

# Modul 13 : Dokter Memeriksa Pasien

---

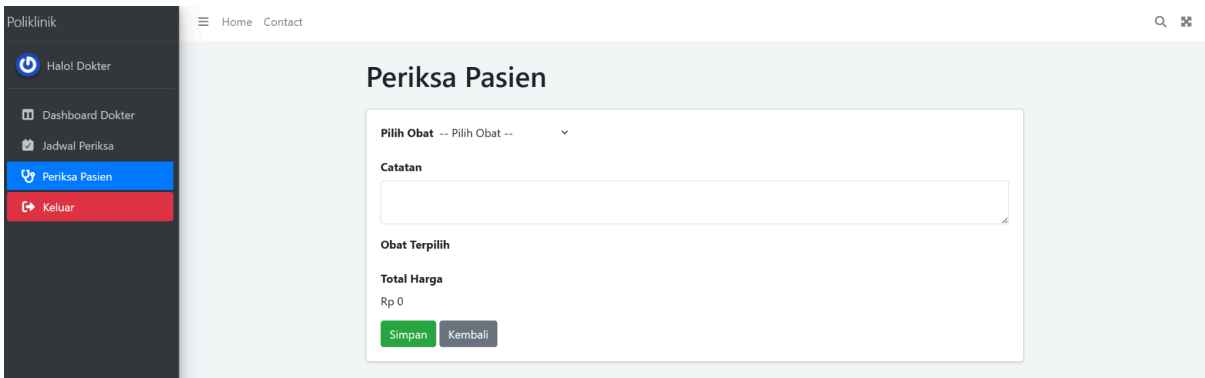
## Tujuan Pembelajaran

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

- Memahami alur bisnis proses pemeriksaan pasien oleh dokter
  - Mampu mengimplementasikan form input untuk diagnosa, keluhan, dan biaya dengan Laravel
  - Mampu mengelola relasi antar model untuk menambahkan catatan obat pada pemeriksaan pasien
  - Mampu menampilkan daftar pasien terdaftar beserta detail pemeriksaan secara informatif
  - Memahami alur bisnis proses melihat riwayat pemeriksaan pasien oleh dokter.
  - Mampu mengimplementasikan relasi antar model dengan Eloquent ORM Laravel.
  - Mampu menampilkan data dengan eager loading untuk optimasi query.
  - Mampu membuat tampilan detail yang informatif untuk riwayat pemeriksaan.
- 

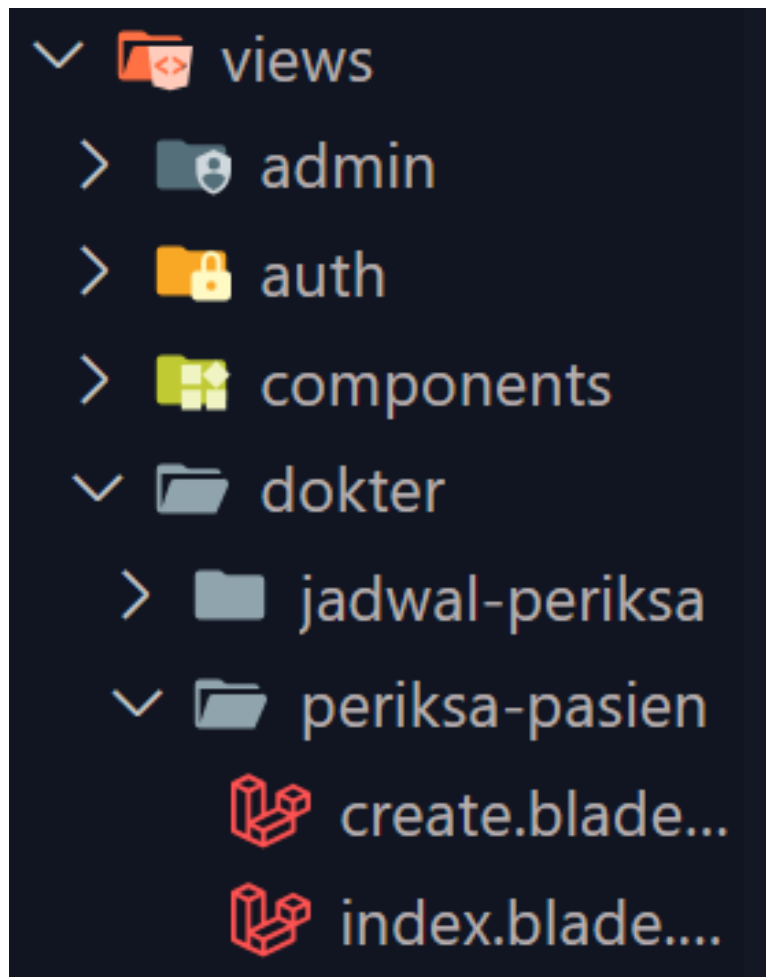
## 13.0 Overview

Mahasiswa diperkenalkan dengan tampilan form pemeriksaan pasien menggunakan Laravel Blade. View ini menampilkan daftar obat yang dapat dipilih dokter melalui dropdown, form catatan untuk mencatat diagnosa atau keterangan tambahan, serta daftar obat yang telah dipilih beserta total biaya. Obat yang dipilih akan ditambahkan secara dinamis ke list menggunakan JavaScript, dan total harga akan dihitung otomatis. Data obat yang dipilih juga disimpan dalam format JSON pada input tersembunyi agar dapat dikirim ke backend bersama informasi biaya pemeriksaan. View ini mendemonstrasikan konsep form input dinamis, relasi antar model (obat-pemeriksaan), serta integrasi frontend-backend dalam Laravel.



The screenshot shows a web application interface for a clinic. On the left is a dark sidebar with the title 'Poliklinik' and navigation links: 'Haloi Dokter', 'Dashboard Dokter', 'Jadwal Periksa', 'Periksa Pasien' (highlighted in blue), and 'Keluar' (highlighted in red). The main content area has a light blue header with 'Home' and 'Contact' links, and a search icon. The title 'Periksa Pasien' is centered. Below it is a form with the following elements: a dropdown menu labeled 'Pilih Obat -- Pilih Obat --', a text area labeled 'Catatan', a section titled 'Obat Terpilih' which is currently empty, a 'Total Harga' label followed by 'Rp 0', and two buttons at the bottom: 'Simpan' (green) and 'Kembali' (grey).

