

Pertemuan 3

Modul 6: CRUD Poli

Tujuan Pembelajaran Pertemuan 3 :

- Mengimplementasikan routing resource di Laravel
- Membangun tampilan form tambah, edit, dan daftar data Poli
- Melakukan validasi input data
- Memberikan feedback kepada pengguna melalui notifikasi sukses/gagal
- Mengorganisasi tampilan admin dengan struktur layout sidebar
- Membuat form input dokter
- Menghubungkan relasi dokter ke poli
- Menampilkan tabel daftar dokter
- Mengimplementasikan fitur edit dan hapus data dokter

6.0 Overview :

Pada modul ini membahas implementasi **CRUD (Create, Read, Update, Delete)** pada data Poli menggunakan framework Laravel. Dalam modul ini peserta diajarkan bagaimana menggunakan `Route::resource()` untuk menghasilkan route CRUD secara otomatis, menghubungkannya dengan `PoliController`, serta membatasi akses hanya untuk pengguna dengan role admin menggunakan middleware. Peserta juga mempelajari pembuatan tampilan halaman index, create, dan edit dengan memanfaatkan layout `x-layouts.app` agar tampilan admin lebih konsisten dan rapi. Pada sisi backend, dilakukan validasi input melalui fungsi bawaan Laravel (`$request->validate()`) di method `store()` dan `update()`, sehingga jika `nama_poli` kosong akan muncul pesan error dan user diarahkan kembali ke form dengan data lama yang tetap tersimpan. Jika validasi berhasil, data akan disimpan atau diperbarui ke database. Selain itu, modul ini juga mengajarkan penggunaan notifikasi berbasis session flash untuk memberikan umpan balik berupa pesan sukses atau gagal yang ditampilkan di halaman index, serta dilengkapi efek otomatis hilang dengan JavaScript. Tampilan admin diorganisasi menggunakan sidebar navigasi yang berisi menu “Dashboard Admin” dan “Manajemen Poli”, sehingga pengguna dapat dengan mudah mengelola data poli. Secara keseluruhan, modul ini melatih peserta untuk membangun fitur CRUD yang lengkap, mulai dari routing, controller, validasi, hingga tampilan yang interaktif dan user-friendly.

Nama Poli	Keterangan	Aksi
Poli Anak	Poli anak yang akan menyelesaikan masalah anak	<input checked="" type="button"/> Edit <input type="button"/> Hapus
Poli Gigi	Poli anak yang akan menyelesaikan masalah anak	<input checked="" type="button"/> Edit <input type="button"/> Hapus

Nama Poli	Keterangan	Aksi
Poli Anak	Poli anak yang akan menyelesaikan masalah anak	<input checked="" type="button"/> Edit <input type="button"/> Hapus
Poli Gigi	Poli anak yang akan menyelesaikan masalah anak	<input checked="" type="button"/> Edit <input type="button"/> Hapus
Poli Umum	Menangani keluhan kesehatan dasar.	<input checked="" type="button"/> Edit <input type="button"/> Hapus

6.1 Route

6.1.1 Pengertian route

Route adalah mekanisme di Laravel untuk menentukan URL (alamat web) dan aksi (apa yang dilakukan ketika URL tersebut diakses). Ketika user mengetikkan URL tertentu, Laravel akan mengarahkannya ke fungsi / controller yang sesuai.

6.1.2 Route Resource

Di Laravel, **Route Resource** adalah fitur bawaan yang digunakan untuk membuat **routing CRUD (Create, Read, Update, Delete)** secara otomatis hanya dengan satu baris kode.

Tanpa `Route::resource()`, kita harus menulis 7 route manual seperti ini:

```
1 Route::get('/polis', [PoliController::class, 'index']);
2 Route::get('/polis/create', [PoliController::class, 'create']);
3 Route::post('/polis', [PoliController::class, 'store']);
4 Route::get('/polis/{id}', [PoliController::class, 'show']);
5 Route::get('/polis/{id}/edit', [PoliController::class, 'edit']);
6 Route::put('/polis/{id}', [PoliController::class, 'update']);
7 Route::delete('/polis/{id}', [PoliController::class, 'destroy']);
8
```

Namun dengan `Route::resource()`, kita cukup menulis **satu baris saja**:

```
1     Route::resource('polis', PoliController::class);
2
```

Kode di atas akan menghasilkan 7 route sekaligus, masing-masing merujuk ke method dalam `PoliController`, 7 route tersebut yaitu:

HTTP Method	URI	Action	Controller Method	Tujuan
GET	/polis	index	index()	Menampilkan halaman daftar data Poli. Biasanya berisi tabel semua data poli yang ada di database.
GET	/polis/create	create	create()	Menampilkan form tambah data. User akan mengisi form ini untuk menambah poli baru.
POST	/polis	store	store()	Menerima data dari form create, melakukan validasi,

				lalu menyimpan ke database
GET	/polis/{id}	show	show()	Menampilkan detail poli tertentu berdasarkan ID. (Jarang dipakai di halaman admin, lebih sering untuk halaman publik.)
GET	/polis/{id}/edit	edit	edit()	Menampilkan form edit untuk data poli tertentu.
PUT/PATCH	/polis/{id}	update	update()	Menerima data dari form edit, melakukan validasi, lalu mengupdate data di database.
DELETE	/polis/{id}	destroy	destroy()	Menghapus data poli dari database berdasarkan ID.

6.2 Controller

Controller adalah bagian dari arsitektur **MVC (Model–View–Controller)** yang berfungsi sebagai penghubung antara model dan view. Controller mengatur logika aplikasi, yaitu menerima request dari pengguna, memproses data melalui model, lalu mengirimkan hasilnya ke view untuk ditampilkan. Dengan adanya controller, alur kerja aplikasi menjadi lebih terstruktur karena pemrosesan data dan tampilan dipisahkan dengan jelas.

6.3 Middleware untuk Role Admin

Middleware `role:admin` digunakan untuk membatasi akses halaman CRUD Poli hanya untuk user yang memiliki role admin.

Jika ada user yang bukan admin mencoba mengakses halaman manajemen poli, middleware akan memblokir akses dan menampilkan pesan error 403 Forbidden atau mengarahkan user ke halaman lain (misalnya dashboard).

Dengan middleware ini, keamanan sistem lebih terjaga karena hanya admin yang bisa menambah, mengedit, dan menghapus data poli.

6.4 Struktur folder

```
poliklinik-app-bengkod/
├── app/
│   └── Http/
│       └── Controllers/
│           └── PoliController.php      # Controller utama CRUD Poli
│
└── app/Models/
    └── Poli.php                  # Model Eloquent Poli
|
└── database/migrations/
    └── 0001_01_01_000000_create_poli_table.php  # Migrasi tabel poli
|
└── database/seeders/
    └── PoliSeeder.php            # Seeder data awal poli
|
└── resources/views/admin/polis/
    ├── index.blade.php          # Halaman daftar poli
    ├── create.blade.php         # Form tambah poli
    └── edit.blade.php           # Form edit poli
|
└── routes/
    └── web.php                 # Route::resource('polis', PoliController::class)
```

6.5 Implementasi

6.5.1. Routing

Tambahkan resource route di `routes/web.php`:

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

6.5.2 Membuat PoliController

a. Generate Resource Controller

Gunakan perintah berikut:

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

b. Isi Method pada Controller

Isi lengkap method seperti index, create, store, edit, update, dan destroy untuk menangani operasi CRUD.

isi controller PoliController:

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Polis;
6 use Illuminate\Auth\Events\Validated;
7 use Illuminate\Http\Request;
8 use PDO;
9
10 class PoliController extends Controller
11 {
12     public function index()
13     {
14         $polis = Poli::all();
15         return view('admin.polis.index', compact('polis'));
16     }
17
18     public function create()
19     {
20         return view('admin.polis.create');
21     }
22
23     public function store(Request $request)
24     {
25         $validated = $request->validate([
26             'nama_poli' => 'required',
27             'keterangan' => 'nullable',
28         ]);
29
30         Poli::create($validated);
31         return redirect()->route('polis.index')->with('success', 'Poli berhasil di tambahkan')->with('type', 'success');
32     }
33
34     public function edit($id)
35     {
36         $poli = Poli::findOrFail($id);
37         return view('admin.polis.edit', compact('poli'));
38     }
39
40     public function update(Request $request, $id)
41     {
42         $validated = $request->validate([
43             'nama_poli' => 'required',
44             'keterangan' => 'nullable',
45         ]);
46
47         $poli = Poli::findOrFail($id);
48         $poli->update($validated);
49         return redirect()->route('polis.index')->with('success', 'Polis berhasil di update');
50     }
51
52     public function destroy($id)
53     {
54         $poli = Poli::findOrFail($id);
55         $poli->delete($poli);
56         return redirect()->route('polis.index')->with('success', 'Polis Berhasil di hapus !');
57     }
}
```

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

c. Penjelasan Function

• Function index

```
1 public function index()
2 {
3     $polis = Poli::all();
4     return view('admin.polis.index', compact('polis'));
5 }
```

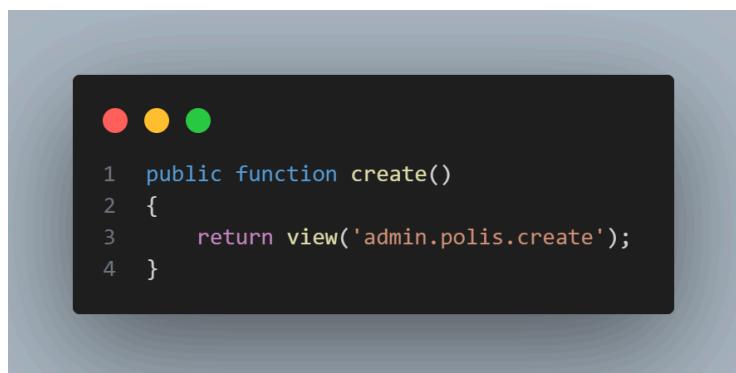
Fungsi ini bertugas untuk menampilkan daftar semua poli.

Query: `$polis = Poli::all();`

- Kode ini mengambil semua data dari tabel polis.

View: `return view('admin.polis.index', compact('polis'));`

- Data poli dikirim ke tampilan `admin.polis.index` menggunakan `compact('polis')`.
- **Function Create**

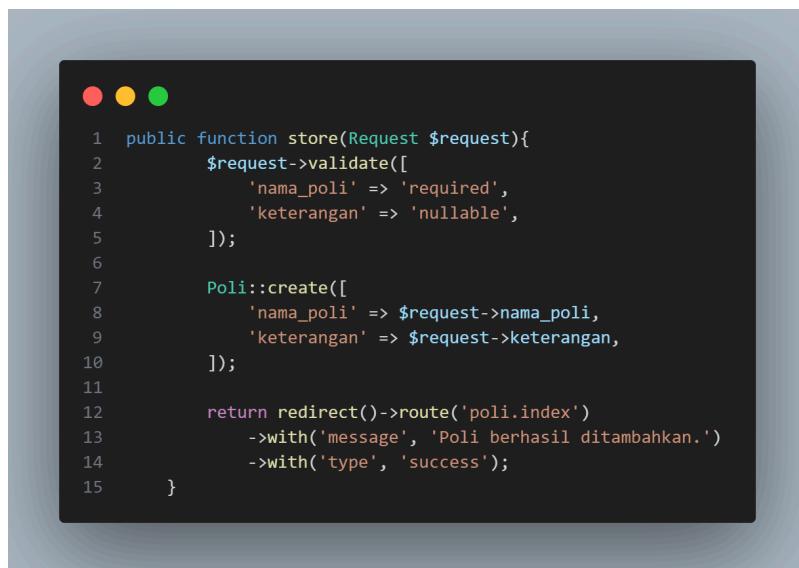


```
1 public function create()
2 {
3     return view('admin.polis.create');
4 }
```

Fungsi ini bertugas hanya untuk menampilkan form penambahan poli baru.

View: `return view('admin.polis.create');`

- Mengarahkan pengguna ke halaman form `admin.polis.create`.
- **Function Store**



```
1 public function store(Request $request){
2     $request->validate([
3         'nama_poli' => 'required',
4         'keterangan' => 'nullable',
5     ]);
6
7     Poli::create([
8         'nama_poli' => $request->nama_poli,
9         'keterangan' => $request->keterangan,
10    ]);
11
12     return redirect()->route('poli.index')
13         ->with('message', 'Poli berhasil ditambahkan.')
14         ->with('type', 'success');
15 }
```

Fungsi ini digunakan untuk menyimpan data poli baru ke database.

Validasi input:

- `namapoli` harus diisi (`required`).
- `keterangan` boleh kosong (`nullable`).

Query: `Poli::create([]);`

- Kode ini menyimpan data poli baru ke tabel `polis`.

Redirect: `return redirect()->route('polis.index')->with('success', 'Poli berhasil di tambahkan');`

- Setelah berhasil menyimpan, diarahkan kembali ke daftar poli dengan pesan sukses.
- **Function Edit**



```

● ● ●

1 public function edit($id)
2 {
3     $poli = Poli::findOrFail($id);
4     return view('admin.polis.edit', compact('poli'));
5 }

```

Fungsi ini bertugas untuk mengambil data poli tertentu yang akan diedit.

Query: `$poli = Poli::findOrFail($id);`

- Kode ini mencari data poli berdasarkan ID. Jika data tidak ditemukan, otomatis error 404.

View: `return view('admin.polis.edit', compact('poli'));`

- Data poli dikirim ke tampilan `admin.polis.edit` untuk ditampilkan di form edit.

- **Function Update**



```

● ● ●

1 public function update(Request $request, $id)
2 {
3     $validated = $request->validate([
4         'nama_poli' => 'required',
5         'keterangan' => 'nullable',
6     ]);
7
8     $poli = Poli::findOrFail($id);
9     $poli->update($validated);
10    return redirect()->route('polis.index')->with('success', 'Polis berhasil di update');
11 }

```

Fungsi ini digunakan untuk memperbarui data poli yang sudah ada.

Validasi input:

- `nama_poli` harus diisi (`required`).
- `keterangan` boleh kosong (`nullable`).

Query: `$poli = Poli::findOrFail($id);`

`$poli->update($validated);`

- Kode ini mencari poli berdasarkan ID, lalu memperbarui data sesuai input.

Redirect: `return`

```
redirect()->route('polis.index')->with('success', 'Poli  
berhasil di update');
```

- Setelah update, diarahkan ke daftar poli dengan pesan sukses.

- **Function Destroy**



Fungsi ini digunakan untuk menghapus data poli dari database.

Query: `$poli = Poli::findOrFail($id);`

```
$poli->delete();
```

- Kode ini mencari poli berdasarkan ID, lalu menghapusnya.

Redirect: `return`

