

# Pertemuan 4

## Tujuan Pembelajaran:

Tujuan Pembelajaran yang harus dicapai dalam modul ini adalah mahasiswa mampu:

- Membuat form input pasien
- Menghubungkan relasi dokter ke poli
- Menampilkan tabel daftar dokter
- Mengimplementasikan fitur edit dan hapus data dokter
- Memahami cara membuat fitur CRUD (Create, Read, Update, Delete) untuk data obat
- Membuat Formulir Tambah Obat
- Menampilkan Daftar Obat dalam Tabel
- Melakukan Validasi Input Data Obat

## Modul 8: CRUD PASIEN OLEH ADMIN

---

### 8.0 OVERVIEW

Di modul ini, kita akan membuat fitur CRUD untuk data pasien. Pertama, kita akan mulai dengan menampilkan daftar pasien yang sudah tersimpan di database. Setelah itu, kita akan belajar menambahkan pasien baru melalui form input yang terhubung langsung ke sistem. Selanjutnya, kita juga akan membahas bagaimana cara mengedit data pasien ketika ada perubahan informasi, serta menghapus data pasien yang sudah tidak diperlukan. Supaya lebih rapi, kita akan menambahkan validasi input, seperti memastikan nomor KTP dan email tetap unik serta mengamankan password dengan hashing.

Terakhir, agar lebih mudah diakses, kita juga akan menambahkan menu Manajemen Pasien pada sidebar. Dengan begitu, kita bisa memahami alur lengkap dari sebuah fitur CRUD di Laravel khusus untuk pengelolaan data pasien.

The screenshot shows a dark-themed Laravel application interface. On the left is a sidebar with a navigation menu. The 'Manajemen Pasien' item is highlighted with a blue background. The main content area is titled 'Data Pasien' and contains a table with one row of data. The table columns are 'Nama Pasien', 'Email', 'No. KTP', 'NO. HP', 'Alamat', and 'Aksi'. The 'Aksi' column contains two buttons: 'Edit' (yellow background) and 'Hapus' (red background). At the bottom of the main content area, there is a copyright notice: 'Copyright ©2024 Poliklinik 2024. All right reserved.'

## 8.1 Membuat PasienController

### 8.1.1 Generate Resource Controller

Di langkah ini kita membuat controller dengan perintah:

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

Dengan opsi `--resource`, Laravel otomatis menyediakan method dasar untuk CRUD.

### 8.1.2 Isi Method pada Controller

Pada tahap ini kita melengkapi setiap method agar bisa menangani data pasien.

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

## app/Http/Controllers/PasienController.php



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\User;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Hash;
8
9 class PasienController extends Controller
10 {
11     public function index()
12     {
13         $pasiens = User::where('role', 'pasien')->with('poli')->get();
14         return view('admin.pasien.index', compact('pasiens'));
15     }
16
17     public function create()
18     {
19         return view('admin.pasien.create');
20     }
21
22     public function store(Request $request)
23     {
24         $request->validate([
25             'nama' => 'required|string|max:255',
26             'alamat' => 'required|string',
27             'no_ktp' => 'required|string|max:16|unique:users,no_ktp',
28             'no_hp' => 'required|string|max:15',
29             'email' => 'required|string|unique:users,email',
30             'password' => 'required|string|min:6',
31         ]);
32
33         User::create([
34             'nama' => $request->nama,
35             'alamat' => $request->alamat,
36             'no_ktp' => $request->no_ktp,
37             'no_hp' => $request->no_hp,
38             'email' => $request->email,
39             'password' => Hash::make($request->password),
40             'role' => 'pasien',
41         ]);
42
43         return redirect()->route('pasien.index')->with('message', 'Data Pasien berhasil di Tambah')->with('type', 'success');
44     }
45
46     public function edit(User $pasien)
47     {
48         return view('admin.pasien.edit', compact('pasien'));
49     }
50
51     public function update(Request $request, User $pasien)
52     {
53         $request->validate([
54             'nama' => 'required|string|max:255',
55             'alamat' => 'required|string',
56             'no_ktp' => 'required|string|max:16|unique:users,no_ktp, ' . $pasien->id,
57             'no_hp' => 'required|string|max:15',
58             'email' => 'required|string|unique:users,email,' . $pasien->id,
59             'password' => 'nullable|string|min:6',
60         ]);
61
62         $updateData = [
63             'nama' => $request->nama,
64             'alamat' => $request->alamat,
65             'no_ktp' => $request->no_ktp,
66             'no_hp' => $request->no_hp,
67             'email' => $request->email
68         ];
69
70         if($request->filed('password')) {
71             $updateData['password'] = Hash::make($request->password);
72         }
73
74         $pasien->update($updateData);
75
76         return redirect()->route('pasien.index')
77             ->with('message','Data Pasien Berhasil di Update')
78             ->with('type','success');
79     }
80
81
82     public function destroy(User $pasien){
83         $pasien->delete();
84         return redirect()->route('pasien.index')
85             ->with('message','Data Pasien Berhasil Di Hapus')
86             ->with('type', 'success');
87     }
88 }
89
90 }
```

Link code untuk PasienController:

<https://drive.google.com/file/d/1wN6wZ4tfEKLTRFVw0DgpuYQA7Inq8P5R/view?usp=sharing>

### 8.1.3 Penjelasan

#### 1. Function Index



```
● ● ●

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

Mengambil semua data pasien dari database dengan query, data dikirim ke view `admin.pasien.index`

- Query: `User::where('role', 'pasien')->get();`  
Mengambil semua data dari tabel users di mana kolom role bernilai pasien.
- View: Data pasien dikirim ke `admin.pasien.index` menggunakan `compact('pasien')`.

#### 2. Function Create

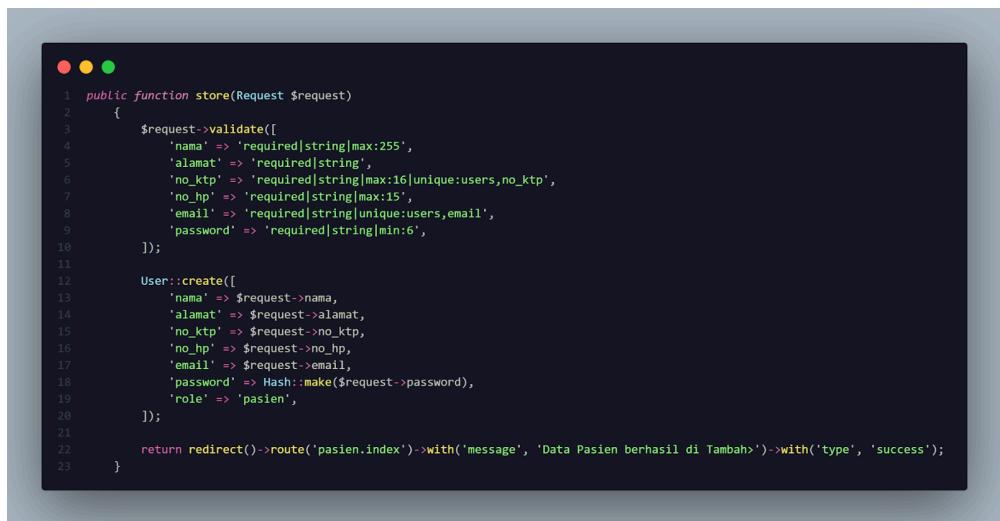


```
● ● ●

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

Hanya menampilkan form untuk menambahkan pasien baru. View yang dituju adalah `admin.pasien.create`.

### 3. Function Store