### 13.1 Tampilan Form & Daftar Pemeriksaan Pasien



Buat file di dalam  
`resources/views/dokter/periksa-pasien/create.blade.php`

```

1 <x-layouts.app title="Periksa Pasien">
2     {{!-- ALERT FLASH MESSAGE --}}
3     <div class="container-fluid px-4 mt-4">
4         <div class="row">
5             <div class="col-lg-8 offset-lg-2">
6                 <h1 class="mb-4">Periksa Pasien</h1>
7
8                 <div class="card">
9                     <div class="card-body">
10                         <form action="{{ route('periksa-pasien.store') }}" method="POST">
11                             @csrf
12
13                             <input type="hidden" name="id_daftar_poli" value="{{ $id }}">
14
15                             <div class="form-group mb-3">
16                                 <label for="obat" class="form-label">Pilih Obat</label>
17                                 <select id="select-obat" class="form-select">
18                                     <option value="">-- Pilih Obat --</option>
19                                     @foreach ($obats as $obat)
20                                         <option value="{{ $obat->id }}" data-nama="{{ $obat->nama_obat }}"
21                                             data-harga="{{ $obat->harga }}">
22                                             {{ $obat->nama_obat }} - Rp{{ number_format($obat->harga) }}
23                                         </option>
24                                     @endforeach
25                                 </select>
26                             </div>
27
28                             <div class="form-group mb-3">
29                                 <label for="catatan" class="form-label">Catatan</label>
30                                 <textarea name="catatan" id="catatan" class="form-control">{{ old('catatan') }}</textarea>
31                             </div>
32
33                             <div class="form-group mb-3">
34                                 <label>Obat Terpilih</label>
35                                 <ul id="obat-terpilih" class="list-group mb-2"></ul>
36                                 <input type="hidden" name="biaya_periksa" id="biaya_periksa" value="0">
37                                 <input type="hidden" name="obat_json" id="obat_json">
38                             </div>
39
40                             <div class="form-group mb-3">
41                                 <label>Total Harga</label>
42                                 <p id="total-harga" class="fw-bold">Rp 0</p>
43                             </div>
44
45                             <button type="submit" class="btn btn-success">Simpan</button>
46                             <a href="{{ route('periksa-pasien.index') }}" class="btn btn-secondary">Kembali</a>
47                         </form>
48                     </div>
49                 </div>
50             </div>
51         </div>
52     </div>
53 </x-layouts.app>

```

\*Sambungan dari code diatas

```
1  <script>
2    const selectObat = document.getElementById('select-obat');
3    const listObat = document.getElementById('obat-terpilih');
4    const inputBiaya = document.getElementById('biaya_periksa');
5    const inputObatJson = document.getElementById('obat_json');
6    const totalHargaEl = document.getElementById('total-harga');
7
8    let daftarObat = [];
9
10   selectObat.addEventListener('change', () => {
11     const selectedOption = selectObat.options[selectObat.selectedIndex];
12     const id = selectedOption.value;
13     const nama = selectedOption.dataset.nama;
14     const harga = parseInt(selectedOption.dataset.harga || 0);
15
16     if (!id || daftarObat.some(o => o.id == id)) {
17       return;
18     }
19
20     daftarObat.push({
21       id,
22       nama,
23       harga
24     });
25     renderObat();
26     selectObat.selectedIndex = 0;
27   });
28
29   function renderObat() {
30     listObat.innerHTML = '';
31     let total = 0;
32
33     daftarObat.forEach((obat, index) => {
34       total += obat.harga;
35
36       const item = document.createElement('li');
37       item.className = 'list-group-item d-flex justify-content-between align-items-center';
38       item.innerHTML = `
39         ${obat.nama} - Rp ${obat.harga.toLocaleString()}
40         <button type="button" class="btn btn-sm btn-danger" onclick="hapusObat(${index})">Hapus</button>
41       `;
42       listObat.appendChild(item);
43     });
44
45     inputBiaya.value = total;
46     totalHargaEl.textContent = `Rp ${total.toLocaleString()}`;
47     inputObatJson.value = JSON.stringify(daftarObat.map(o => o.id));
48   }
49
50   function hapusObat(index) {
51     daftarObat.splice(index, 1);
52     renderObat();
53   }
54 </script>
55
```

link drive : [create](#)

Kemudian buat halaman daftar pasien yang telah di periksa.

Buat file di dalam `resources/views/dokter/periksa-pasien/index.blade.php`

```

1 <x-layouts.app title="Periksa Pasien">
2
3 <div class="container-fluid px-4 mt-4">
4 <div class="row">
5 <div class="col-lg-12">
6
7 <h1 class="mb-4">Periksa Pasien</h1>
8
9 {{-- ALERT FLASH MESSAGE --}}
10 @if (session('message'))
11 <div class="alert alert-{{ session('type') ?? 'success' }} alert-dismissible fade show"
12 role="alert">
13 <strong>{{ session('type') == 'danger' ? 'Error!' : 'Berhasil!' }}</strong>
14 {{ session('message') }}
15 <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
16 </div>
17 @endif
18
19 <div class="table-responsive">
20 <table class="table table-bordered">
21 <thead class="thead-light">
22 <tr>
23 <th>ID</th>
24 <th>Pasien</th>
25 <th>Keluhan</th>
26 <th>No Antrian</th>
27 <th style="width: 150px;">Aksi</th>
28 </tr>
29 </thead>
30 <tbody>
31 @forelse ($daftarPasien as $dp)
32 <tr>
33 <td>{{ $loop->iteration }}</td>
34 <td>{{ $dp->pasien->nama }}</td>
35 <td>{{ $dp->keluhan }}</td>
36 <td>{{ $dp->no_antrian }}</td>
37 <td>
38 @if ($dp->periksas->isEmpty())
39 <span class="badge bg-success">Sudah Diperiksa</span>
40 @else
41 <a href="{{ route('periksa-pasien.create', $dp->id) }}"
42 class="btn btn-sm btn-warning">
43 <i class="fas fa-edit"></i> Periksa
44 </a>
45 @endif
46 </td>
47 </tr>
48 @empty
49 <tr>
50 <td colspan="5" class="text-center">Tidak ada data pasien periksa.</td>
51 </tr>
52 @endforelse
53 </tbody>
54 </table>
55 </div>
56
57 </div>
58 </div>
59 </div>
60
61 <script>
62 setTimeout(() => {
63 const alert = document.querySelector('.alert');
64 if (alert) {
65 alert.classList.remove('show');
66 alert.classList.add('fade');
67 setTimeout(() => alert.remove(), 200);
68 }
69 }, 2000);
70 </script>
71
72 </x-layouts.app>
73