```
redirect()->route('polis.index')->with('success', 'Polis  
Berhasil di hapus !');
```

- Setelah penghapusan, diarahkan kembali ke daftar poli dengan notifikasi sukses.

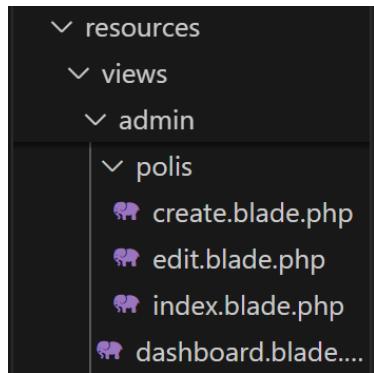
6.5.3 Form Tambah/Edit Poli

Form ini dipakai untuk **menambah** atau **mengedit** data poli.

- **Field `nama_poli`** → input teks, wajib diisi.
- **Field `keterangan`** → textarea, boleh kosong.
- Saat **tambah**, form diarahkan ke **route `polis.store`**.
- Saat **edit**, form diarahkan ke **route `polis.update`** dengan method **PUT/PATCH**, dan field otomatis terisi data lama.
- Semua form dilindungi oleh **CSRF token (@csrf)** agar aman.

a. Menyiapkan Tampilan Blade

Struktur folder:



b. polis/index.blade.php

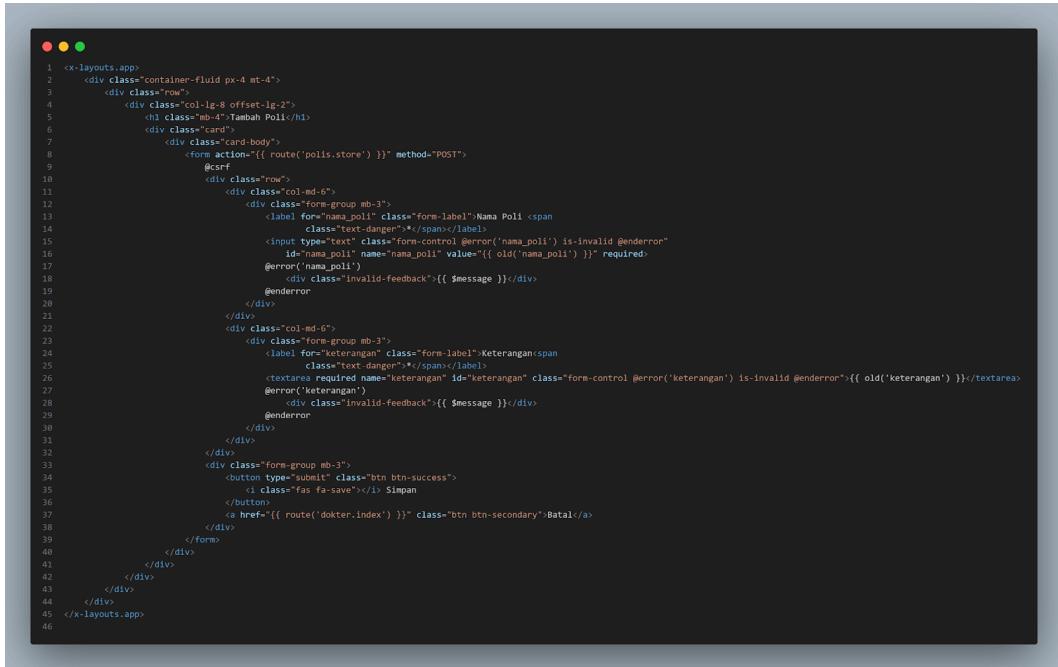
```
<x-layouts.app title="Data Polisi">
@if (session('success'))
    <div class="alert alert-success" id="alert">{{ session('success') }}</div>
@endif
<div class="container-fluid px-4 mt-4">
    <div class="row">
        <div class="col-lg-12">
            {{-- Alert flash message --}}
            @if (session('message'))
                <div class="alert alert-{{ session('type', 'success') }} alert-dismissible fade show" role="alert">
                    {{ session('message') }}
                    <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
                </div>
            @endif
            <h1 class="mb-4">Data Polisi</h1>
            <a href="{{ route('polis.create') }}" class="btn btn-primary mb-3">
                <i class="fas fa-plus"></i> Tambah Poli
            </a>
            <div class="table-responsive">
                <table class="table table-bordered">
                    <thead class="thead-light">
                        <tr>
                            <th>Nama Poli</th>
                            <th>Keterangan</th>
                            <th style="width: 150px;">Aksi</th>
                        </tr>
                    </thead>
                    <tbody>
                        @forelse ($polis as $poli)
                            <tr>
                                <td>{{ $poli->nama_poli }}</td>
                                <td>{{ $poli->keterangan }}</td>
                                <td>
                                    <a href="{{ route('polis.edit', $poli->id) }}" class="btn btn-sm btn-warning">
                                        <i class="fas fa-edit"></i> Edit
                                    </a>
                                    <form action="{{ route('polis.destroy', $poli->id) }}" method="POST" style="display: inline-block;">
                                        @csrf
                                        @method('DELETE')
                                        <button class="btn btn-sm btn-danger" onclick="return confirm('Yakin ingin menghapus Poli ini ?')">
                                            <i class="fas fa-trash"></i> Hapus
                                        </button>
                                    </form>
                                </td>
                            </tr>
                        @empty
                            <tr>
                                <td class="text-center" colspan="3">Belum ada Poli</td>
                            </tr>
                        @endforelse
                    </tbody>
                </table>
            </div>
        </div>
        <script>
            setTimeout(() => {
                const alert = document.getElementById('alert');
                if (alert) alert.remove();
            }, 3000);
        </script>
    </x-layouts.app>

```

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

c. polis/create.blade.php

Form untuk menambahkan data poli baru. Field: nama poli, keterangan.

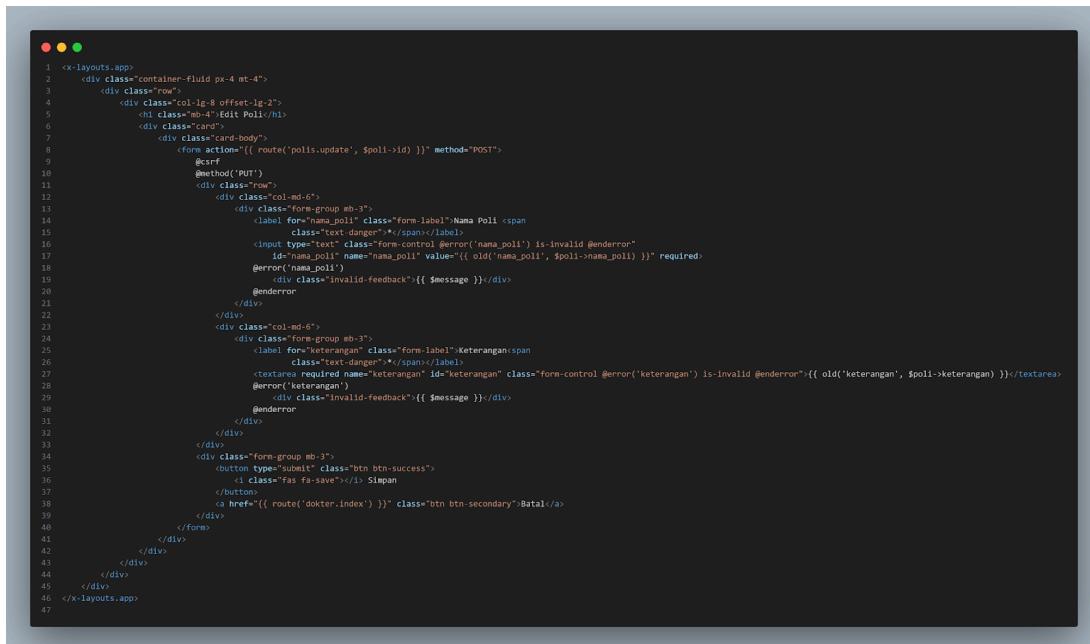


```
1 <x-layouts.app>
2   <div class="container-fluid px-4 mt-4">
3     <div class="row">
4       <div class="col-lg-8 offset-lg-2">
5         <div class="mb-4">Tambah Poli</div>
6         <div class="card">
7           <div class="card-body">
8             <form action="{{ route('polis.store') }}" method="POST">
9               @csrf
10              <div class="row">
11                <div class="col-md-6">
12                  <div class="form-group mb-3">
13                    <label for="nama_poli" class="form-label">Nama Poli <span
14                      class="text-danger"></span></label>
15                    <input type="text" class="form-control @error('nama_poli') is-invalid @enderror"
16                      id="nama_poli" name="nama_poli" value="{{ old('nama_poli') }}" required>
17                    @error('nama_poli')
18                      <div class="invalid-feedback">{{ $message }}</div>
19                    @enderror
20                  </div>
21                </div>
22                <div class="col-md-6">
23                  <div class="form-group mb-3">
24                    <label for="keterangan" class="form-label">Keterangan:<span
25                      class="text-danger"></span></label>
26                    <textarea required name="keterangan" id="keterangan" class="form-control @error('keterangan') is-invalid @enderror">{{ old('keterangan') }}</textarea>
27                    @error('keterangan')
28                      <div class="invalid-feedback">{{ $message }}</div>
29                    @enderror
30                  </div>
31                </div>
32              </div>
33              <div class="form-group mb-3">
34                <button type="submit" class="btn btn-success">
35                  <i class="fas fa-save"></i> Simpan
36                </button>
37                <a href="{{ route('dokter.index') }}" class="btn btn-secondary">Batal</a>
38              </div>
39            </form>
40          </div>
41        </div>
42      </div>
43    </div>
44  </div>
45 </x-layouts.app>
```

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

d. polis/edit.blade.php

Form untuk mengedit data poli. Field sama seperti create, tetapi keterangan bersifat opsional.



