

**Laporan Praktikum
Pemrograman Web 2
Pertemuan 3
Routing, View dan Blade Laravel**



Nama Mahasiswa: Tegar Raditya Hikmawan
NIM:24/535872/SV/24321
Kelas: B2
Dosen Pengampu: Dinar Nugroho Pratomo, S.Kom., M.IM., M.Cs.

**Sekolah Vokasi
Universitas Gadjah Mada
Yogyakarta
2025**

Daftar Isi

Bab 1. Pendahuluan	2
1.1 Latar Belakang	2
1.2 Tujuan Praktikum	2
Bab 2. Dasar Teori	3
2.1 Pengenalan Laravel	3
2.2 Routing	4
2.3 View	4
2.4 Blade	4
Bab 3. Hasil dan Pembahasan	6
3.1 Potongan Kode	6
3.2 Hasil Kode	10
3.3 Penjelasan	11
Bab 4. Kesimpulan dan Saran	12
4.1 Kesimpulan	12

Bab 1. Pendahuluan

1.1 Latar Belakang

Pemrograman web saat ini menjadi salah satu keterampilan penting dalam bidang teknologi informasi. Aplikasi berbasis web semakin banyak digunakan, sehingga diperlukan kemampuan untuk membangun aplikasi yang tidak hanya berjalan dengan baik, tetapi juga mudah dikembangkan dan dikelola.

Laravel adalah salah satu framework PHP yang banyak digunakan karena menyediakan berbagai fitur yang memudahkan pengembangan, seperti sistem routing, template Blade, dan penge-lolaan database dengan Eloquent. Dengan menggunakan Laravel, pengembang dapat membuat aplikasi web secara lebih cepat, terstruktur, dan aman.

Praktikum ini bertujuan untuk memberikan pemahaman kepada mahasiswa mengenai pemrograman web lanjutan menggunakan Laravel. Melalui latihan langsung, mahasiswa akan belajar membangun aplikasi web dinamis dengan konsep-konsep penting seperti autentikasi, manajemen data, dan penerapan pola MVC. Dengan demikian, mahasiswa diharapkan memiliki dasar yang kuat untuk mengembangkan aplikasi web modern.

1.2 Tujuan Praktikum

1. Mahasiswa mampu mengenali Routing, View dan Blade Laravel
2. Mahasiswa mengetahui cara membuat Routing, View dan Blade Laravel
3. Mahasiswa dapat menggunakan / menerapkan Routing, View dan Blade Laravel pada project

Bab 2. Dasar Teori

2.1 Pengenalan Laravel

Laravel adalah salah satu framework PHP paling populer dan banyak digunakan di dunia pengembangan web. Dibuat oleh Taylor Otwell, Laravel dirancang untuk membuat proses pengembangan aplikasi web menjadi lebih mudah, lebih cepat, dan lebih menyenangkan. Dengan pendekatan yang elegan dan syntax yang mudah dibaca, Laravel menyediakan alat dan fitur yang kuat untuk membantu pembangunan aplikasi dari yang sederhana hingga kompleks.[1]

Keunggulan Laravel antara lain[1]:

- Eloquent ORM : Eloquent menyederhanakan interaksi dengan database dengan cara yang lebih intuitif dan object-oriented
- Migrations : Kemudahan dalam mengelola struktur database aplikasi
- Blade Template Engines : Fitur untuk menyaring HTTP request yang masuk ke aplikasi
- Middleware : Fitur untuk menyaring HTTP request yang masuk ke aplikasi
- Routing yang mudah : Sistem routing yang sederhana namun fleksibel
- Autentikasi : Sistem otentikasi yang lengkap dan mudah dikonfigurasi
- Laravel Mix : Tool untuk mengelola asset dan melakukan compiling

Arsitektur aplikasi Laravel dirancang untuk menjadi intuitif dan fleksibel[1]. Salah satu prinsip dan pola desain software yang diadopsi Laravel adalah Model View Controller [1]. Model View Controller atau yang dapat disingkat MVC adalah sebuah pola arsitektur dalam membuat sebuah aplikasi dengan cara memisahkan kode menjadi tiga bagian yang terdiri dari[5]:

- Model

Bagian yang bertugas untuk menyiapkan, mengatur, memanipulasi, dan mengorganisasikan data yang ada di database.