```

link drive : [index](#)

---

## 13.2 Menambahkan List Periksa Pasien di Sidebar

Pergi ke halaman `resources/views/components/partials/sidebar.blade.php`  
Di bawah list Jadwal Periksa di tambahkan list Periksa Pasien



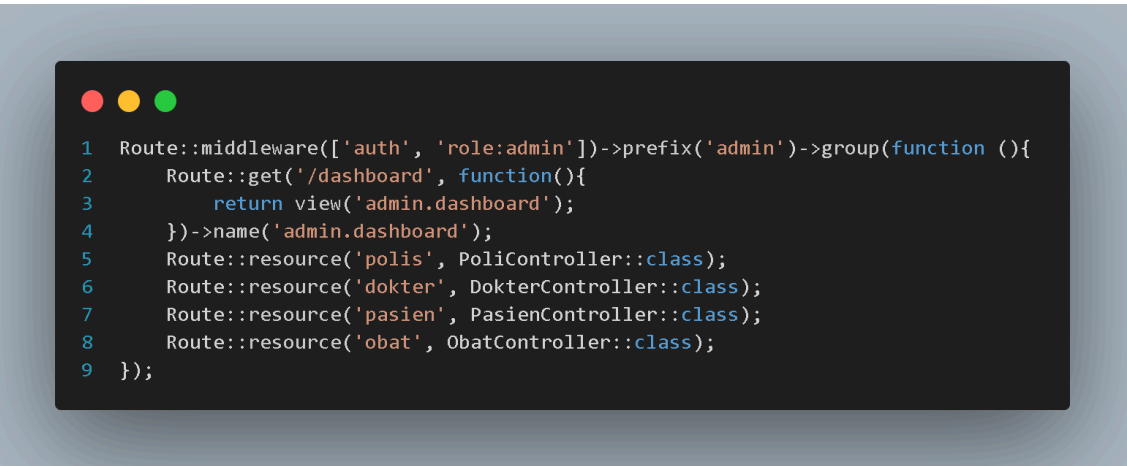
```
1 <li class="nav-item">
2   <a href="{{ route('periksa-pasien.index') }}"
3     class="nav-link {{ request()->routeIs('periksa-pasien.*') ? 'active' : '' }}">
4     <i class="nav-icon fas fa-stethoscope"></i>
5     <p>
6       Periksa Pasien
7     </p>
8   </a>
9 </li>
```

---

## 13.3 Menambahkan Route Periksa Pasien

Beralih ke halaman `routes/web.php`. kita tambahkan route periksa pasien setelah Jadwal Periksa yang ada di dalam middleware role dokter.

`routes/components/web.php`

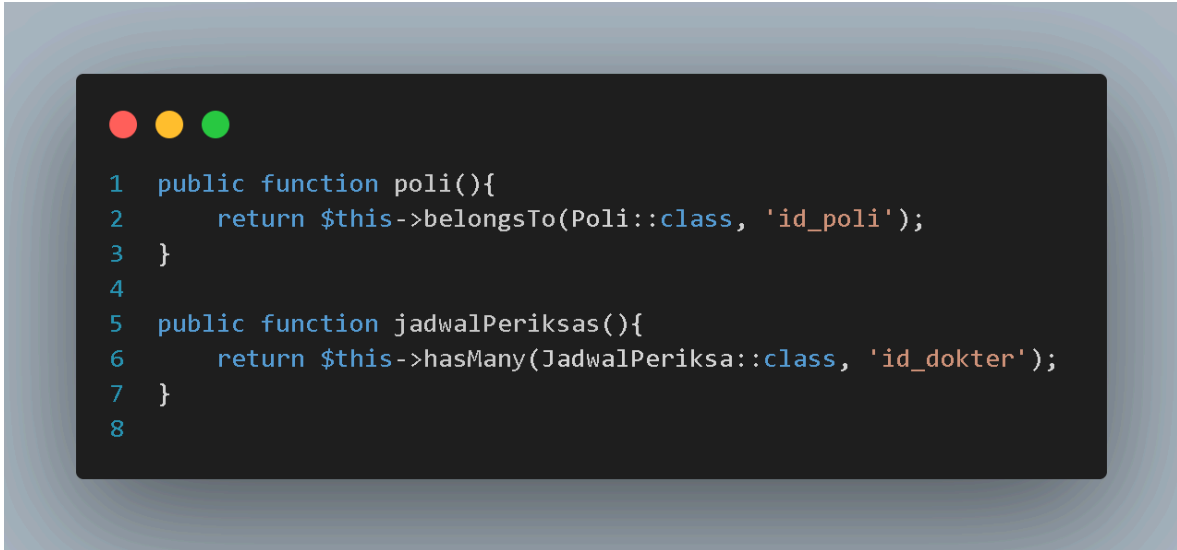


```
1 Route::middleware(['auth', 'role:admin'])->prefix('admin')->group(function (){
2   Route::get('/dashboard', function(){
3     return view('admin.dashboard');
4   })->name('admin.dashboard');
5   Route::resource('polis', PoliController::class);
6   Route::resource('dokter', DokterController::class);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

## 13.4 Modifikasi Model User

Sekarang beralih ke halaman `app/Models/User.php`. kemudian cek lagi di bagian relasi harus ada tambahan seperti ini di bawah function casts:

`app/Models/User.php`