```
1 public function store(Request $request)
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         'no_hp' => 'required|string|max:15',
8         'email' => 'required|string|unique:users,email',
9         'password' => 'required|string|min:6',
10    ]);
11
12    User::create([
13        'nama' => $request->nama,
14        'alamat' => $request->alamat,
15        'no_ktp' => $request->no_ktp,
16        'no_hp' => $request->no_hp,
17        'email' => $request->email,
18        'password' => Hash::make($request->password),
19        'role' => 'pasien',
20    ]);
21
22    return redirect()->route('pasien.index')->with('message', 'Data Pasien berhasil di Tambah')->with('type', 'success');
23 }
```

Berfungsi untuk menyimpan data pasien baru ke database.

- Validasi input: Pastikan semua data sesuai ketentuan. Misalnya:
  - `no_ktp` dan `email` harus unik.
  - `password` minimal 6 karakter.
- Query create: `User::create([ ... ]);`
  - Menyimpan data pasien ke tabel users.
  - Password diamankan menggunakan `Hash::make()`.
- Redirect: Kembali ke daftar pasien dengan pesan sukses

### 4. Function Edit



```
1 public function edit(User $pasien)
2 {
3     return view('admin.pasien.edit', compact('pasien'));
4 }
```

- Mengambil data pasien yang ingin diedit (secara otomatis by ID karena route model binding).
- Mengirim data tersebut ke view `admin.pasien.edit`. Data ini akan ditampilkan di form edit agar bisa diperbarui.

## 5. Function Update



```
1  public function update(Request $request, User $pasien)
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,' . $pasien->id,
7          'no_hp' => 'required|string|max:15',
8          'email' => 'required|string|unique:users,email,' . $pasien->id,
9          'password' => 'nullable|string|min:6',
10     ]);
11
12     $updateData = [
13         'nama' => $request->nama,
14         'alamat' => $request->alamat,
15         'no_ktp' => $request->no_ktp,
16         'no_hp' => $request->no_hp,
17         'email' => $request->email
18     ];
19
20     if($request->filled('password')) {
21         $updateData['password'] = Hash::make($request->password);
22     }
23
24
25     $pasien->update($updateData);
26
27     return redirect()->route('pasien.index')
28         ->with('message','Data Pasien Berhasil di Update')
29         ->with('type','success');
30 }
```

Bertugas memperbarui data pasien yang sudah ada.

- Validasi input: Data `no_ktp` dan `email` tetap unik, tetapi mengabaikan ID pasien yang sedang diedit.
- Update data: Membuat array `$updateData` berisi field yang ingin diubah. Jika password diisi, password di-hash dan dimasukkan ke `$updateData`.
- Query update: `$pasien->update($updateData);` Mengupdate data pasien di tabel `users`.
- Redirect Kembali ke daftar pasien dengan pesan sukses.

## 6. Function Destroy

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

Digunakan untuk menghapus data pasien.

- Menghapus data pasien dari tabel users menggunakan:  
`$pasien->delete();`
- Lalu mengarahkan kembali ke daftar pasien dengan notifikasi sukses.

## 8.2 MENAMBAHKAN ROUTES

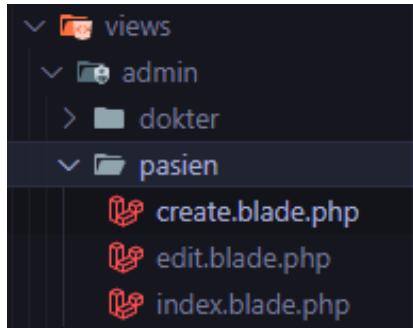
Tambahkan ke dalam file `routes/web.php`