```
1 <x-layouts.app>
2   <div class="container-fluid px-4 mt-4">
3     <div class="row">
4       <div class="col-lg-8 offset-lg-2">
5         <div class="mb-4">Edit Poli</div>
6         <div class="card">
7           <div class="card-body">
8             <form action="{{ route('polis.update', $poli->id) }}" method="POST">
9               @csrf
10              @method('PUT')
11              <div class="row">
12                <div class="col-md-6">
13                  <div class="form-group mb-3">
14                    <label for="nama_poli" class="form-label">Nama Poli <span
15                      class="text-danger"></span></label>
16                    <input type="text" class="form-control @error('nama_poli') is-invalid @enderror"
17                      id="nama_poli" name="nama_poli" value="{{ old('nama_poli', $poli->nama_poli) }}" required>
18                    @error('nama_poli')
19                      <div class="invalid-feedback">{{ $message }}</div>
20                    @enderror
21                  </div>
22                </div>
23                <div class="col-md-6">
24                  <div class="form-group mb-3">
25                    <label for="keterangan" class="form-label">Keterangan:<span
26                      class="text-danger"></span></label>
27                    <textarea required name="keterangan" id="keterangan" class="form-control @error('keterangan') is-invalid @enderror">{{ old('keterangan', $poli->keterangan) }}</textarea>
28                    @error('keterangan')
29                      <div class="invalid-feedback">{{ $message }}</div>
30                    @enderror
31                  </div>
32                </div>
33              </div>
34              <div class="form-group mb-3">
35                <button type="submit" class="btn btn-success">
36                  <i class="fas fa-save"></i> Simpan
37                </button>
38                <a href="{{ route('dokter.index') }}" class="btn btn-secondary">Batal</a>
39              </div>
40            </form>
41          </div>
42        </div>
43      </div>
44    </div>
45  </div>
46 </x-layouts.app>
```

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

6.5.4 Validasi

Validasi digunakan untuk memastikan data yang dikirim dari form sudah sesuai aturan sebelum disimpan ke database.

Pada Laravel, validasi bisa dilakukan langsung di controller menggunakan `$request->validate()`.

Contoh pada `PoliController.php`:

```
1 $validated = $request->validate([
2     'nama_poli' => 'required',
3     'keterangan' => 'nullable',
4 ]);
```

Penjelasan:

- `nama_poli` → wajib diisi (`required`).
- `keterangan` → opsional atau boleh kosong (`nullable`).

👉 Mekanisme validasi di Laravel

- Validasi dilakukan di controller (pada method `store()` dan `update()`) menggunakan fungsi bawaan Laravel `validate()`.
- Jika `nama_poli` kosong, akan muncul pesan error dan user otomatis diarahkan kembali ke form.
- Jika validasi gagal, Laravel otomatis:
 - Mengembalikan user ke halaman form.
 - Mengirimkan pesan error ke view.
 - Mengisi ulang input sebelumnya dengan `old()`.
- Jika validasi berhasil, data akan diteruskan untuk disimpan atau diperbarui ke database.

6.5.5 Notifikasi

Setelah operasi berhasil (tambah, update, hapus), kita perlu memberi tahu pengguna. Laravel menyediakan **session flash** untuk menyimpan pesan sementara yang hanya tampil 1 kali pada halaman berikutnya.

Di controller, pesan dikirim menggunakan `with()` (secara internal sama seperti `session()->flash()`):

```
● ● ●  
1 return redirect()->route('polis.index')  
2     ->with('success', 'Poli berhasil di tambahkan')  
3     ->with('type', 'success');
```

Lalu di `resources/views/admin/polis/index.blade.php`, pesan ditampilkan menggunakan **alert Bootstrap**:

```
● ● ●  
1 @if (session('success'))  
2     <div class="alert alert-success" id="alert">{{ session('success') }}</div>  
3 @endif
```

Agar user experience lebih baik, notifikasi akan **otomatis menghilang setelah beberapa detik** menggunakan JavaScript:

```
● ● ●  
1 <script>  
2     setTimeout(() => {  
3         const alert = document.getElementById('alert');  
4         if (alert) alert.remove();  
5     }, 3000);  
6 </script>
```

6.5.6 Tampilan Admin: Sidebar & Navigasi

a. Layout Utama

📁 resources/views/layouts/components/app.blade.php

- Berisi struktur dasar halaman admin.
- Ada **sidebar** (kiri), **header**, **footer**, dan `{{ $slot }}` untuk menampilkan konten halaman.
- Template memakai **AdminLTE + Bootstrap**.

b. Sidebar Navigasi

Tambahkan menu baru untuk "Manajemen Dokter" ke dalam sidebar layout, di role admin dibawah Dashboard Admin diatas Manajemen Poli:

resources/views/components/partials/sidebar.blade.php

```
● ● ●  
1 <li class="nav-item">  
2     <a href="{{ route('polis.index') }}" class="nav-link {{ request()->routeIs('polis.*') ? 'active' : '' }}>  
3         <i class="nav-icon fas fa-hospital"></i>  
4         <p>  
5             Manajemen Poli  
6         </p>  
7     </a>  
8 </li>
```

- Sidebar muncul di sebelah kiri.
- Menyediakan menu:
 - **Dashboard Admin** → route `dashboard`
 - **Manajemen Poli** → route `polis.index`

c. Halaman Konten (Index Poli)

📁 resources/views/admin/polis/index.blade.php

- Meng-extend layout dengan <x-layouts.app title="Data Poli">.
 - **x-layouts.app** → memanggil **komponen layout** yang ada di file: 📁 resources/views/layouts/components/app.blade.php
 - **title="Data Poli"** → mengirimkan data (props) ke layout untuk ditampilkan pada bagian <title> halaman.
 - **{{ \$slot }}** di dalam layout akan diganti dengan isi konten halaman, misalnya tabel Data Poli di index.blade.php.
- Menampilkan tabel **Data Poli** (nama_poli, keterangan).
- Ada tombol **Tambah, Edit, Hapus**.
- Menyertakan notifikasi (alert) sukses/gagal.

d. Controller & Model

📁 app/Http/Controllers/PoliController.php

📁 app/Models/Poli.php

- **Controller** mengambil data dari **Model Poli**.
- Data dikirim ke view admin/polis/index.blade.php.

6.6 Kesimpulan

Modul ini mengajarkan kamu:

- Menggunakan **Route::resource()** untuk membuat CRUD routing otomatis.
- Membuat form tambah/edit poli.
- Validasi input di controller menggunakan **request->validate()**.
- Menampilkan notifikasi sukses/gagal menggunakan session flash.
- Mengelola tampilan admin dengan layout + sidebar yang konsisten.



7. Modul CRUD Dokter Oleh Admin

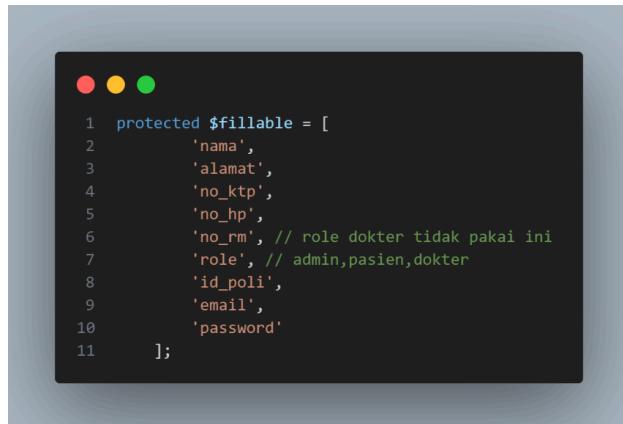
7.0 Overview

Pada modul ini, kita akan membangun fitur **CRUD** data dokter yang bisa diakses oleh **Admin**. Kita akan mulai dari **menambah** data dokter baru, lalu lanjut ke cara **melihat** daftarnya. Tidak hanya itu, kita juga akan bahas cara **mengedit** dan **menghapus** data jika diperlukan. Modul ini akan membahas langkah demi langkah, dari merangkai kode di **controller** dan **routes** hingga membuat **tampilan** yang fungsional. Kita juga akan belajar cara menghubungkan data dokter dengan data poli melalui **relasi** yang tepat. Singkatnya, modul ini membekali kita dengan pemahaman praktis tentang cara membuat fitur pengelolaan data yang aman dan terstruktur.

7.1 Analisis Model Dokter

7.1.1 Struktur Model user.php

Model `User` sudah mendukung entitas dokter dengan kolom dan relasi ke poli:



```
1 protected $fillable = [
2     'nama',
3     'alamat',
4     'no_ktp',
5     'no_hp',
6     'no_rm', // role dokter tidak pakai ini
7     'role', // admin,pasien,dokter
8     'id_poli',
9     'email',
10    'password'
11];
```

7.2 Membuat DokterController

7.2.1 Generate Resource Controller

Gunakan perintah berikut:

```
php artisan make:controller DokterController --resource
```

7.2.2 Isi Method pada Controller

Isi lengkap method seperti index, create, store, edit, update, dan destroy untuk menangani operasi CRUD.

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\Polis;
6 use App\Models\User;
7 use Illuminate\Http\Request;
8 use Illuminate\Support\Facades\Hash;
9
10 class DokterController extends Controller
11 {
12     /**
13      * Display a listing of the resource.
14      */
15     public function index()
16     {
17         // dimana role adalah dokter
18         $dokters = User::where('role', 'dokter')->with('poli')->get();
19         return view('admin.dokter.index', compact('dokters'));
20     }
21
22     /**
23      * Show the form for creating a new resource.
24      */
25     public function create()
26     {
27         $polis = Polis::all();
28         return view('admin.dokter.create', compact('polis'));
29     }
30
31     /**
32      * Store a newly created resource in storage.
33      */
34     public function store(Request $request)
35     {
36         //1. membuat validasi
37         $data = $request->validate([
38             'name' => 'required|string|max:255',
39             'alamat' => 'required|string',
40             'no_ktp' => 'required|string|max:16|unique:users,no_ktp',
41             'no_hp' => 'required|string|max:13',
42             'id_poli' => 'required|integer|exists:poli,id', // intinya id nya ada di poli
43             'email' => 'required|string|unique:users,email',
44             'password' => 'required|string|min:6',
45         ]);
46         // dd($data);
47
48         User::create([
49             'name' => $request->name,
50             'alamat' => $request->alamat,
51             'no_ktp' => $request->no_ktp,
52             'no_hp' => $request->no_hp,
53             'id_poli' => $request->id_poli,
54             'email' => $request->email,
55             'password' => Hash::make($request->password),
56             'role' => 'dokter',
57         ]);
58
59
60
61         return redirect()->route('dokter.index')
62             ->with('message', 'Data Dokter Berhasil di Tambahkan')
63             ->with('type', 'success');
64     }
65
66     /**
67      * Show the form for editing the specified resource.
68      */
69     public function edit(User $dokter)
70     {
71         // $polis = Polis::all(); // Janggap, kam heneran
72         $polis = Polis::all();
73         return view('admin.dokter.edit', compact('dokter', 'polis'));
74     }
75
76     /**
77      * Update the specified resource in storage.
78      * $dokter adalah route model binding jadi yang harus nya kita buat
79      * $dokter = User::findOrFail($id); kita bisa membuat menjadi parameter,
80      * namun jika menggunakan cara tersebut kita route nya tidak bisa admin/dokter{id}/edit namun
81      * seperti admin/dokter/{dokter}/edit
82      */
83
84     public function update(Request $request, User $dokter)
85     {
86         $request->validate([
87             'name' => 'required|string|max:255',
88             'alamat' => 'required|string',
89             'no_ktp' => 'required|string|max:16|unique:users,no_ktp',
90             'no_hp' => 'required|string|max:13|unique:users,no_hp',
91             'id_poli' => 'required|integer|exists:poli,id', // intinya id nya ada di poli
92             'email' => 'required|string|unique:users,email',
93             'password' => 'nullable|string|min:6',
94             'password' => 'nullable|string|min:6',
95         ]);
96
97         $updateData = [
98             'name' => $request->name,
99             'alamat' => $request->alamat,
100            'no_ktp' => $request->no_ktp,
101            'no_hp' => $request->no_hp,
102            'id_poli' => $request->id_poli,
103            'email' => $request->email,
104        ];
105
106         //update password bila password diisi
107         if ($request->tolded('password')) {
108             $dokter->password = Hash::make($request->password);
109         }
110
111         //disimpan
112         $dokter->update($updateData);
113
114         return redirect()->route('dokter.index')
115             ->with('message', 'Data Dokter Berhasil di ubah')
116             ->with('type', 'success');
117     }
118
119     /**
120      * Remove the specified resource from storage.
121      */
122     public function destroy(User $dokter)
123     {
124         $dokter->delete();
125         return redirect()->route('dokter.index')
126             ->with('message', 'Data Dokter Berhasil dihapus')
127             ->with('type', 'success');
128     }
129 }
```

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

7.2.3 Penjelasan Function

1. Function index



```
● ● ●
1 public function index()
2 {
3     // dimana role adalah dokter
4     $dokters = User::where('role', 'dokter')->with('poli')->get();
5     return view('admin.dokter.index', compact('dokters'));
6 }
```

Fungsi ini bertugas untuk menampilkan daftar semua dokter.

Query: `User::where('role', 'dokter')->with('poli')->get();`

- Kode ini mengambil semua data dari tabel **users** di mana kolom **role** bernilai **dokter**.
- Tambahan `->with('poli')` memastikan data **poli** yang terkait dengan dokter juga ikut diambil.

View: `return view('admin.dokter.index', compact('dokters'));`

- Data dokter dikirim ke tampilan `admin.dokter.index` menggunakan `compact('dokters')`.

2. Function create



```
● ● ●
1 public function create()
2 {
3     $polis = Poli::all();
4     return view('admin.dokter.create', compact('polis'));
5 }
```

Fungsi ini hanya bertugas untuk mengarahkan ke halaman formulir tambah data dokter.

Query: `$polis = Poli::all();`

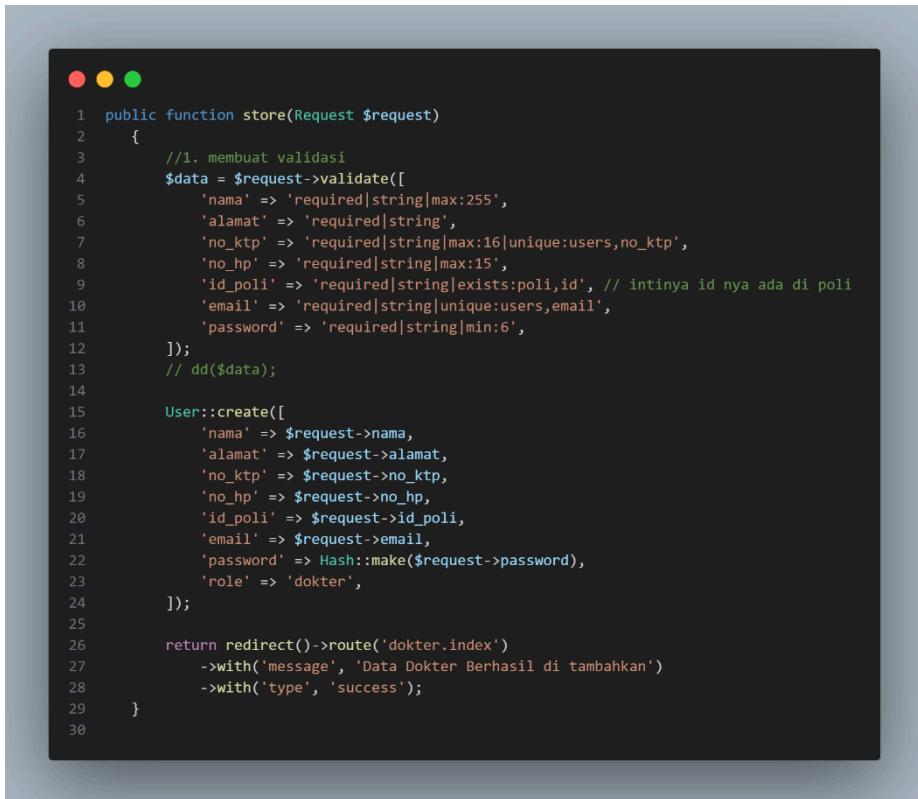
- Kode ini mengambil semua data dari tabel polis.