```
1 public function poli(){
2     return $this->belongsTo(Poli::class, 'id_poli');
3 }
4
5 public function jadwalPeriksas(){
6     return $this->hasMany(JadwalPeriksa::class, 'id_dokter');
7 }
8
```

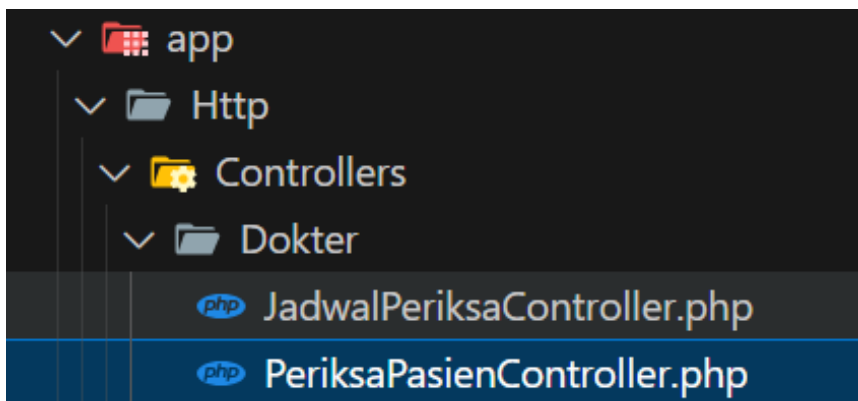
---

## 13.5 Membuat Controller Periksa Pasien

### 13.5.1: Generate Resource Controller

Gunakan perintah berikut:

```
[TERMINAL]
php artisan make:controller Dokter/PeriksaPasienController
```



## 13.5.2: Isi Method pada Controller

Method yang digunakan:

1. `index()`  
→ Menampilkan daftar pasien yang akan diperiksa oleh dokter yang sedang login.
2. `create($id)`  
→ Menampilkan form untuk melakukan pemeriksaan terhadap pasien berdasarkan ID pendaftaran.
3. `store(Request $request)`  
→ Menyimpan data hasil pemeriksaan dan obat yang diberikan ke database. isi lengkap method seperti `index`, `create`, dan `store` untuk menangani proses pemeriksaan pasien oleh dokter.

```
1  <?php
2
3  namespace App\Http\Controllers\dokter;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\DaftarPoli;
7  use App\Models\DetailPeriksa;
8  use App\Models\Obat;
9  use App\Models\Periksa;
10 use Illuminate\Http\Request;
11 use Illuminate\Support\Facades\Auth;
12
13 class PeriksaPasienController extends Controller
14 {
15     public function index()
16     {
17         $dokterId = Auth::id();
18
19         $daftarPasien = DaftarPoli::with(['pasien', 'jadwalPeriksa', 'periksas'])
20             ->whereHas('jadwalPeriksa', function ($query) use ($dokterId) {
21                 $query->where('id_dokter', $dokterId);
22             })
23             ->orderBy('no_antrian')
24             ->get();
25
26         return view('dokter.periksa-pasien.index', compact('daftarPasien'));
27     }
28
29     public function create($id)
30     {
31         $obats = Obat::all();
32         return view('dokter.periksa-pasien.create', compact('obats', 'id'));
33     }
34
35     public function store(Request $request)
36     {
37         $request->validate([
38             'obat_json' => 'required',
39             'catatan' => 'nullable|string',
40             'biaya_periksa' => 'required|integer',
41         ]);
42
43         $obatIds = json_decode($request->obat_json, true);
44
45         $periksa = Periksa::create([
46             'id_daftar_poli' => $request->id_daftar_poli,
47             'tgl_periksa' => now(),
48             'catatan' => $request->catatan,
49             'biaya_periksa' => $request->biaya_periksa + 150000,
50         ]);
51
52         foreach ($obatIds as $idObat) {
53             DetailPeriksa::create([
54                 'id_periksa' => $periksa->id,
55                 'id_obat' => $idObat,
56             ]);
57         }
58
59         return redirect()->route('periksa-pasien.index')->with('success', 'Data periksa berhasil disimpan.');
```



### 13.5.3: Penjelasan

#### 1. Function Index

Fungsi `index` mengambil ID dokter yang sedang login dengan `Auth::id()`, lalu mengambil data pasien dari model `DaftarPoli` beserta relasi yang dibutuhkan, yaitu `pasien`, `jadwalPeriksa`, dan `periksas`. Query menggunakan `whereHas` untuk memfilter pasien yang dijadwalkan ke dokter yang bersangkutan, serta mengurutkan hasil berdasarkan `no_antrian`. Data yang diperoleh kemudian dikirim ke view `dokter.periksa-pasien.index` untuk ditampilkan dalam bentuk daftar pasien yang siap diperiksa.

```
1 public function index()
2 {
3     $dokterId = Auth::id();
4
5     $daftarPasien = DaftarPoli::with(['pasien', 'jadwalPeriksa', 'periksas'])
6         ->whereHas('jadwalPeriksa', function ($query) use ($dokterId) {
7             $query->where('id_dokter', $dokterId);
8         })
9         ->orderBy('no_antrian')
10        ->get();
11
12    return view('dokter.periksa-pasien.index', compact('daftarPasien'));
13 }
```

#### 2. Function Create

Fungsi `create` pada controller ini bertujuan untuk menampilkan form pemeriksaan pasien. Parameter `$id` digunakan untuk mengetahui pasien atau daftar poli yang akan diperiksa. Fungsi ini mengambil semua data obat dari model `Obat` menggunakan `Obat::all()` agar dokter bisa memilih obat yang akan diresepkan. Selanjutnya, data obat dan ID pasien dikirim ke view `dokter.periksa-pasien.create` menggunakan `compact`, sehingga form pemeriksaan dapat menampilkan daftar obat yang tersedia dan mengaitkan pemeriksaan dengan pasien yang sesuai.