```
Route::resource('pasien', PasienController::class);
```

```
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      Route::resource('dokter', DokterController::class);
7      Route::resource('pasien', PasienController::class);
8  });
});
```

## 8.3 MENYIAPKAN TAMPILAN

Membuat struktur folder sesuai dengan gambar berikut:



### 8.3.1 pasien/index.blade.php

Menampilkan daftar pasien dalam tabel.

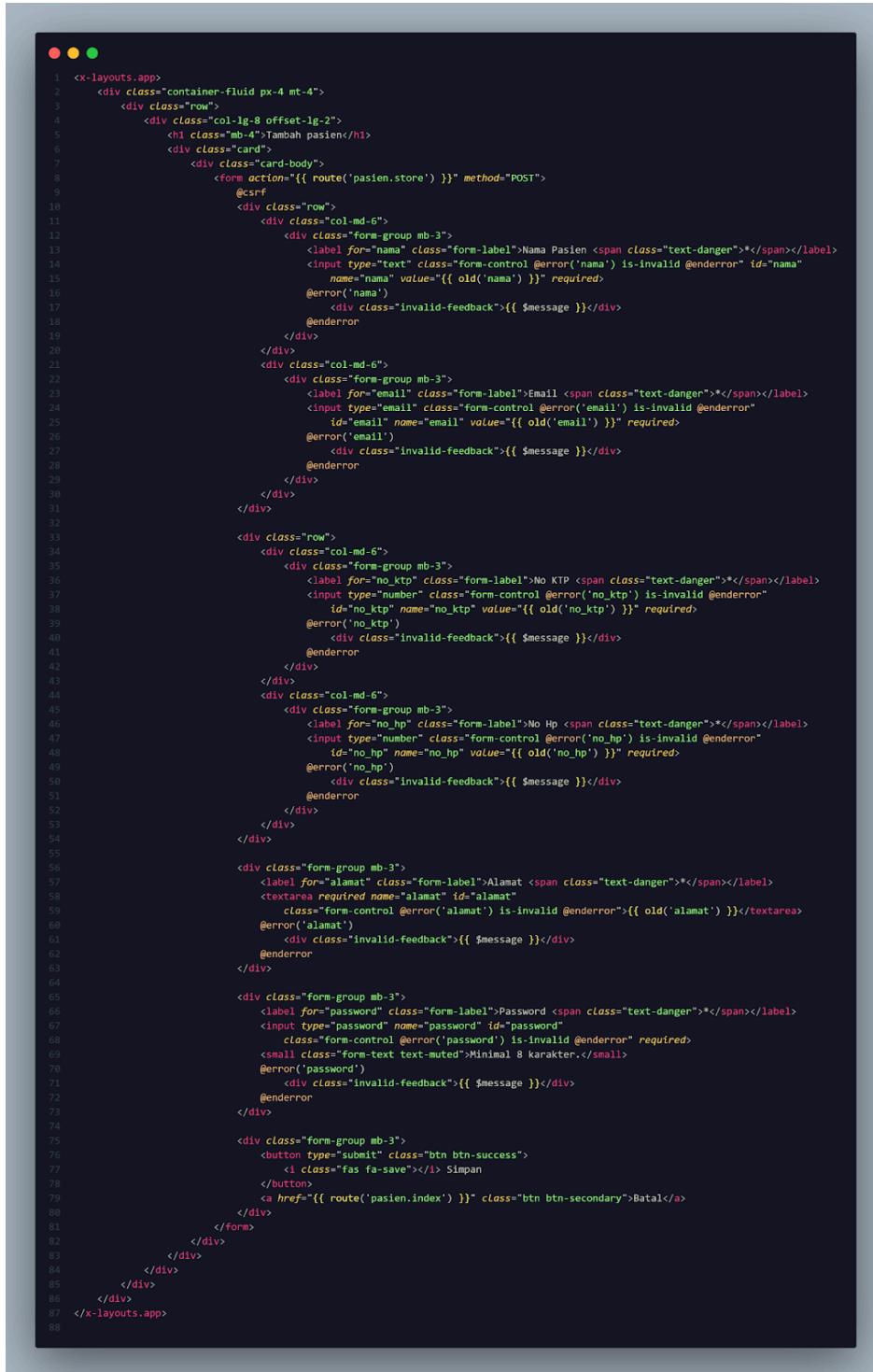
```
1 <x-layouts.app title="Data Pasien">
2   <div class="container-fluid px-4 mt-4">
3     <div class="row">
4       <div class="col-lg-12">
5         {{-- Alert flash message --}}
6         @if (session('message'))
7           @if (session('type') == 'success')
8             <div class="alert alert-success" 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        <h1 class="mb-4">Data Pasien</h1>
14
15        <a href="{{ route('pasien.create') }}" class="btn btn-primary mb-3">
16          <i class="fas fa-plus"></i> Tambah Pasien
17        </a>
18
19        <div class="table-responsive">
20          <table class="table table-bordered">
21            <thead class="thead-light">
22              <tr>
23                <th>Nama Pasien</th>
24                <th>Email</th>
25                <th>No. KTP</th>
26                <th>NO. HP</th>
27                <th>Alamat</th>
28                <th style="width: 150px;">Aksi</th>
29              </tr>
30            </thead>
31            <tbody>
32              @forelse ($pasiens as $pasien)
33                <tr>
34                  <td>{{ $pasien->name }}</td>
35                  <td>{{ $pasien->email }}</td>
36                  <td>{{ $pasien->no_ktp }}</td>
37                  <td>{{ $pasien->no_hp }}</td>
38                  <td>{{ $pasien->alamat }}</td>
39                  <td>
40                    <a href="{{ route('pasien.edit', $pasien->id) }}" class="btn btn-sm btn-warning">
41                      <i class="fas fa-edit"></i> Edit
42                    </a>
43                    <form action="{{ route('pasien.destroy', $pasien->id) }}" method="POST" style="display: inline-block;">
44                      @csrf
45                      @method('DELETE')
46                      <button class="btn btn-sm btn-danger" onclick="return confirm('Yakin ingin menghapus pasien ini ?')">
47                        <i class="fas fa-trash"></i> Hapus
48                      </button>
49                    </form>
50                  </td>
51                </tr>
52              @empty
53                <tr>
54                  <td class="text_center" colspan="7">
55                    Belum ada Pasien
56                  </td>
57                </tr>
58              @endif
59            </tbody>
60          </table>
61        </div>
62      </div>
63
64      <script>
65        setTimeout(() => {
66          const alert = document.querySelector('.alert');
67          if (alert) {
68            alert.classList.remove('show');
69            alert.classList.add('fade');
70            setTimeout(() => alert.remove(), 500);
71          }
72        }, 2000);
73      </script>
74    </x-layouts.app>
75  
```

Link file txt (index.blade.php) :

[https://drive.google.com/file/d/1CNABXeC4r85MM0laTK3MrGAwhyxRB04U/view?usp=drive\\_link](https://drive.google.com/file/d/1CNABXeC4r85MM0laTK3MrGAwhyxRB04U/view?usp=drive_link)

### 8.3.2 pasien/create.blade.php

Perhatikan khususnya pada bagian form, `name` harus sesuai dengan validasi yang ada di function `store` di PasienController.



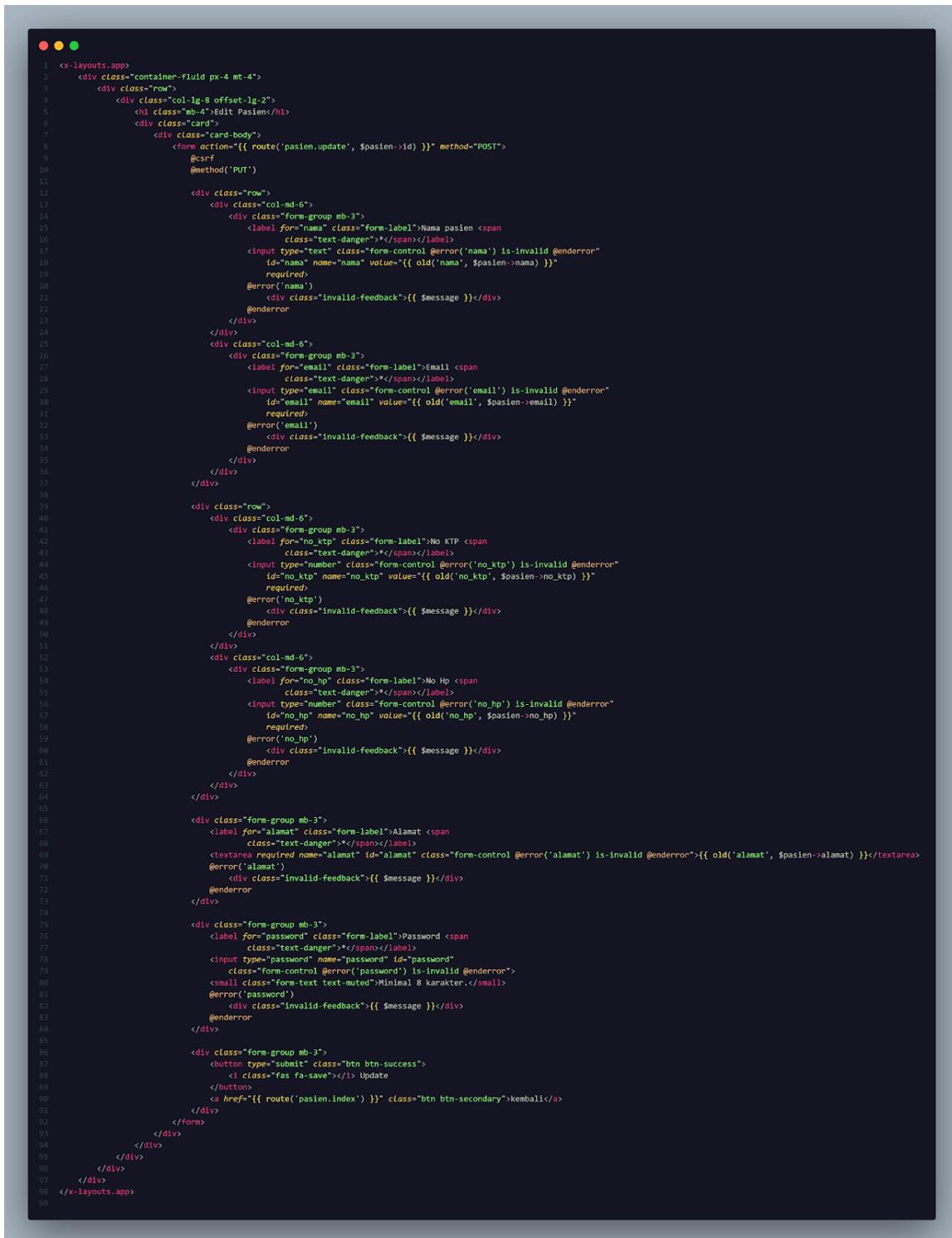
```
1 <x-layouts.app>
2   <div class="container-fluid px-4 mt-4">
3     <div class="row">
4       <div class="col-lg-8 offset-lg-2">
5         <h1 class="mb-4">Tambah pasien</h1>
6         <div class="card">
7           <div class="card-body">
8             <form action="{{ route('pasien.store') }}" 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 Pasien <span class="text-danger">*</span></label>
14                    <input type="text" class="form-control @error('nama') is-invalid @enderror" id="nama"
15                      name="nama" value="{{ old('nama') }}" required>
16                    @error('nama')
17                      <div class="invalid-feedback">{{ $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">*</span></label>
24                    <input type="email" class="form-control @error('email') is-invalid @enderror"
25                      id="email" name="email" value="{{ old('email') }}" required>
26                    @error('email')
27                      <div class="invalid-feedback">{{ $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">*</span></label>
36                    <input type="number" class="form-control @error('no_ktp') is-invalid @enderror"
37                      id="no_ktp" name="no_ktp" value="{{ old('no_ktp') }}" required>
38                    @error('no_ktp')
39                      <div class="invalid-feedback">{{ $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">*</span></label>
46                    <input type="number" class="form-control @error('no_hp') is-invalid @enderror"
47                      id="no_hp" name="no_hp" value="{{ old('no_hp') }}" required>
48                    @error('no_hp')
49                      <div class="invalid-feedback">{{ $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">*</span></label>
57              <textarea required name="alamat" id="alamat"
58                class="form-control @error('alamat') is-invalid @enderror">{{ old('alamat') }}</textarea>
59              @error('alamat')
60                <div class="invalid-feedback">{{ $message }}</div>
61              @enderror
62            </div>
63            <div class="form-group mb-3">
64              <label for="password" class="form-label">Password <span class="text-danger">*</span></label>
65              <input type="password" name="password" id="password"
66                class="form-control @error('password') is-invalid @enderror" required>
67              @error('password')
68                <small class="form-text text-muted">Minimal 8 karakter.</small>
69              @enderror
70              <div class="invalid-feedback">{{ $message }}</div>
71            </div>
72            <div class="form-group mb-3">
73              <button type="submit" class="btn btn-success">
74                <i class="fas fa-save"></i> Simpan
75              </button>
76              <a href="{{ route('pasien.index') }}" class="btn btn-secondary">Batal</a>
77            </div>
78          </div>
79        </div>
80      </div>
81    </div>
82  </div>
83</x-layouts.app>
```

Link file txt (create.blade.php) :

[https://drive.google.com/file/d/1AzPZcQxiQrwK9kZwF2i2ut1FtxyhS8HB/view?usp=drive\\_link](https://drive.google.com/file/d/1AzPZcQxiQrwK9kZwF2i2ut1FtxyhS8HB/view?usp=drive_link)

### 8.3.3 pasien/edit.blade.php

Form untuk mengubah data pasien, strukturnya cukup mirip dengan create baik name dan bentuk input.



