post` yang mengarah ke function `submit` di dalam `PasienPoliController`.

---

12.3 Membuat View



Tambahkan file baru daftar.blade.php di dalam folder pasien di dalam views

```
1 <!-- layouts.app title="Poli" -->
2 <div class="container-fluid px-4 mt-4">
3   <div class="row">
4     <div class="col-lg-12">
5       <!-- Alert flash message -->
6       @if (session('message'))
7         <div class="alert alert-{{ session('type', 'success') }} alert-dismissible fade show" role="alert">
8           {{ session('message') }}
9           <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
10        </div>
11      @endif
12      <h1 class="mb-4">Poli</h1>
13
14      <section class="content">
15        <div class="container-fluid">
16          <div class="row">
17            <div class="col-4">
18              <div class="card">
19                <div class="card-header bg-gray">Daftar Poli</div>
20                <div class="card-body">
21                  @if ($errors->any())
22                    <div class="alert alert-danger">
23                      <strong>Terjadi Kesalahan</strong>
24                      <ul>
25                        @foreach ($errors->all() as $error)
26                          <li>{{ $error }}</li>
27                        @endforeach
28                      </ul>
29                    </div>
30                  @endif
31                  <form action="{{ route('pasien.daftar.submit') }}" method="POST">
32                    @csrf
33                    <input type="hidden" name="id_pasien" value="{{ $user->id }}">
34
35                    <div class="mb-3">
36                      <label for="no_rm" class="form-label">Nomor Rekam Medis</label>
37                      <input type="text" class="form-control" id="no_rm" name="no_rm"
38                        value="{{ $user->no_rm }}" placeholder="Nomor Rekam Medis"
39                        disabled>
40                    </div>
41
42                    <div class="mb-3">
43                      <label for="selectPoli" class="form-label">Pilih Poli</label>
44                      <select name="id_poli" id="selectPoli" class="form-control">
45                        <option value="">Pilih Poli --</option>
46                        @foreach ($polis as $poli)
47                          <option value="{{ $poli->id }}">{{ $poli->nama_poli }}
48                        </option>
49                        @endforeach
50                      </select>
51                    </div>
52
53                    <div class="mb-3">
54                      <label for="selectJadwal" class="form-label">Pilih Jadwal</label>
55                      <select name="id_jadwal" id="selectJadwal" class="form-control">
56                        <option value="">Pilih Jadwal --</option>
57                        @foreach ($jadwals as $jadwal)
58                          <option value="{{ $jadwal->id }}">
59                            <div id="poli">{{ $jadwal->okter->poli->id }}</div>
60                            <div id="jadwal">{{ $jadwal->nama_poli }}</div>
61                            <div id="jadwal">{{ $jadwal->hari }}</div>
62                            <div id="jadwal">{{ $jadwal->jam_mulai }}</div>
63                            <div id="jadwal">{{ $jadwal->jam_selesai }}</div>
64                            <div id="jadwal">{{ $jadwal->okter->nama }}</div>
65                          </option>
66                        @endforeach
67                      </select>
68                    </div>
69
70                    <div class="mb-3">
71                      <label for="keluhan" class="form-label">Keluhan</label>
72                      <textarea name="keluhan" id="keluhan" rows="3" class="form-control"></textarea>
73                    </div>
74
75                    <button type="submit" name="submit"
76                      class="btn btn-primary">Daftar</button>
77                  </form>
78                </div>
79              </div>
80            </div>
81          </div>
82        </div>
83      </section>
84    </div>
85  </div>
86</div>
87</div>
88</div>
89</div>
90</div>
91</div>
92</div>
93</div>
94</div>
95</div>
96</div>
97</div>
98</div>
99</div>
100</div>
101</div>
102</div>
103</div>
104</div>
105</div>
106</div>
107</div>
108</div>
109</div>
110</div>
111</div>
112</div>
113</div>
114</div>
115</div>
116</div>
117</div>
```

Code dapat dilihat pada link berikut : [daftar.blade.php](#)

Pada view tersebut, kita buat form yang memiliki action yang memanggil route pasien.daftar.submit ketika user submit form. Dalam form tersebut kita ingin menampilkan nomor rekam medis dari pasien yang sedang login, kemudian ada pilihan poli yang datanya kita ambil dari variabel \$polis, setelah itu kita tampilkan pilihan jadwal yang datanya diambil dari variabel \$jadwals, yang terakhir kita berikan inputan keluhan agar pasien bisa menyampaikan keluhan yang dimilikinya. Kemudian di paling bawah kita tambahkan script javascript untuk membuat ketika pasien memilih poli, maka jadwal yang tersedia akan menyesuaikan

berdasarkan poli yang dipilih. Dan sebaliknya, jika pasien belum memilih poli tetapi sudah memilih jadwal terlebih dahulu, maka pada inputan poli akan terisi secara otomatis dengan poli yang ada pada jadwal periksa.

## 12.4 Menyesuaikan Sidebar



Pada file sidebar.blade.php di dalam folder partials di dalam folder components. Sesuaikan dengan code yang ada sebelumnya, di dalam sidebar menu, di bagian role pasien kita tambahkan shortcut berupa link yang mengarah ke route pasien.daftar, route ini akan mengarahkan kita ke view daftar yang barusan kita buat. Sehingga untuk berpindah ke halaman daftar poli kita cukup menekan link yang ada di sidebar tersebut.