```
1 public function create($id)
2 {
3     $obats = Obat::all();
4     return view('dokter.periksa-pasien.create', compact('obats', 'id'));
5 }
```

### 3. Function Store

Fungsi `store` berfungsi untuk menyimpan hasil pemeriksaan pasien ke database. Pertama, data yang dikirim melalui form divalidasi, termasuk daftar obat (`obat_json`), catatan (`catatan`), dan biaya pemeriksaan (`biaya_periksa`). Data obat dalam format JSON di-decode menjadi array untuk diproses.

Selanjutnya, dibuat record baru di tabel `Periksa` dengan mengisi `id_daftar_poli`, `tgl_periksa` saat ini, catatan dokter, dan total biaya pemeriksaan ditambah biaya administrasi sebesar Rp150.000. Setelah itu, setiap obat yang dipilih akan disimpan di tabel `DetailPeriksa` sebagai relasi antara pemeriksaan dan obat yang diresepkan.

Terakhir, fungsi ini melakukan redirect ke halaman daftar pemeriksaan pasien dengan pesan sukses, sehingga dokter dapat melihat daftar pasien yang telah diperiksa.

```
1 public function store(Request $request)
2 {
3     $request->validate([
4         'obat_json' => 'required',
5         'catatan' => 'nullable|string',
6         'biaya_periksa' => 'required|integer',
7     ]);
8
9     $obatIds = json_decode($request->obat_json, true);
10
11     $periksa = Periksa::create([
12         'id_daftar_poli' => $request->id_daftar_poli,
13         'tgl_periksa' => now(),
14         'catatan' => $request->catatan,
15         'biaya_periksa' => $request->biaya_periksa + 150000,
16     ]);
17
18     foreach ($obatIds as $idObat) {
19         DetailPeriksa::create([
20             'id_periksa' => $periksa->id,
21             'id_obat' => $idObat,
22         ]);
23     }
24
25     return redirect()->route('periksa-pasien.index')->with('success', 'Data pemeriksaan berhasil disimpan.');
```

## 13.6 Testing

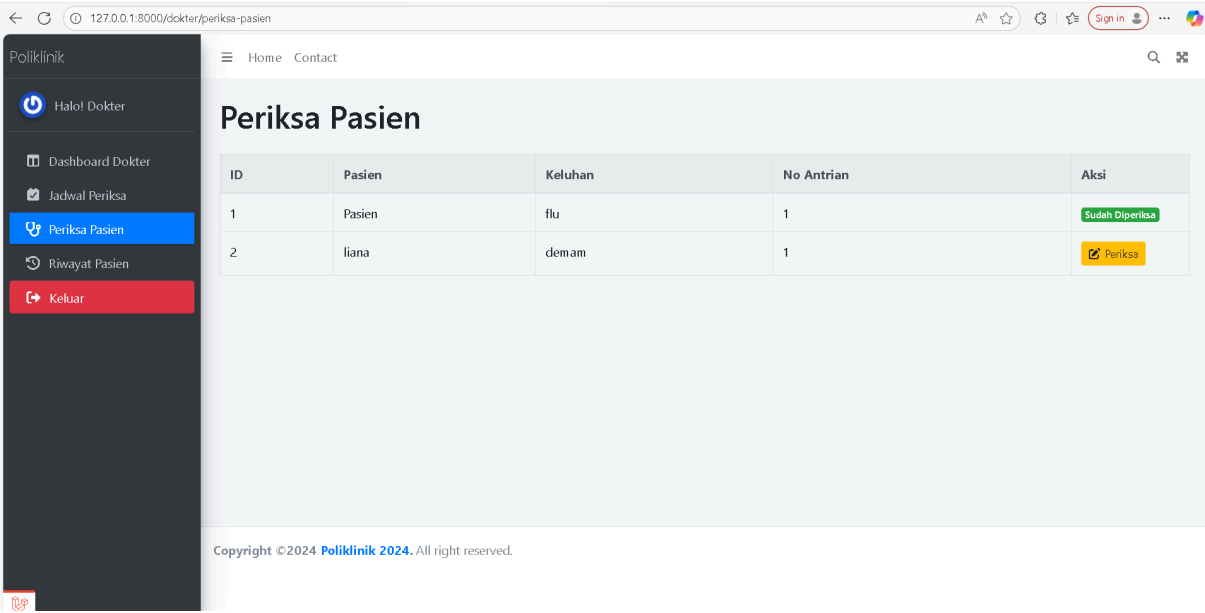
Tahap terakhir adalah mencoba semua fitur.

### 13.6.1 Login sebagai dokter

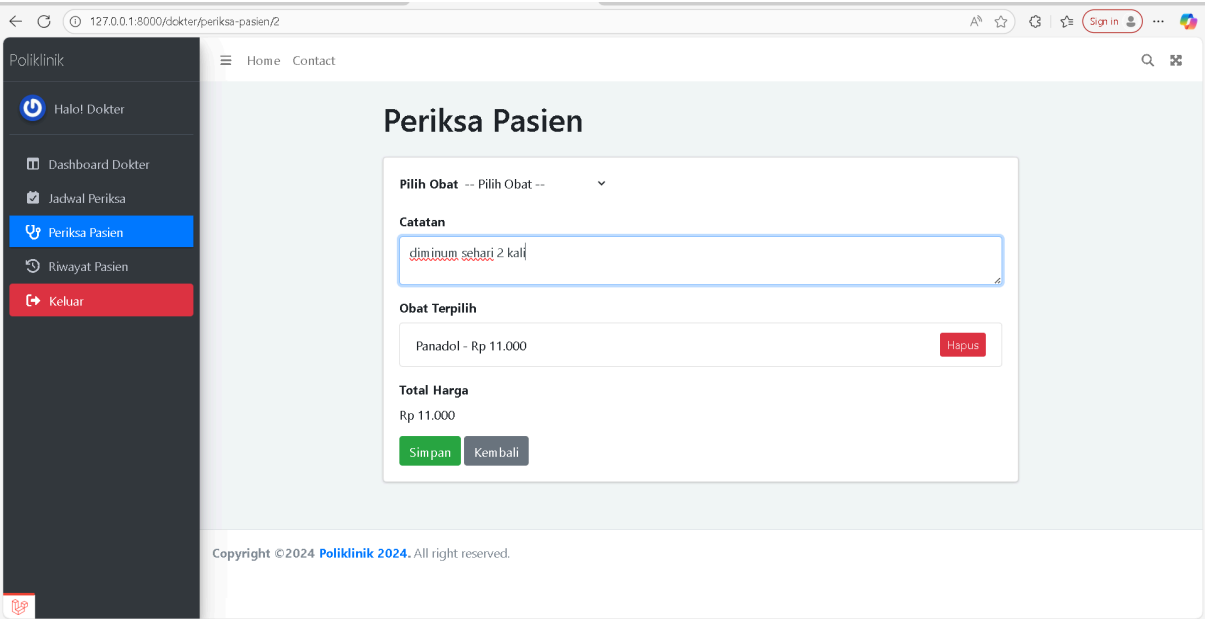
Email: [dokter@gmail.com](mailto:dokter@gmail.com)

Password: dokter

Tampilan halaman Periksa Piwayat



Tampilan ketika dokter akan memeriksa pasien yang sudah mendaftar



# Modul 14: Riwayat Pasien untuk Dokter

## 14.0 Overview

Pada modul ini, kita akan membuat sebuah fitur yang memungkinkan dokter untuk melihat riwayat pemeriksaan pasien yang pernah diperiksa sebelumnya. Dokter dapat melihat daftar semua pasien yang telah diperiksa, kemudian dapat melihat detail pemeriksaan termasuk **catatan dokter, obat yang diresepkan, dan total biaya pemeriksaan**. Kurang lebih seperti ini hasil akhirnya:

### Halaman Index:

The screenshot displays the 'Riwayat Pasien' (Patient History) page. On the left is a dark sidebar with navigation links: 'Halof Dokter', 'Dashboard Dokter', 'Jadwal Periksa', 'Periksa Pasien', 'Riwayat Pasien', and 'Keluar'. The main content area has a light blue header with 'Riwayat Pasien' and a search icon. Below the header is a table with the following data:

No Antrian	Nama Pasien	Keluhan	Tanggal Periksa	Biaya	Aksi
1	al	sakit	22/09/2025	Rp 243.000	<a href="#">Detail</a>

At the bottom of the page, there is a copyright notice: 'Copyright ©2024 Poliklinik 2024. All right reserved.'

- Tabel daftar riwayat pemeriksaan dengan kolom: **No Antrian, Nama Pasien, Keluhan, Tanggal Periksa, Biaya, dan Aksi Detail.**
- **Button detail** untuk melihat informasi lengkap pemeriksaan

## Halaman Detail:

#	Nama Obat	Harga
1	parasetamol	Rp 16.000
2	Tremenza	Rp 67.000
3	Alpara	Rp 10.000

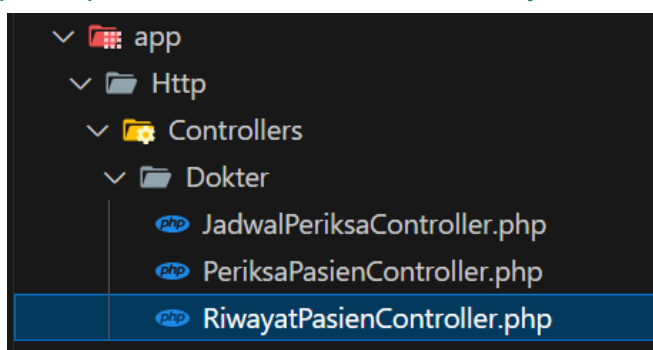
- **Informasi lengkap pasien** (nama, email, no antrian, keluhan, poli, dokter)
- **Detail pemeriksaan** (tanggal, biaya, catatan dokter)
- **Daftar obat yang diresepkan**
- **Total biaya pemeriksaan**

Kita akan membuat controller yang memiliki 2 fungsi, satu untuk **menampilkan daftar riwayat pemeriksaan**, satunya lagi untuk menampilkan **detail pemeriksaan**. Kemudian kita buat **route untuk memanggil controller** tersebut ketika kita mengakses url yang kita inginkan. Setelah itu kita buat **view (tampilan)** dari halaman seperti pada contoh di awal. Jangan lupa tambahkan button atau shortcut di bagian sidebar agar tidak perlu menuliskan url secara manual.

## 14.1 Membuat Controller Baru

### 14.1.1: Generate Resource Controller

`app/http/Controllers/dokter/RiwayatPasienController.php`



Untuk membuat file controller bernama `RiwayatPasienController.php` di dalam folder dokter, kita bisa menggunakan perintah

[TERMINAL]