```
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">Edit Pasien</h1>
6         <div class="card">
7           <div class="card-body">
8             <form action="{{ route('pasien.update', $pasien->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 pasien <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', $pasien->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', $pasien->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', $pasien->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', $pasien->no_hp) }}"
58                      required
59                      @error('no_hp')
60                        <div class="invalid-feedback">{{ $message }}</div>
61                      @enderror
62                </div>
63              </div>
64            </div>
65
66            <div class="form-group mb-3">
67              <label for="alamat" class="form-label">Alamat <span
68                  class="text-danger">*</span></label>
69              <textarea required name="alamat" id="alamat" class="form-control @error('alamat') is-invalid @enderror{{ old('alamat', $pasien->alamat) }}</textarea>
70              @error('alamat')
71                <div class="invalid-feedback">{{ $message }}</div>
72              @enderror
73            </div>
74
75            <div class="form-group mb-3">
76              <label for="password" class="form-label">Password <span
77                  class="text-danger">*</span></label>
78              <input type="password" name="password" id="password"
79                  class="form-control @error('password') is-invalid @enderror{{ old('password', $pasien->password) }}<br>
80              <small class="form-text text-muted">Minimal 8 karakter.</small>
81              @error('password')
82                <div class="invalid-feedback">{{ $message }}</div>
83              @enderror
84            </div>
85
86            <div class="form-group mb-3">
87              <button type="submit" class="btn btn-success">
88                <i class="fas fa-save"></i> Update
89              </button>
90              <a href="{{ route('pasien.index') }}" class="btn btn-secondary">kembali</a>
91            </div>
92          </form>
93        </div>
94      </div>
95    </div>
96  </div>
97 </x-layouts.app>
```

Link file txt (edit.blade.php) :

[https://drive.google.com/file/d/1Oj3OrbcikzSsMoCqpsW9wrxXEw9OTReA/view?usp=drive\\_link](https://drive.google.com/file/d/1Oj3OrbcikzSsMoCqpsW9wrxXEw9OTReA/view?usp=drive_link)

## 8.4 MENAMBAHKAN SIDEBAR NAVIGASI

Supaya fitur mudah diakses, kita tambahkan menu baru untuk “Manajemen Pasien” ke dalam sidebar layout. Tempatkan setelah menu Manajemen Dokter.



```
1      <li class="nav-item">
2          <a href="{{ route('pasien.index') }}" 
3              class="nav-link {{ request()->routeIs('pasien.*') ? 'active' : '' }}>
4                  <i class="nav-icon fas fa-user-injured"></i>
5                  <p>
6                      Manajemen Pasien
7                  </p>
8          </a>
9      </li>
10
11
```

## 8.5 TESTING FITUR

Tahap terakhir adalah mencoba semua fungsi CRUD.

### 8.5.1 Login Sebagai Admin

Email: [admin@gmail.com](mailto:admin@gmail.com)

Password: admin

### 8.5.2 Uji Semua Fitur

- Tambah pasien baru
- Edit pasien
- Hapus pasien

## 9. Modul CRUD Obat oleh admin

### Overview :

Di modul ini, kita akan belajar bagaimana membuat fitur CRUD untuk data obat pada role admin. Pertama, kita akan mulai dengan menampilkan daftar obat yang sudah tersimpan di database supaya kita bisa melihat data obat yang ada. Setelah itu, kita akan belajar cara menambahkan data obat baru lewat form input yang langsung tersambung dengan database.

Tidak berhenti di sana, kita juga akan membahas bagaimana cara mengedit data obat jika ada perubahan, serta menghapus data obat yang tidak lagi dibutuhkan. Supaya proses input data lebih rapi dan sesuai aturan, kita akan menambahkan validasi pada form input. Dengan modul ini, kita akan memahami alur lengkap pembuatan fitur CRUD di Laravel mulai dari input, edit, hapus, hingga menampilkan data.

### 9.1. Menganalisis Model



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Obat extends Model
8 {
9     protected $table = 'obat';
10
11    protected $fillable = [
12        'nama_obat',
13        'kemasan',
14        'harga'
15    ];
16
17    public function detailPeriksas(){
18        return $this->hasMany(DetailPeriksa::class, 'id_obat');
19    }
20 }
21
```

Model Obat merepresentasikan tabel "obat" dalam database dan menentukan kolom-kolom yang boleh diisi atau diubah, yaitu nama\_obat, kemasan, dan harga. Properti \$fillable memastikan hanya kolom-kolom tersebut yang dapat diisi secara massal untuk keamanan dan konsistensi data. Selain itu, model ini juga mendefinisikan relasi one-to-many dengan model DetailPeriksa melalui fungsi detailPeriksas(), yang menunjukkan bahwa satu obat dapat terkait dengan banyak data pemeriksaan pasien.



## 9.2. Membuat ObatController

### 9.2.1. Generate Resource Controller

Gunakan perintah berikut:

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

### 9.2.2: 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\Obat;
6 use Illuminate\Http\Request;
7
8 class ObatController extends Controller
9 {
10     public function index(){
11         $obats = Obat::all();
12         return view('admin.obat.index', compact('obats'));
13     }
14
15     public function create(){
16         return view('admin.obat.create');
17     }
18
19     public function store(Request $request){
20         $request->validate([
21             'nama_obat' => 'required|string',
22             'kemasan' => 'required|string',
23             'harga' => 'required|integer',
24         ]);
25
26         Obat::create([
27             'nama_obat' => $request->nama_obat,
28             'kemasan' => $request->kemasan,
29             'harga' => $request->harga
30         ]);
31
32         return redirect()->route('obat.index')
33             ->with('message','Data Obat Berhasil dibuat')
34             ->with('type','success');
35     }
36
37     public function edit(string $id){
38         $obat = Obat::findOrFail($id);
39         // return view('admin.obat.edit', compact('obat'));
40         return view('admin.obat.edit')->with([
41             'obat'=> $obat
42         ]);
43         // sama saja seperti compact('obat') -> bedanya bisa ubah nama key
44     }
45 }
```