View: `return view('admin.dokter.create', compact('polis'));`

- Fungsi ini mengarahkan pengguna ke tampilan `admin.dokter.create`, yaitu halaman yang berisi formulir untuk menambah data dokter.

- Data polis dikirimkan ke tampilan tersebut menggunakan `compact('polis')` agar bisa digunakan untuk menampilkan daftar poli yang tersedia.

3. Function store



```

1  public function store(Request $request)
2  {
3      //1. membuat validasi
4      $data = $request->validate([
5          'nama' => 'required|string|max:255',
6          'alamat' => 'required|string',
7          'no_ktp' => 'required|string|max:16|unique:users,no_ktp',
8          'no_hp' => 'required|string|max:15',
9          'id_poli' => 'required|string|exists:poli,id', // intinya id nya ada di poli
10         'email' => 'required|string|unique:users,email',
11         'password' => 'required|string|min:6',
12     ]);
13     // dd($data);
14
15     User::create([
16         'nama' => $request->nama,
17         'alamat' => $request->alamat,
18         'no_ktp' => $request->no_ktp,
19         'no_hp' => $request->no_hp,
20         'id_poli' => $request->id_poli,
21         'email' => $request->email,
22         'password' => Hash::make($request->password),
23         'role' => 'dokter',
24     ]);
25
26     return redirect()->route('dokter.index')
27         ->with('message', 'Data Dokter Berhasil di tambahkan')
28         ->with('type', 'success');
29 }
30

```

Fungsi ini bertugas untuk memproses data dari formulir dan menyimpannya sebagai data dokter baru.

Validasi input: `request()->validate([...]);`

- Sebelum data disimpan, fungsi ini melakukan validasi untuk memastikan semua data yang diisi di formulir sudah benar. Misalnya, `email` dan `no_ktp` harus unik, `password` minimal 6 karakter, dan `id_poli` harus ada di tabel poli.

Query create: `User::create([...]);`

- Data yang sudah divalidasi kemudian disimpan ke dalam tabel users sebagai pengguna baru dengan peran dokter. `Hash::make()` digunakan untuk mengenkripsi password agar aman.

Redirect

- Setelah data berhasil disimpan, pengguna akan diarahkan kembali ke halaman daftar dokter (`'dokter.index'`) dengan pesan sukses.

4. Function edit



```
1 public function edit(User $dokter)
2 {
3     $polis = Poli::all();
4     return view('admin.dokter.edit', compact('dokter', 'polis'));
5 }
```

Fungsi ini bertugas untuk menampilkan halaman formulir edit dengan data dokter yang sudah ada.

- Mengambil data dokter yang ingin diedit (secara otomatis by ID karena route model binding).
- Mengirim data tersebut ke view '`admin.dokter.edit`'.

5. Function update



```
1 public function update(Request $request, User $dokter)
2 {
3     $request->validate([
4         'nama' => 'required|string|max:255',
5         'alamat' => 'required|string',
6         // 'no_ktp' => 'required|string|max:16|unique:users,no_ktp',
7         // "Email harus unik, tapi jangan hitung email si dokter yang ini."
8         // 'no_hp' => 'required|string|max:15',
9         // 'id_poli' => 'required|string|exists:poli,id', // intinya id nya ada di poli
10        // 'email' => 'required|string|unique:users,email,' . $dokter->id,
11        // No KTP harus unik, tapi jangan hitung NO KTP si dokter yang ini."
12        'password' => 'nullable|string|min:6',
13    ]);
14
15
16
17     $updateData = [
18         'nama' => $request->nama,
19         'alamat' => $request->alamat,
20         'no_ktp' => $request->no_ktp,
21         'no_hp' => $request->no_hp,
22         'id_poli' => $request->id_poli,
23         'email' => $request->email,
24     ];
25
26
27     //update password bila password disisi
28     if ($request->filled('password')) {
29         $dokter->password = Hash::make($request->password);
30     }
31
32     //disimpan
33     $dokter->update($updateData);
34
35     return redirect()->route('dokter.index')
36         ->with('message', 'Data Dokter Berhasil di ubah')
37         ->with('type','success');
38 }
```

Fungsi ini bertugas untuk memperbarui data dokter yang sudah ada.

Validasi Unik

- Data `no_ktp` dan `email` tetap unik, tetapi mengabaikan ID pasien yang sedang diedit.

Update Data: `$updateData = [...]`

- Membuat sebuah array bernama `$updateData` yang berisi semua data baru dari form yang ingin diperbarui

Pembaruan Password: `if ($request->filled('password')) { ... }`

- Memeriksa apakah password baru diisi. Jika ya, password lama akan diperbarui dengan password baru yang sudah dienkripsi. Jika tidak, password lama akan tetap utuh.

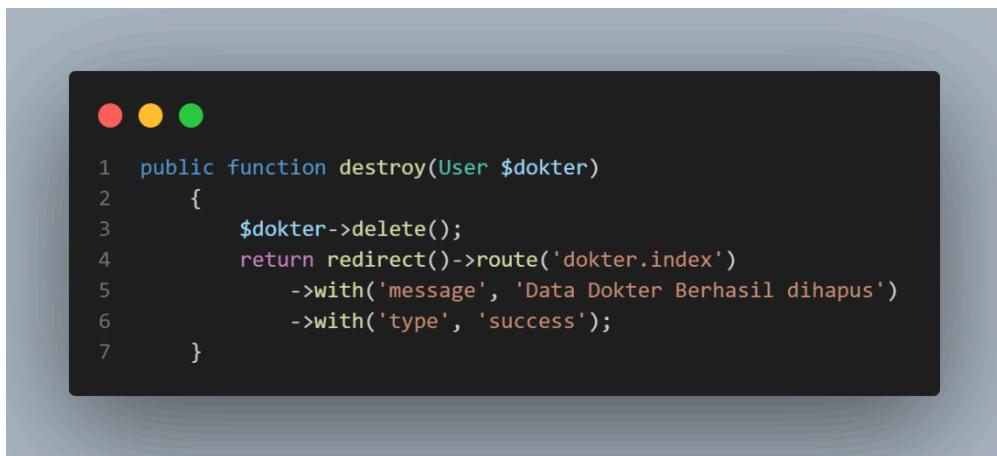
Query Update: `$dokter->update($updateData);`

- Memperbarui data dokter di database dengan data yang sudah diubah dari formulir.

Redirect

- Setelah data berhasil diperbarui, pengguna akan diarahkan kembali ke halaman daftar dokter (`'dokter.index'`) dengan pesan sukses

6. Function destroy



```
 1 public function destroy(User $dokter)
 2 {
 3     $dokter->delete();
 4     return redirect()->route('dokter.index')
 5         ->with('message', 'Data Dokter Berhasil dihapus')
 6         ->with('type', 'success');
 7 }
```

Fungsi ini bertugas untuk menghapus data dokter dari database.

Query: `$dokter->delete();`

- Kode ini menghapus data dokter yang dipilih dari tabel users.