```
php artisan make:controller dokter/RiwayatPasienController
```

Di dalam file ini kita akan membuat fungsi untuk menampilkan daftar riwayat pemeriksaan dan fungsi untuk menampilkan detail pemeriksaan.

### 14.1.2: Isi Method pada Controller

Isi lengkap method seperti `index`, dan `show` untuk menangani operasi CRUD.

```
1  <?php
2
3  namespace App\Http\Controllers\Dokter;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\Periksa;
7  use Illuminate\Http\Request;
8
9  class RiwayatPasienController extends Controller
10 {
11     public function index(){
12         $riwayatPasien = Periksa::with([
13             'daftarPoli.pasien',
14             'daftarPoli.jadwalPeriksa.dokter',
15             'detailPeriksas.obat'
16         ])->orderBy('tgl_periksa','desc')->get();
17
18         return view('riwayat-pasien.index', compact('riwayatPasien'));
19     }
20
21     public function show($id){
22         $periksa = Periksa::with([
23             'daftarPoli.pasien',
24             'daftarPoli.jadwalPeriksa.dokter.poli',
25             'detailPeriksas.obat'
26         ])->findOrFail($id);
27
28         return view('riwayat-pasien.show', compact('periksa'));
29     }
30 }
31
```

### 14.1.3: Penjelasan

#### 1. Function Index