```

45
46     public function update(Request $request, string $id){
47         $request->validate([
48             'nama_obat' => 'required|string',
49             'kemasan' => 'nullable|string',
50             'harga' => 'required|integer',
51         ]);
52
53         $obat = Obat::findOrFail($id);
54         $obat->update([
55             'nama_obat' => $request->nama_obat,
56             'kemasan' => $request->kemasan,
57             'harga' => $request->harga
58         ]);
59
60         return redirect()->route('obat.index')
61             ->with('message','Data Obat berhasil di edit')
62             ->with('type', 'success');
63     }
64
65     public function destroy(string $id){
66         $obat = Obat::findOrFail($id);
67         $obat->delete();
68
69         return redirect()->route('obat.index')
70             ->with('message','Data Obat berhasil di Hapus')
71             ->with('type', 'success');
72     }
73 }
74

```

Link file txt (ObatController.php) :

[https://drive.google.com/file/d/1DZVS4SxHEzZpXUksWXC0H8bcNPPbtm\\_M/view?usp=sharing](https://drive.google.com/file/d/1DZVS4SxHEzZpXUksWXC0H8bcNPPbtm_M/view?usp=sharing)

### 9.2.3: Penjelasan

#### 1. Function Index

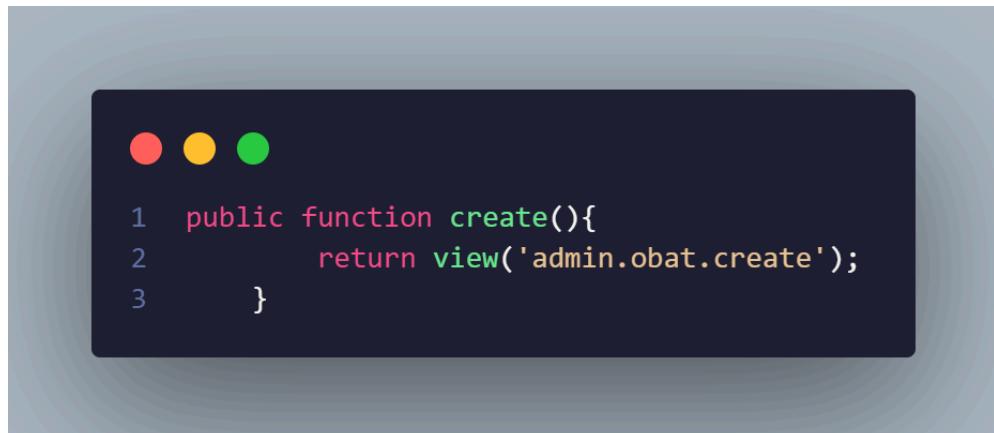
```

1  public function index(){
2      $obats = Obat::all();
3      return view('admin.obat.index', compact('obats'));
4  }

```

- **Query:** `Obat::all();`  
Mengambil semua data dari tabel `Obat`.
- **View:** Data pasien dikirim ke `admin.obat.index` menggunakan `compact('obats')`. Ini berguna untuk menampilkan tampilan menu obat.

## 2. Function Create



```
● ● ●

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

- Hanya mengarahkan ke halaman form untuk menambahkan data obat baru (`admin.obat.create`).

## 3. Function Store



```
● ● ●

1 public function store(Request $request){
2     $request->validate([
3         'nama_obat' => 'required|string',
4         'kemasan' => 'required|string',
5         'harga' => 'required|integer',
6     ]);
7
8     Obat::create([
9         'nama_obat' => $request->nama_obat,
10        'kemasan' => $request->kemasan,
11        'harga' => $request->harga
12    ]);
13
14     return redirect()->route('obat.index')
15         ->with('message', 'Data Obat Berhasil dibuat')
16         ->with('type', 'success');
17 }
```

- Validasi input: Untuk memastikan semua data sesuai ketentuan. Misalnya:
  - `nama_obat` dan `kemasan` harus terisi dan berbentuk string.
  - `harga` harus terisi dan berbentuk integer.
- Query create:  
`Obat::create([...]);`  
Menyimpan data obat ke tabel `obats`.
- Redirect:  
Kembali ke tampilan daftar obat dengan membawa key `message` dengan nilai `Data Obat Berhasil dibuat` dan key `type` dengan nilai `success`. Berguna untuk menampilkan alert.

#### 4. Function Edit

```
● ● ●

1 public function edit(string $id){
2     $obat = Obat::findOrFail($id);
3     // return view('admin.obat.edit', compact('obat'));
4     return view('admin.obat.edit')->with([
5         'obat'=> $obat
6     ]);
7     // sama saja seperti compact('obat') -> bedanya bisa ubah nama key
8 }
```

- **Query:** `Obat::findOrFail($id)`  
Mengambil data obat sesuai dengan `id` yang didapat dari route parameter.
- **View:** Data obat kemudian dikirim ke `admin.obat.edit` untuk ditampilkan pada form edit obat.

#### 5. Function Update

```
● ● ●

1 public function update(Request $request, string $id){
2     $request->validate([
3         'nama_obat' => 'required|string',
4         'kemasan' => 'nullable|string',
5         'harga' => 'required|integer',
6     ]);
7
8     $obat = Obat::findOrFail($id);
9     $obat->update([
10         'nama_obat' => $request->nama_obat,
11         'kemasan' => $request->kemasan,
12         'harga' => $request->harga
13     ]);
14
15     return redirect()->route('obat.index')
16         ->with('message', 'Data Obat berhasil di edit')
17         ->with('type', 'success');
18 }
```

- Validasi input: Untuk memastikan semua data sesuai ketentuan. Misalnya:
  - `nama_obat` harus terisi dan berbentuk string.
  - `kemasan` tidak harus terisi namun harus berbentuk string.
  - `harga` harus terisi dan berbentuk integer.
- Query Find:  
`$obat = Obat::findOrFail($id)`  
Mengambil data obat sesuai dengan `id` yang didapat dari route parameter.
- Query Update:

```
$obat->update([...]);  
Mengupdate data obat ke tabel obats.
```

- Redirect:  
Kembali ke tampilan daftar obat dengan membawa key `message` dengan nilai `Data Obat Berhasil diedit` dan key `type` dengan nilai `success`. Berguna untuk menampilkan alert.

## 6. Function Destroy



```
1 public function destroy(string $id){  
2     $obat = Obat::findOrFail($id);  
3     $obat->delete();  
4  
5     return redirect()->route('obat.index')  
6         ->with('message','Data Obat berhasil di Hapus')  
7         ->with('type','success');  
8 }
```

- **Query Find:**  
`$obat = Obat::findOrFail($id)`  
Mengambil data obat sesuai dengan `id` yang didapat dari route parameter.
- **Query Update:**  
`$obat->delete();`  
Menghapus data pasien sesuai dengan hasil dari `$obat`.
- **Redirect:**  
Kembali ke tampilan daftar obat dengan membawa key `message` dengan nilai `Data Obat Berhasil dihapus` dan key `type` dengan nilai `success`. Berguna untuk menampilkan alert.