Redirect

- Setelah data berhasil dihapus, pengguna akan diarahkan kembali ke halaman daftar dokter (`'dokter.index'`) dengan pesan sukses

7.3 Menambahkan Routes

Tambahkan ke dalam file routes/web.php:

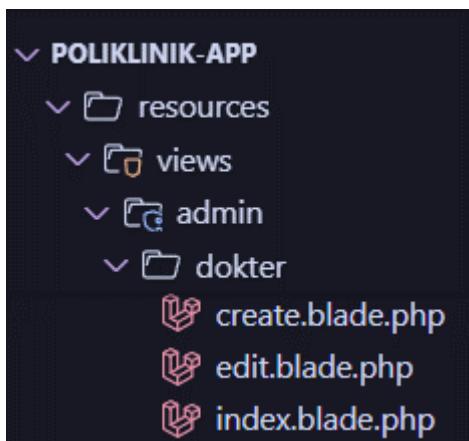


```
1 //Admin
2 Route::middleware(['auth', 'role:admin'])->prefix('admin')->group(function () {
3     Route::get('/dashboard', function () {
4         return view('admin.dashboard');
5     })->name('admin.dashboard');
6     Route::resource('polis', PoliController::class);
7     Route::resource('dokter', DokterController::class);
8     Route::resource('pasien', PasienController::class);
9     Route::resource('obat', ObatController::class);
10 });

```

7.4 Menyiapkan Tampilan Blade

Struktur folder:



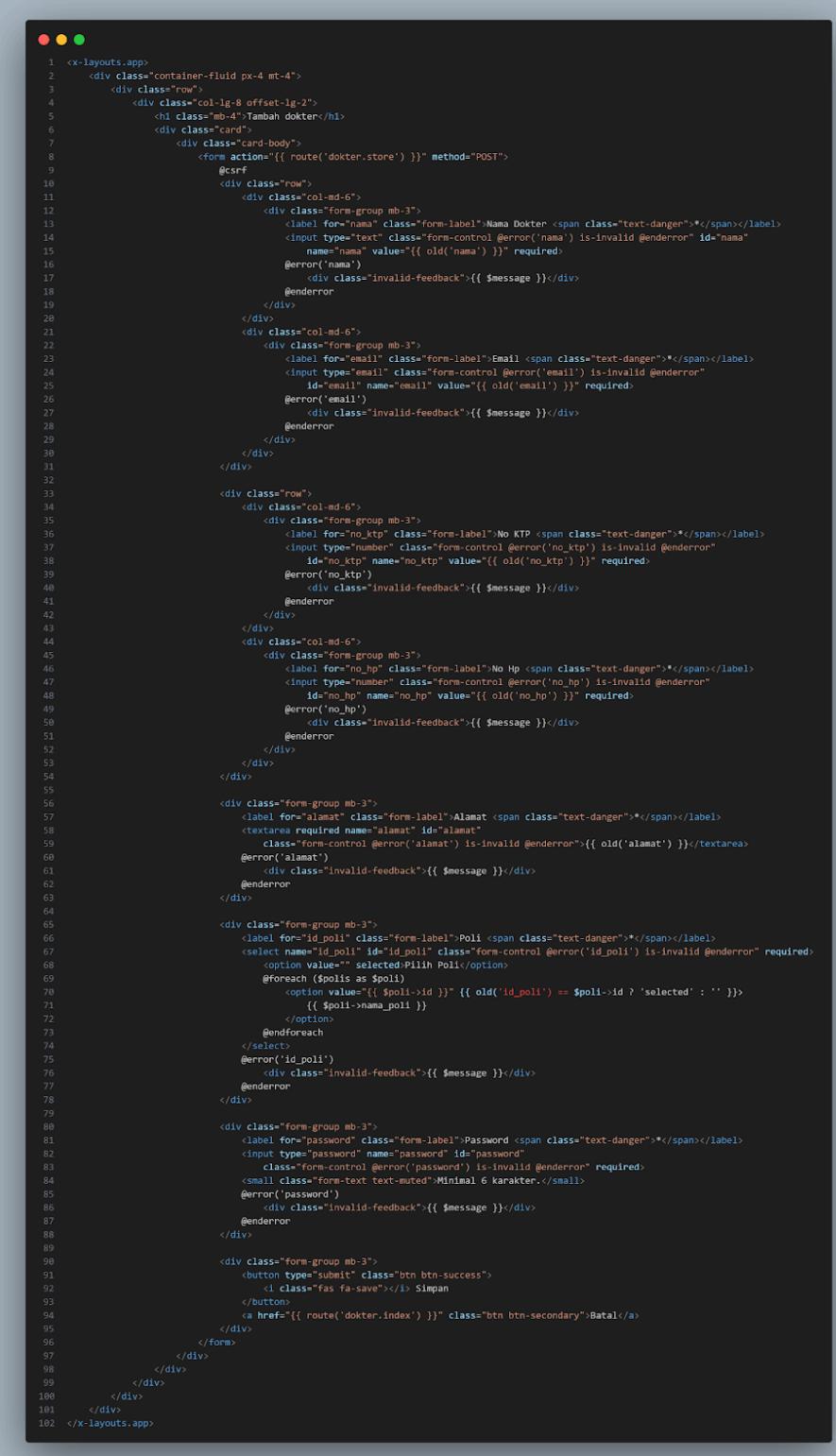
7.4.1 dokter/index.blade.php

```
1  <x-layouts.app title="Data Dokter">
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">Data Dokter</h1>
15
16             <a href="{{ route('dokter.create') }}" class="btn btn-primary mb-3">
17                 <i class="fas fa-plus"></i>Tambah dokter
18             </a>
19
20             <div class="table-responsive">
21                 <table class="table table-bordered">
22                     <thead class="thead-light">
23                         <tr>
24                             <th>Nama Dokter</th>
25                             <th>Email</th>
26                             <th>No. KTP</th>
27                             <th>No. HP</th>
28                             <th>Alamat</th>
29                             <th>Poli</th>
30                             <th style="width: 150px;">Aksi</th>
31                         </tr>
32                     </thead>
33                     <tbody>
34                         @forelse ($dokters as $dokter )
35                             <tr>
36                                 <td>{{ $dokter->nama }}</td>
37                                 <td>{{ $dokter->email }}</td>
38                                 <td>{{ $dokter->no_ktp }}</td>
39                                 <td>{{ $dokter->no_hp }}</td>
40                                 <td>{{ $dokter->alamat }}</td>
41                                 <td>
42                                     <span class="badge bg-info">
43                                         {{ $dokter->poli }} ?? 'Belum Di pilih' 
44                                     </span>
45                                 </td>
46                                 <td>
47                                     <a href="{{ route('dokter.edit', $dokter->id) }}" class="btn btn-sm btn-warning">
48                                         <i class="fas fa-edit"></i>Edit
49                                     </a>
50                                     <form action="{{ route('dokter.destroy', $dokter->id) }}" method="POST" style="display: inline-block;">
51                                         @csrf
52                                         @method('DELETE')
53                                         <button class="btn btn-sm btn-danger" onclick="return confirm('Yakin ingin menghapus dokter ini ?')">
54                                             <i class="fas fa-trash"></i> Hapus
55                                         </button>
56                                     </form>
57                                 </td>
58                             </tr>
59                         @empty
60                             <tr>
61                                 <td class="text-center" colspan="7">
62                                     Belum ada Dokter
63                                 </td>
64                             </tr>
65                         @endforelse
66                     </tbody>
67                 </table>
68             </div>
69         </div>
70     </div>
71 </div>
72
73     <script>
74         setTimeout(() => {
75             const alert = document.querySelector('.alert');
76             if (alert) {
77                 alert.classList.remove('show');
78                 alert.classList.add('fade');
79                 setTimeout(() => alert.remove(), 500);
80             }
81         }, 2000);
82     </script>
83
84 </x-layouts.app>
85
```