```
1 public function index(){
2     $riwayatPasien = Periksa::with([
3         'daftarPoli.pasien',
4         'daftarPoli.jadwalPeriksa.dokter',
5         'detailPeriksas.obat'
6     ])->orderBy('tgl_periksa','desc')->get();
7
8     return view('riwayat-pasien.index', compact('riwayatPasien'));
9 }
10
```

- **Query:** `Periksa::with(['daftarPoli.pasien', 'daftarPoli.jadwalPeriksa.dokter', 'detailPeriksas.obat'])->orderBy('tgl_periksa','desc')->get();`

Mengambil semua data dari model `Periksa` dan melakukan *eager loading* (load semua relasi model sekaligus) pada relasi-relasi berikut:

- `daftarPoli.pasien`: Mengambil data pasien yang terkait dengan setiap pemeriksaan pada daftar poli.
- `daftarPoli.jadwalPeriksa.dokter`: Mengambil data dokter yang terkait dengan jadwal pemeriksaan pada setiap poli.
- `detailPeriksas.obat`: Mengambil data obat-obat yang diberikan pada setiap pemeriksaan pasien.

Kemudian, hasil query diurutkan berdasarkan tanggal pemeriksaan (`tgl_periksa`) secara menurun (`desc`), dan data tersebut diambil menggunakan `get()`.

- **View:** Data yang telah diperoleh dari query tersebut akan dikirim ke tampilan `dokter.riwayat-pasien.index` menggunakan fungsi `compact('riwayatPasien')`, yang membuat variabel `$riwayatPasien` tersedia di dalam tampilan tersebut.

## 2. Function Show

```
1 public function show($id){
2     $periksa = Periksa::with([
3         'daftarPoli.pasien',
4         'daftarPoli.jadwalPeriksa.dokter.poli',
5         'detailPeriksas.obat'
6     ]->findOrFail($id);
7
8     return view('riwayat-pasien.show', compact('periksa'));
9 }
```

### - Penjelasan:

- Mengambil satu data dari tabel **Periksa** berdasarkan **id** yang diberikan.
- Eager loading digunakan untuk memuat relasi-relasi seperti diatas.
- Fungsi **findOrFail(\$id)** digunakan untuk mencari data berdasarkan **id**. Jika data tidak ditemukan, maka aplikasi akan menampilkan error 404.

- **View:** Data yang diperoleh kemudian dikirim ke tampilan **dokter.riwayat-pasien.show** dengan variabel **\$periksa** yang dikirim menggunakan **compact('periksa')**.

## 14.2 Menambahkan Route

Pada **routes/web.php**, tambahkan code seperti ini:

```
1 Route::middleware(['auth', 'role:dokter'])->prefix('dokter')->group( function(){
2     Route::get('/dashboard', function(){
3         return view('dokter.dashboard');
4     })->name('dokter.dashboard');
5     Route::resource('jadwal-periksa', DokterJadwalPeriksaController::class);
6     Route::get('/periksa-pasien', [PeriksaPasienController::class, 'index'])->name('periksa-pasien.index');
7     Route::post('/periksa-pasien', [PeriksaPasienController::class, 'store'])->name('periksa-pasien.store');
8     Route::get('/periksa-pasien/{id}', [PeriksaPasienController::class, 'create'])->name('periksa-pasien.create');
9     Route::get('/riwayat-pasien', [RiwayatPasienController::class, 'index'])->name('dokter.riwayat-pasien.index');
10    Route::get('/riwayat-pasien/{id}', [RiwayatPasienController::class, 'show'])->name('dokter.riwayat-pasien.show');
11 });
```

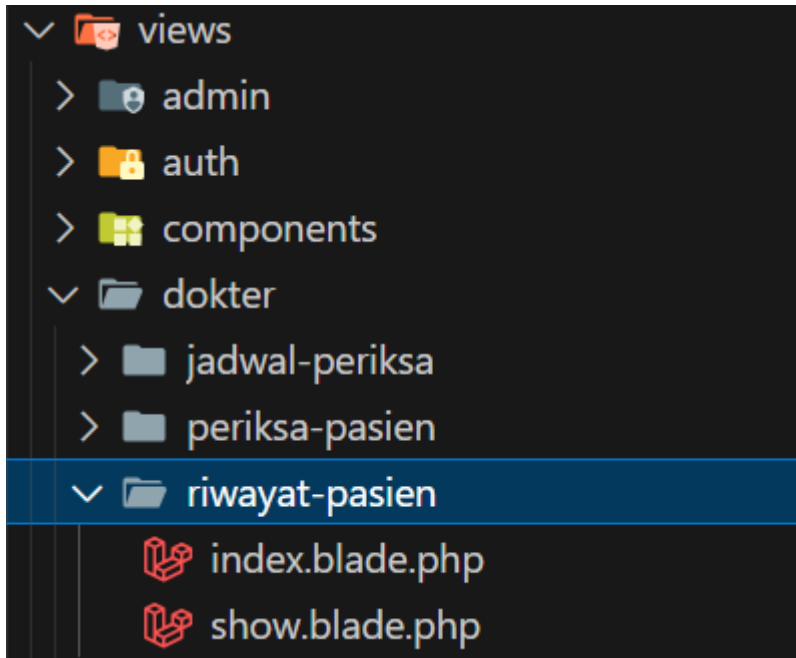
Kita tambahkan use statement untuk **RiwayatPasienController** yang ada di dalam folder dokter. Di dalam prefix dokter, kita tambahkan 2 route baru. Route get pertama memiliki url **/riwayat-pasien** yang akan memanggil fungsi **index** pada **RiwayatPasienController**. Route get kedua akan memanggil fungsi **show** dengan parameter ID pemeriksaan.



---

## 14.3 Membuat View Detail

### 14.3.1: Generate Resource View



## 14.3.2: View Index Riwayat Pasien

views/dokter/riwayat-pasien/index.blade.php