### 9.3. Menambah Routes

Tambahkan ke dalam file `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     Route::resource('dokter', DokterController::class);
7     Route::resource('pasien', PasienController::class);
8     Route::resource('obat', ObatController::class);
9 });
});
```

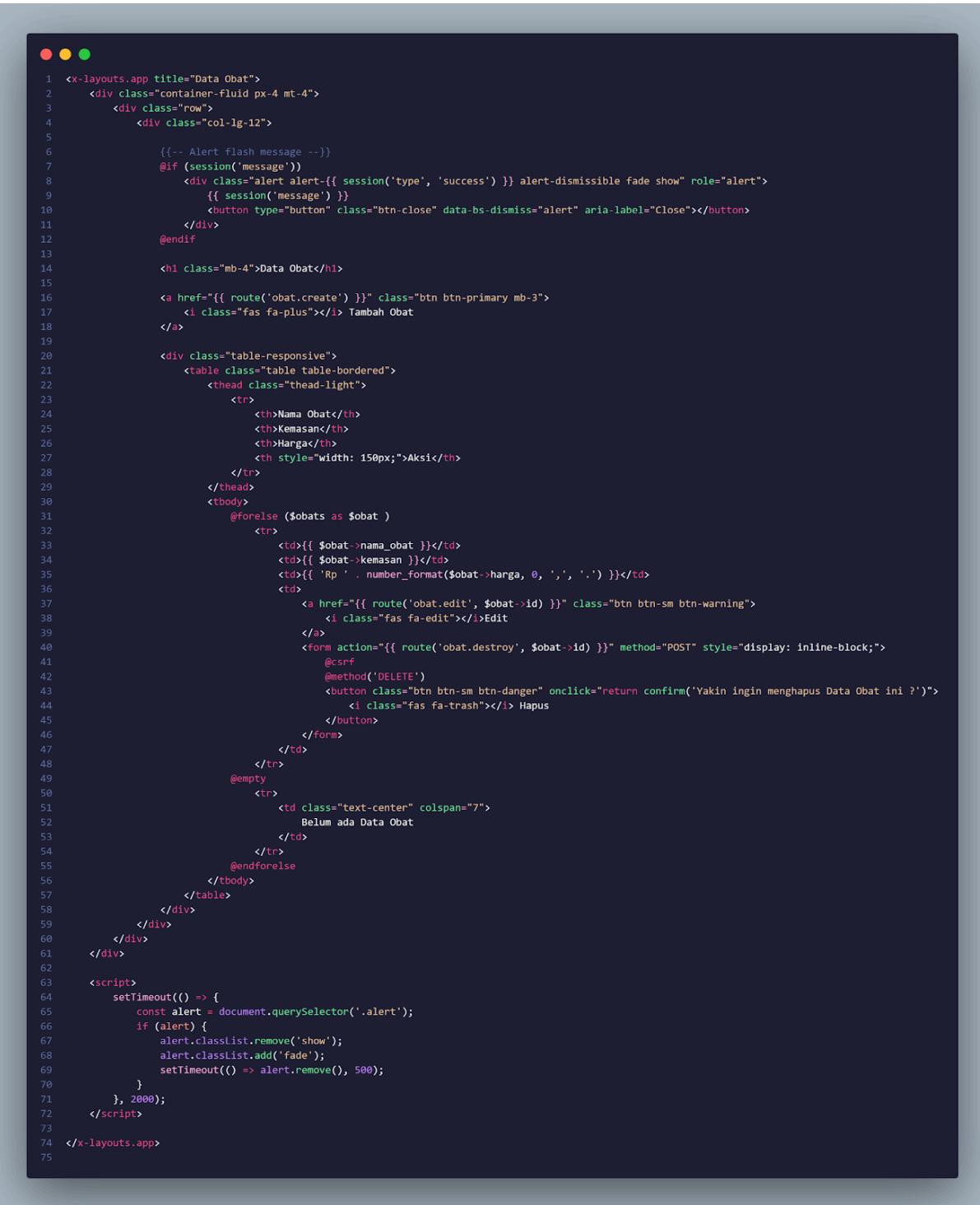
### 9.4. Menyiapkan Tampilan

Membuat struktur file sesuai dengan contoh berikut:

```
└── admin
    ├── dokter
    └── obat
        ├── create.blade.php
        ├── edit.blade.php
        └── index.blade.php
```

#### 9.4.1. obat/index.blade.php

Untuk menampilkan daftar obat. Perhatikan pada `forelse ($obats as $obat)` pada `$obats` harus sesuai dengan `obats` yang ada di `compat('obats')` di `ObatController`



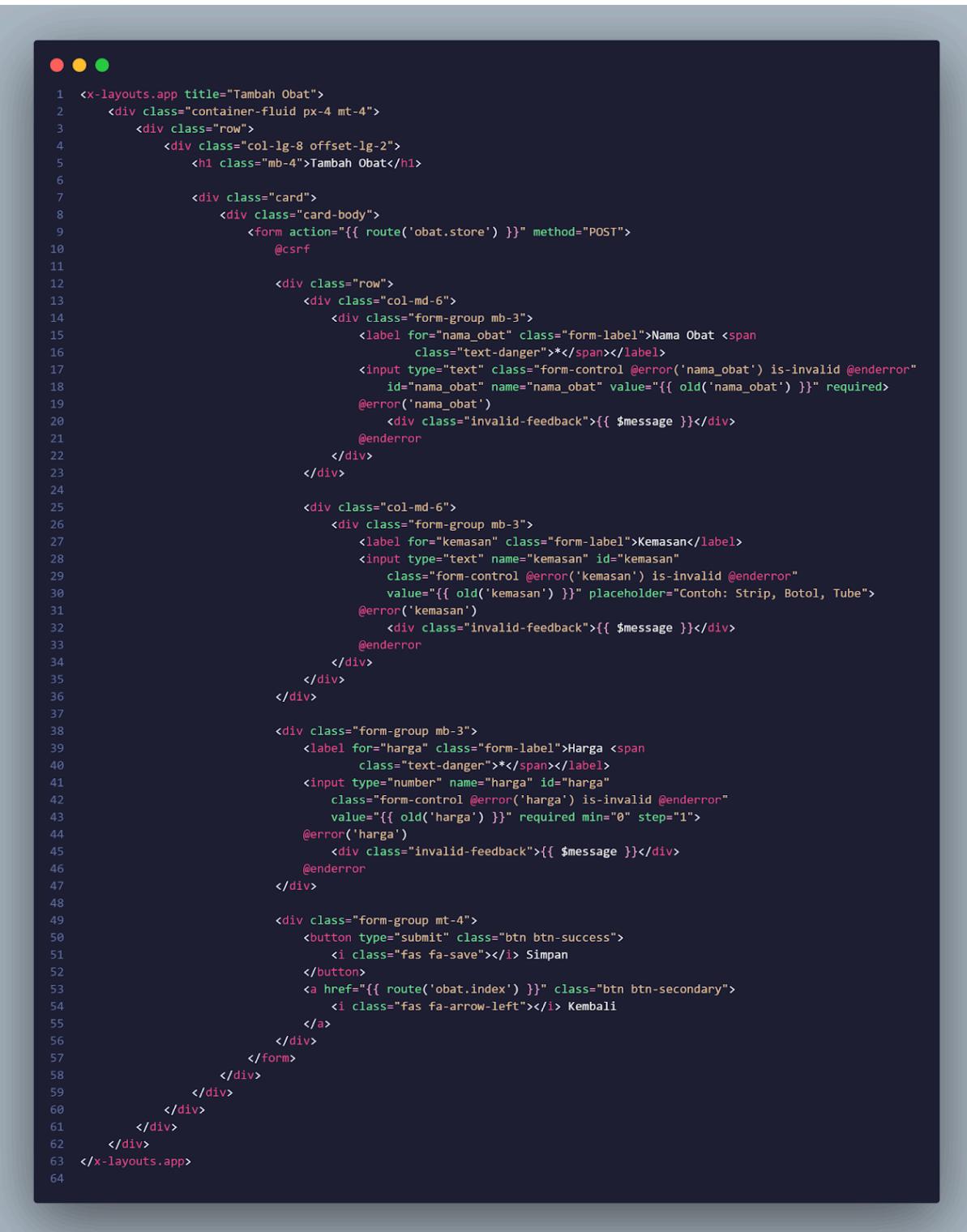
```
1 <x-layouts.app title="Data Obat">
2     <div class="container-fluid px-4 mt-4">
3         <div class="row">
4             <div class="col-lg-12">
5
6                 {{-- Alert flash message --}}
7                 @if (session('message'))
8                     <div class="alert alert-{{ session('type', '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 Obat</h1>
15
16            <a href="{{ route('obat.create') }}" class="btn btn-primary mb-3">
17                <i class="fas fa-plus"></i> Tambah Obat
18            </a>
19
20            <div class="table-responsive">
21                <table class="table table-bordered">
22                    <thead class="thead-light">
23                        <tr>
24                            <th>Nama Obat</th>
25                            <th>Kemasan</th>
26                            <th>Harga</th>
27                            <th style="width: 150px;">Aksi</th>
28                        </tr>
29                    </thead>
30                    <tbody>
31                        @forelse ($obats as $obat )
32                            <tr>
33                                <td>{{ $obat->nama_obat }}</td>
34                                <td>{{ $obat->kemasan }}</td>
35                                <td>{{ 'Rp ' . number_format($obat->harga, 0, ',', '.') }}</td>
36                                <td>
37                                    <a href="{{ route('obat.edit', $obat->id) }}" class="btn btn-sm btn-warning">
38                                        <i class="fas fa-edit"></i> Edit
39                                    </a>
40                                    <form action="{{ route('obat.destroy', $obat->id) }}" method="POST" style="display: inline-block;">
41                                        @csrf
42                                        @method('DELETE')
43                                        <button class="btn btn-sm btn-danger" onclick="return confirm('Yakin ingin menghapus Data Obat ini ?')">
44                                            <i class="fas fa-trash"></i> Hapus
45                                        </button>
46                                    </form>
47                                </td>
48                            </tr>
49                        @empty
50                            <tr>
51                                <td class="text-center" colspan="7">
52                                    Belum ada Data Obat
53                                </td>
54                            </tr>
55                        @endforelse
56                    </tbody>
57                </table>
58            </div>
59        </div>
60    </div>
61
62    <script>
63        setTimeout(() => {
64            const alert = document.querySelector('.alert');
65            if (alert) {
66                alert.classList.remove('show');
67                alert.classList.add('fade');
68                setTimeout(() => alert.remove(), 500);
69            }
70        }, 2000);
71    </script>
72
73
74 </x-layouts.app>
75
```

Link :

<https://drive.google.com/file/d/1xwlnHTYaovlUuDMHzl5M-ePaiBcp7cz/view?usp=sharing>

#### 9.4.2. obat/create.blade.php

Untuk menampilkan form tambah obat. Perhatikan khususnya pada bagian form, name harus sesuai dengan validasi yang ada di function store di `ObatController`.



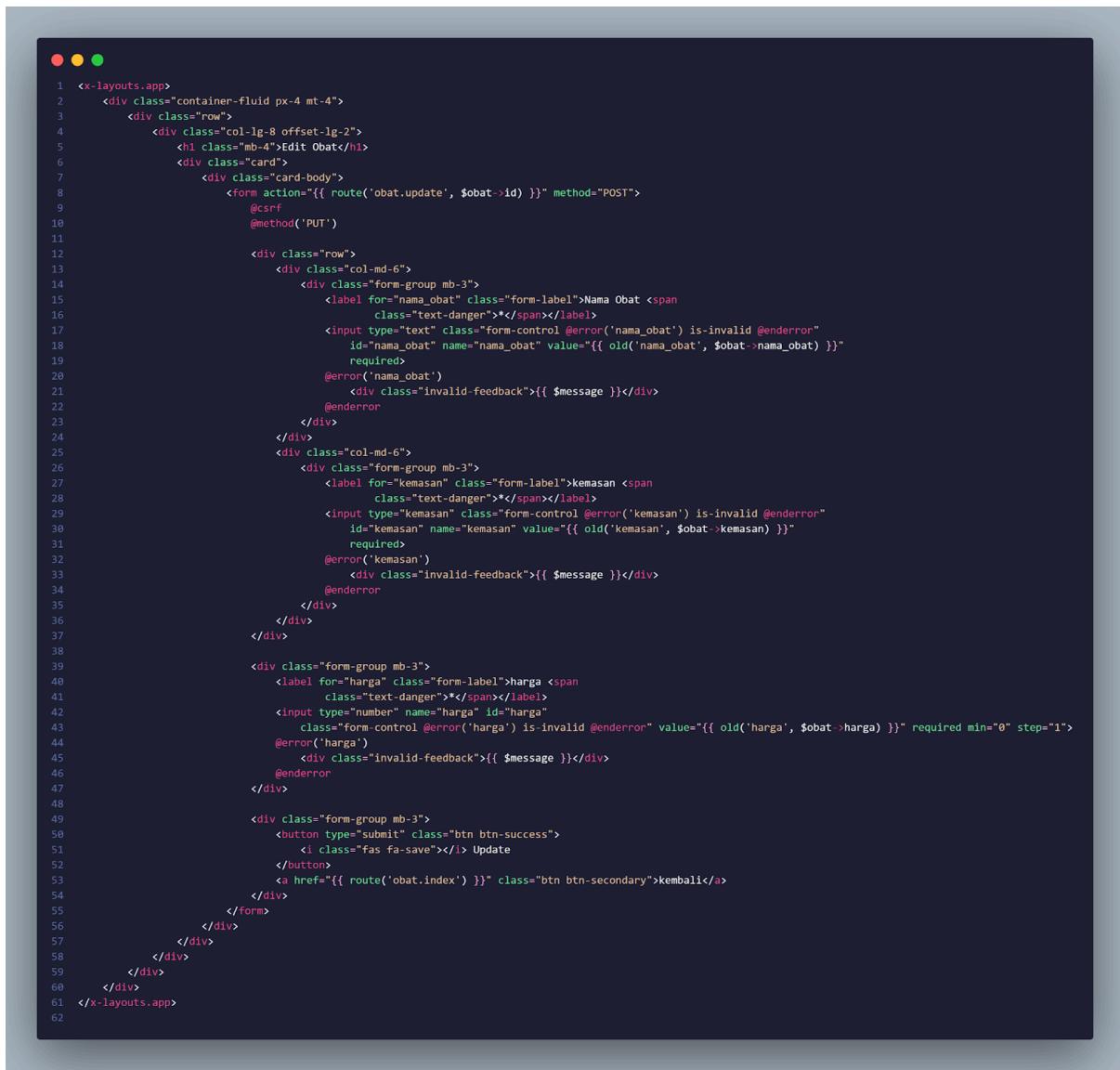
```
1 <x-layouts.app title="Tambah Obat">
2     <div classe="container-fluid px-4 mt-4">
3         <div class="row">
4             <div class="col-lg-8 offset-lg-2">
5                 <h1 class="mb-4">Tambah Obat</h1>
6
7                 <div class="card">
8                     <div class="card-body">
9                         <form action="{{ route('obat.store') }}" method="POST">
10                            @csrf
11
12                            <div class="row">
13                                <div class="col-md-6">
14                                    <div class="form-group mb-3">
15                                        <label for="nama_obat" class="form-label">Nama Obat <span
16                                            class="text-danger">*</span></label>
17                                        <input type="text" class="form-control @error('nama_obat') is-invalid @enderror"
18                                              id="nama_obat" name="nama_obat" value="{{ old('nama_obat') }}" required>
19                                         @error('nama_obat')
20                                             <div class="invalid-feedback">{{ $message }}</div>
21                                         @enderror
22                                     </div>
23                                 </div>
24
25                                 <div class="col-md-6">
26                                     <div class="form-group mb-3">
27                                         <label for="kemasan" class="form-label">Kemasan</label>
28                                         <input type="text" name="kemasan" id="kemasan"
29                                             class="form-control @error('kemasan') is-invalid @enderror"
30                                             value="{{ old('kemasan') }}" placeholder="Contoh: Strip, Botol, Tube">
31                                         @error('kemasan')
32                                             <div class="invalid-feedback">{{ $message }}</div>
33                                         @enderror
34                                     </div>
35                                 </div>
36                             </div>
37
38                             <div class="form-group mb-3">
39                                 <label for="harga" class="form-label">Harga <span
40                                     class="text-danger">*</span></label>
41                                 <input type="number" name="harga" id="harga"
42                                     class="form-control @error('harga') is-invalid @enderror"
43                                     value="{{ old('harga') }}" required min="0" step="1">
44                                 @error('harga')
45                                     <div class="invalid-feedback">{{ $message }}</div>
46                                 @enderror
47                             </div>
48
49                             <div class="form-group mt-4">
50                                 <button type="submit" class="btn btn-success">
51                                     <i class="fas fa-save"></i> Simpan
52                                 </button>
53                                 <a href="{{ route('obat.index') }}" class="btn btn-secondary">
54                                     <i class="fas fa-arrow-left"></i> Kembali
55                                 </a>
56                             </div>
57                         </form>
58                     </div>
59                 </div>
60             </div>
61         </div>
62     </div>
63 </x-layouts.app>
```

Link :

[https://drive.google.com/file/d/1bMU-kKJD\\_LL9\\_SMdZVgfsjVUws9WTBFi/view?usp=drive\\_link](https://drive.google.com/file/d/1bMU-kKJD_LL9_SMdZVgfsjVUws9WTBFi/view?usp=drive_link)

## 9.4.2. obat/edit.blade.php

Untuk menampilkan form edit data obat. Cukup mirip dengan create baik name dan bentuk input.