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

7.4.2 dokter/create.blade.php

Form untuk menambahkan data dokter baru. Field: nama, email, no KTP, no HP, alamat, poli (dropdown), dan password.



```
1 <x-layouts.app>
2   <div class="container-fluid px-4 mt-4">
3     <div class="row">
4       <div class="col-lg-8 offset-lg-2">
5         <h1 class="mb-4">Tambah dokter</h1>
6         <div class="card">
7           <div class="card-body">
8             <form action="{{ route('dokter.store') }}" method="POST">
9               @csrf
10              <div class="row">
11                <div class="col-md-6">
12                  <div class="form-group mb-3">
13                    <label for="nama" class="form-label">Nama Dokter <span class="text-danger"><!-->{{ $message }}</div>
18                    @enderror
19                  </div>
20                </div>
21                <div class="col-md-6">
22                  <div class="form-group mb-3">
23                    <label for="email" class="form-label">Email <span class="text-danger"><!-->{{ $message }}</div>
28                    @enderror
29                  </div>
30                </div>
31              </div>
32            <div class="row">
33              <div class="col-md-6">
34                <div class="form-group mb-3">
35                  <label for="no_ktp" class="form-label">No KTP <span class="text-danger"><!-->{{ $message }}</div>
40                  @enderror
41                </div>
42              </div>
43              <div class="col-md-6">
44                <div class="form-group mb-3">
45                  <label for="no_hp" class="form-label">No Hp <span class="text-danger"><!-->{{ $message }}</div>
50                  @enderror
51                </div>
52              </div>
53            </div>
54          </div>
55          <div class="form-group mb-3">
56            <label for="alamat" class="form-label">Alamat <span class="text-danger"><!-->{{ old('alamat') }}{{ $message }}</div>
61            @enderror
62          </div>
63        </div>
64        <div class="form-group mb-3">
65          <label for="id_poli" class="form-label">Poli <span class="text-danger"><!-->Pilih Poli</option>
68            @foreach ($polis as $poli)
69              <option value="{{ $poli->id }}" {{ old('id_poli') == $poli->id ? 'selected' : '' }}>
70                {{ $poli->nama_poli }}
71              </option>
72            @endforeach
73          </select>
74          @error('id_poli')
75            <div class="invalid-feedback">{{ $message }}</div>
76          @enderror
77        </div>
78        <div class="form-group mb-3">
79          <label for="password" class="form-label">Password <span class="text-danger"><!-->Minimal 6 karakter.</small>
83          @error('password')
84            <div class="invalid-feedback">{{ $message }}</div>
85          @enderror
86        </div>
87        <div class="form-group mb-3">
88          <button type="submit" class="btn btn-success">
89            <i class="fas fa-save"></i> Simpan
90          </button>
91          <a href="{{ route('dokter.index') }}" class="btn btn-secondary">Batal</a>
92        </div>
93      </div>
94    </div>
95  </div>
96</x-layouts.app>
```

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

7.4.3: dokter/edit.blade.php

Form untuk mengedit data dokter. Field sama seperti create, tetapi password bersifat opsional.



```
1 <x>layouts.app
2   <div class="container-fluid px-4 mt-4">
3     <div class="row">
4       <div class="col-lg-8 offset-lg-2">
5         <div class="mb-4">Edit Dokter</div>
6         <div class="card">
7           <div class="card-body">
8             <form action="{{ route('dokter.update', $dokter->id) }}" method="POST">
9               @csrf
10              @method("PUT")
11
12              <div class="row">
13                <div class="col-md-6">
14                  <div class="form-group mb-3">
15                    <label for="nama" class="form-label">Nama Dokter <span
16                      class="text-danger"><*></span></label>
17                    <input type="text" class="form-control @error('nama') is-invalid @enderror"
18                      id="nama" name="nama" value="{{ old('nama', $dokter->nama) }}"
19                      required>
20                    @error('nama')
21                      <div class="invalid-feedback">{{ $message }}</div>
22                    @enderror
23                  </div>
24                </div>
25                <div class="col-md-6">
26                  <div class="form-group mb-3">
27                    <label for="email" class="form-label">Email <span
28                      class="text-danger"><*></span></label>
29                    <input type="email" class="form-control @error('email') is-invalid @enderror"
30                      id="email" name="email" value="{{ old('email', $dokter->email) }}"
31                      required>
32                    @error('email')
33                      <div class="invalid-feedback">{{ $message }}</div>
34                    @enderror
35                  </div>
36                </div>
37              </div>
38
39              <div class="row">
40                <div class="col-md-6">
41                  <div class="form-group mb-3">
42                    <label for="no_ktp" class="form-label">No KTP <span
43                      class="text-danger"><*></span></label>
44                    <input type="number" class="form-control @error('no_ktp') is-invalid @enderror"
45                      id="no_ktp" name="no_ktp" value="{{ old('no_ktp', $dokter->no_ktp) }}"
46                      required>
47                    @error('no_ktp')
48                      <div class="invalid-feedback">{{ $message }}</div>
49                    @enderror
50                  </div>
51                </div>
52                <div class="col-md-6">
53                  <div class="form-group mb-3">
54                    <label for="no_hp" class="form-label">No Hp <span
55                      class="text-danger"><*></span></label>
56                    <input type="number" class="form-control @error('no_hp') is-invalid @enderror"
57                      id="no_hp" name="no_hp" value="{{ old('no_hp', $dokter->no_hp) }}"
58                      required>
59                    @error('no_hp')
60                      <div class="invalid-feedback">{{ $message }}</div>
61                    @enderror
62                  </div>
63                </div>
64
65              <div class="form-group mb-3">
66                <label for="alamat" class="form-label">Alamat <span
67                  class="text-danger"><*></span></label>
68                <textarea required name="alamat" id="alamat" class="form-control @error('alamat') is-invalid @enderror{{ old('alamat', $dokter->alamat) }}</textarea>
69                @error('alamat')
70                  <div class="invalid-feedback">{{ $message }}</div>
71                @enderror
72              </div>
73
74              <div class="form-group mb-3">
75                <label for="id_poli" class="form-label">Poli <span class="text-danger"><*></span></label>
76                <select name="id_poli" id="id_poli"
77                  class="form-control @error('id_poli') is-invalid @enderror" required>
78                  <option value="" {{ old('id_poli', $dokter->id_poli) ? '' : 'selected' }}>
79                    Pilih Poli</option>
80                  @foreach ($polis as $poli)
81                    <option value="{{ $poli->id }}>
82                      {{ (string) $poli->id_poli, $dokter->id_poli } == (string) $poli->id ? 'selected' : '' }}>
83                      {{ $poli->nama_poli ?? 'Poli Tidak Dikenal' }}</option>
84                  @endforeach
85                  <option>
86                    @enderror
87                  </select>
88                  @error('id_poli')
89                    <div class="invalid-feedback">{{ $message }}</div>
90                  @enderror
91              </div>
92
93              <div class="form-group mb-3">
94                <label for="password" class="form-label">Password <span
95                  class="text-danger"><*></span></label>
96                <input type="password" name="password" id="password"
97                  class="form-control @error('password') is-invalid @enderror"
98                  small class="form-text text-muted">Minimal 6 Karakter.</small>
99                @error('password')
100                  <div class="invalid-feedback">{{ $message }}</div>
101                @enderror
102              </div>
103
104              <div class="form-group mb-3">
105                <button type="submit" class="btn btn-success">
106                  <i class="fas fa-save"></i> Update
107                </button>
108                <a href="{{ route('dokter.index') }}" class="btn btn-secondary">kembali</a>
109              </div>
110            </form>
111        </div>
112      </div>
113    </div>
114  </div>
115 </x>layouts.app
116 <1>
```

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

7.5 Menambahkan Sidebar Navigasi

Tambahkan menu baru untuk "Manajemen Dokter" ke dalam sidebar layout, di role admin dibawah Dashboard Admin diatas Manajemen Poli:

`resources/views/components/partials/sidebar.blade.php`



```
1 <li class="nav-item">
2   <a href="{{ route('dokter.index') }}" class="nav-link {{ request()->routeIs('dokter.*') ? 'active' : '' }}">
3     <i class="nav-icon fas fa-user-md"></i>
4     <p>
5       Manajemen Dokter
6     </p>
7   </a>
8 </li>
```

7.6 Testing Fitur

7.6.1 Jalankan Migration & Seeder

```
php artisan migrate
php artisan db:seed --class=UserSeeder
```

7.6.2 Login Sebagai Admin

- Email: admin@gmail.com
- Password: admin

7.6.3 Uji Semua Fitur

- Tambah dokter
- Edit dokter
- Hapus dokter
- Lihat relasi poli di daftar dokter