```
1 <x-layouts.app title="Riwayat Pasien">
2   <div class="container-fluid px-4 mt-4">
3     <div class="row">
4       <div class="col-12">
5         <h1 class="mb-4">Riwayat Pasien</h1>
6
7         <div class="card">
8           <div class="card-body">
9             <div class="table-responsive">
10              <table class="table table-striped">
11                <thead>
12                  <tr>
13                    <th>No Antrian</th>
14                    <th>Nama Pasien</th>
15                    <th>Keluhan</th>
16                    <th>Tanggal Periksa</th>
17                    <th>Biaya</th>
18                    <th style="width: 150px;">Aksi</th>
19                  </tr>
20                </thead>
21                <tbody>
22                  @forelse($riwayatPasien as $riwayat)
23                    <tr>
24                      <td>{{ $riwayat->daftarPoli->no_antrian }}</td>
25                      <td>{{ $riwayat->daftarPoli->pasien->name }}</td>
26                      <td>{{ $riwayat->daftarPoli->keluhan }}</td>
27                      <td>{{ \Carbon\Carbon::parse($riwayat->tgl_periksa)->format('d/m/Y') }}</td>
28                      <td>Rp {{ number_format($riwayat->biaya_periksa, 0, ',', '.') }}</td>
29                      <td>
30                        <a href="{{ route('riwayat-pasien.show', $riwayat) }}"
31                          class="btn btn-info btn-sm">
32                          <i class="fas fa-eye"></i> Detail
33                        </a>
34                      </td>
35                    </tr>
36                  @empty
37                    <tr>
38                      <td colspan="6" class="text-center">Belum ada riwayat pemeriksaan</td>
39                    </tr>
40                  @endforelse
41                </tbody>
42              </table>
43            </div>
44          </div>
45        </div>
46      </div>
47    </div>
48  </x-layouts.app>
```

link drive : [index](#)

## 14.3.3: View Detail Riwayat Pasien

views/dokter/riwayat-pasien/show.blade.php

```
1 <x-layouts.app title="Detail Riwayat Pasien">
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="d-flex justify-content-between align-items-center mb-4">
6           <h2>Detail Riwayat</h2>
7           <a href="{{ route('riwayat-pasien.index') }}" class="btn btn-secondary">
8             <i class="fas fa-arrow-left"></i> Kembali
9           </a>
10        </div>
11      </div>
12    <div class="card">
13      <h5 class="card-header">Informasi Pasien</h5>
14      <div class="card-body">
15        <p><strong>Nama Pasien:</strong> {{ $periksa->daftarPoli->pasien->nama }}</p>
16        <p><strong>No. Antrian:</strong> {{ $periksa->daftarPoli->no_antrian }}</p>
17        <p><strong>Keluhan:</strong> {{ $periksa->daftarPoli->keluhan }}</p>
18        <p><strong>Poli:</strong> {{ $periksa->daftarPoli->jadwalPeriksa->dokter->poli->nama_poli }}</p>
19        <p><strong>Dokter:</strong> {{ $periksa->daftarPoli->jadwalPeriksa->dokter->nama }}</p>
20        <p><strong>Tanggal Periksa:</strong> {{ \Carbon\Carbon::parse($periksa->tgl_periksa)->format('d/m/Y H:i') }}</p>
21      </div>
22    </div>
23
24    <div class="card mb-3">
25      <h5 class="card-header">Catatan Dokter</h5>
26      <div class="card-body">
27        <p>{{ $periksa->catatan ?: 'Tidak ada catatan' }}</p>
28      </div>
29    </div>
30
31    <div class="card mb-3">
32      <h5 class="card-header">Obat yang Diresepkan</h5>
33      <div class="card-body">
34        @if($periksa->detailPerikhas && $periksa->detailPerikhas->count() > 0)
35          <table class="table table-sm">
36            <thead>
37              <tr>
38                <th>#</th>
39                <th>Nama Obat</th>
40                <th>Harga</th>
41              </tr>
42            </thead>
43            <tbody>
44              @foreach($periksa->detailPerikhas as $index => $detail)
45                <tr>
46                  <td>{{ $index + 1 }}</td>
47                  <td>{{ $detail->obat->nama_obat }}</td>
48                  <td>Rp {{ number_format($detail->obat->harga, 0, ',', '.') }}</td>
49                </tr>
50              @endforeach
51            </tbody>
52          </table>
53        @else
54          <p class="text-muted">Tidak ada obat yang diresepkan</p>
55        @endif
56      </div>
57    </div>
58
59    <div class="card">
60      <div class="card-body text-center">
61        <h5 class="card-title">Total Biaya Periksa</h5>
62        <h3 class="text-primary">Rp {{ number_format($periksa->biaya_periksa, 0, ',', '.') }}</h3>
63      </div>
64    </div>
65  </div>
66 </div>
67 </div>
68 </x-layouts.app>
69
```

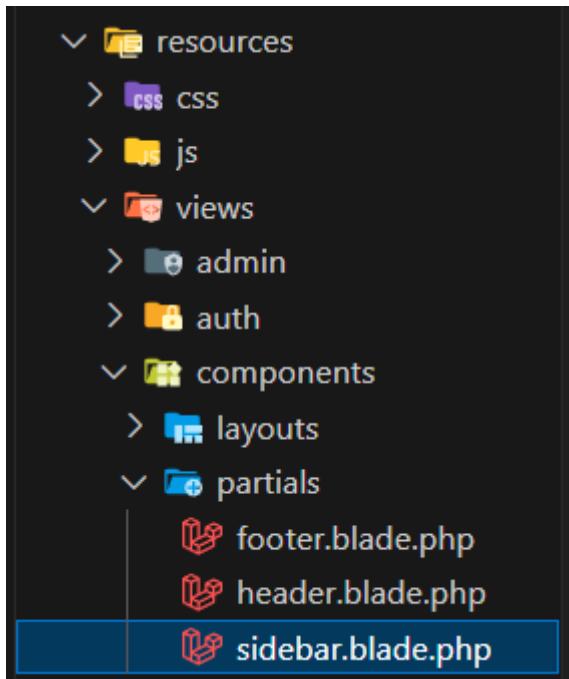
link drive : [show](#)



## 14.4 Menambahkan Shortcut di Sidebar

Tambahkan shortcut pada

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



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

Sesuaikan dengan code yang ada sebelumnya, di dalam sidebar menu, di bagian role dokter kita tambahkan shortcut berupa link yang mengarah ke route `dokter.riwayat-pasien.index`. Route ini akan mengarahkan kita ke view index yang baru saja kita buat. Sehingga untuk berpindah ke halaman riwayat pasien kita cukup menekan link yang ada di sidebar tersebut. Menggunakan **wildcard** \* pada routes agar menu tetap aktif ketika berada di halaman detail.

---



## **14.5 Testing Fitur**

Tahap terakhir adalah mencoba semua fitur.

### **14.5.1 Login Sebagai Dokter**

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

Password: dokter

### **14.5.2 Uji Semua Fitur**

- Index View
- Show Detail Riwayat Pasien