```
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">Edit Obat</h1>
6         <div class="card">
7           <div class="card-body">
8             <form action="{{ route('obat.update', $obat->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_obat" class="form-label">Nama Obat <span
16                      class="text-danger">*</span></label>
17                    <input type="text" class="form-control @error('nama_obat') is-invalid @enderror"
18                      id="nama_obat" name="nama_obat" value="{{ old('nama_obat', $obat->nama_obat) }}"
19                      required>
20                    @error('nama_obat')
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="kemasan" class="form-label">kemasan <span
28                      class="text-danger">*</span></label>
29                    <input type="kemasan" class="form-control @error('kemasan') is-invalid @enderror"
30                      id="kemasan" name="kemasan" value="{{ old('kemasan', $obat->kemasan) }}"
31                      required>
32                    @error('kemasan')
33                      <div class="invalid-feedback">{{ $message }}</div>
34                    @enderror
35                  </div>
36                </div>
37              </div>
38
39              <div class="form-group mb-3">
40                <label for="harga" class="form-label">harga <span
41                  class="text-danger">*</span></label>
42                <input type="number" name="harga" id="harga"
43                  class="form-control @error('harga') is-invalid @enderror" value="{{ old('harga', $obat->harga) }}"
44                  required min="0" step="1">
45                @error('harga')
46                  <div class="invalid-feedback">{{ $message }}</div>
47                @enderror
48              </div>
49
50              <div class="form-group mb-3">
51                <button type="submit" class="btn btn-success">
52                  <i class="fas fa-save"></i> Update
53                </button>
54                <a href="{{ route('obat.index') }}" class="btn btn-secondary">kembali</a>
55              </div>
56            </div>
57          </div>
58        </div>
59      </div>
60    </x-layouts.app>
61
62
```

Link:

[https://drive.google.com/file/d/1BXJDHXPiUxpU8HZjTPsdbqwlEJ-3G1hb/view?usp=drive\\_link](https://drive.google.com/file/d/1BXJDHXPiUxpU8HZjTPsdbqwlEJ-3G1hb/view?usp=drive_link)

## 9.5. Menambahkan Sidebar Navigasi

Tambahkan menu baru untuk **Manajemen Obat** ke dalam sidebar layout. tempatkan setelah **Manajemen Pasien**



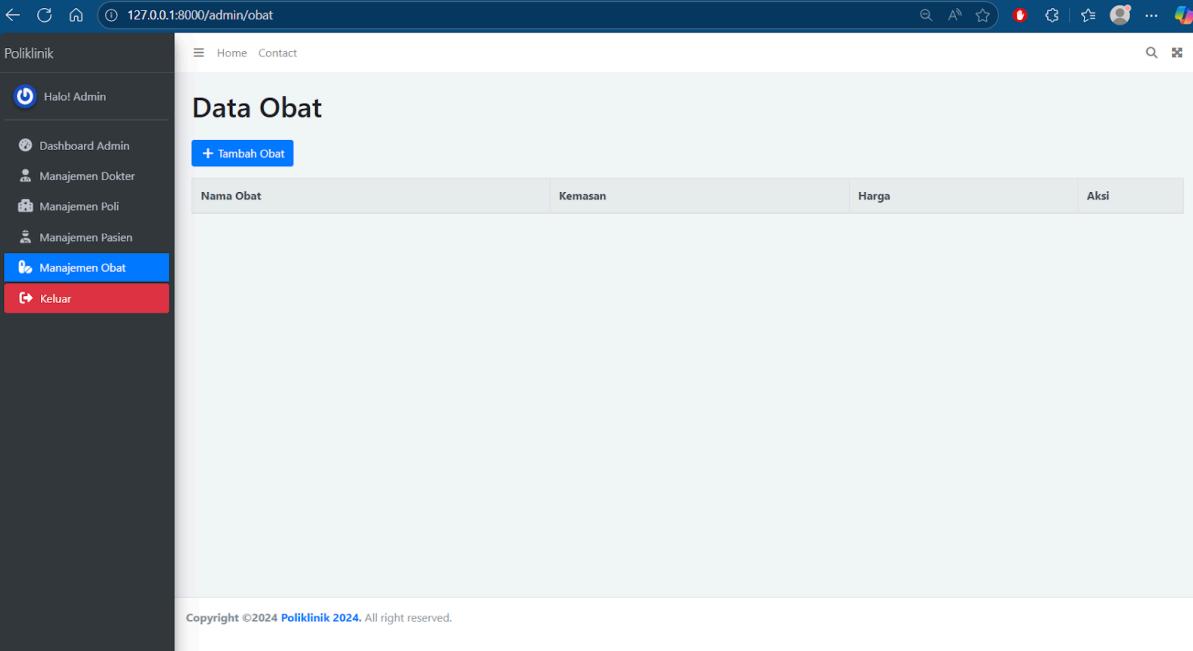
```
1 <li class="nav-item">
2   <a href="#">{} route('obat.index')>
3     class="nav-link {{ request()->routeIs('obat.*') ? 'active' : '' }}"
4       <i class="nav-icon fas fa-pills"></i>
5       <p>
6         Manajemen Obat
7       </p>
8     </a>
9   </li>
```

## 9.6. Testing

### Login Sebagai Admin

- Email: [admin@gmail.com](mailto:admin@gmail.com)
- Password: **admin**

### Hasil



The screenshot shows a web application interface. On the left is a dark sidebar with a navigation menu. The 'Manajemen Obat' item is highlighted with a blue background. The main content area is titled 'Data Obat' and contains a table with four columns: 'Nama Obat', 'Kemasan', 'Harga', and 'Aksi'. Above the table is a blue button labeled '+ Tambah Obat'. At the bottom of the page, there is a copyright notice: 'Copyright ©2024 Poliklinik 2024. All right reserved.'

← ⏪ ⏹ ⏺ 127.0.0.1:8000/admin/obat/create

Halo! Admin

Dashboard Admin

Manajemen Dokter

Manajemen Poli

Manajemen Pasien

**Manajemen Obat**

Keluar

☰ Home Contact

Tambah Obat

Nama Obat \*

Kemasan

Contoh: Strip, Botol, Tube

Harga \*

Simpan Kembali

Copyright ©2024 Poliklinik 2024. All right reserved.

The screenshot shows a web application interface for managing a medical clinic. On the left is a dark sidebar with various administrative links. The main area is titled 'Tambah Obat' (Add Medicine) and contains three input fields: 'Nama Obat' (Medicine Name), 'Kemasan' (Packaging), and 'Harga' (Price). Below these fields are two buttons: 'Simpan' (Save) and 'Kembali' (Back). The 'Manajemen Obat' link in the sidebar is highlighted with a blue background, indicating it is the active section. The footer of the page includes a copyright notice for 'Poliklinik 2024'.