- View

Bagian yang bertugas untuk menampilkan informasi dalam bentuk Graphical User Interface (GUI).

- Controller

Bagian yang bertugas untuk menghubungkan serta mengatur model dan view agar dapat saling terhubung.

2.2 Routing

Routing adalah proses pengiriman data maupun informasi ke pengguna melalui sebuah permintaan yang dilakukan kepada alamat yang sudah terdaftar, lalu alamat tersebut akan memproses dari permintaan kita tadi. Setelah proses selesai maka akan mengembalikan sebuah output atau hasil dari proses tersebut. [2]

Ketika project laravel dibuat, akan dibuat juga sebuah folder bernama routes. Di dalam folder tersebut terdapat 4 file yang memiliki fungsi masing-masing, antara lain yaitu [2] :

- api.php : Digunakan untuk membuat routing API. Di dalam file ini kita juga dapat membuat core service API dengan menggunakan Laravel.
- channels.php : File ini digunakan untuk membuat routing yang bersifat broadcasting event, seperti notification.
- console.php : File ini digunakan untuk membuat routing command yang berjalan di terminal. Jadi kita juga bisa membuat perintah artisan kita sendiri.
- web.php : File ini digunakan untuk membuat routing web biasa.

2.3 View

Seperti yang sudah disebutkan sebelumnya, huruf "V" pada MVC memiliki arti "Views". Views atau "the presentation logic" berfungsi untuk menampilkan data yang dibuat oleh Controller (the application logic). Views dibuat dengan susunan kode HTML yang akan ditampilkan. Untuk menggunakan views menggunakan global function view(). [3]

Views disimpan pada folder **resources/views**. Untuk membuat views kita menggunakan template engine yang disediakan oleh Laravel yaitu Blade dengan ekstensi **.blade.php**. Kita juga bisa mengirim data ke views, baik langsung melalui routes atau controller. Caranya antara lain adalah dengan menggunakan data array, method with(), dan compact(). [3]

2.4 Blade

Blade merupakan salah satu fitur dari Laravel yang sangat bermanfaat dalam proses pengembangan halaman web. Blade templating biasanya digunakan untuk memudahkan membuat tampilan pada halaman web. Dengan menggunakan Blade Templating kita tidak perlu menulis kode berulang kali untuk membuat sebuah tampilan yang berulang seperti header, navbar, dan footer. [4]

Setelah kita masuk pada folder views, disana kita bisa membuat file **.blade.php** kita. Salah satu fitur yang disediakan dan paling sering digunakan adalah blade directive, digunakan untuk meringkas penulisan kode PHP hanya dengan "@".

Dengan blade directive kita bisa melakukan layouting dengan mudah menggunakan pewarisian template atau template inheritance. Kita melakukannya menggunakan fitur antara lain "@extends()", "@yield()", dan "@section()". "@extends()" berfungsi untuk mengambil template dari file lain, biasanya layout. Dalam layout, jika kita ingin mengganti isi menjadi berbeda setiap halaman, kita bisa menggunakan "@yield()". Lalu untuk memasukkan isi yang berbeda,

kita tinggal menggantinya pada "@section()". Selain ketiga itu, juga terdapat satu lagi yaitu "@include()". Berbeda dengan ketiga sebelumnya, yang bisa dibilang satu paket, "@include()" bisa bekerja sendiri, dimana kita bisa memasukkan file blade lain, dan menaruhnya tanpa perlu pewarisan, dimana saja. [4]

Bab 3. Hasil dan Pembahasan

Pada praktikum kali ini, diperintahkan untuk membuat sebuah halaman website yang dimana akan menampilkan hasil dari eksplorasi fitur routing dan view. Sekaligus untuk mengimplementasikan fitur templating Laravel yaitu blade.

3.1 Potongan Kode

web.php:

```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5 Route::get('/', function () {
6     return view('welcome');
7 });
8
9 Route::get('/see/{id}', function ($id) {
10    return view('see', [
11        'id' => $id
12    ]);
13});
```

layouts/master.blade.php:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width,
6         initial-scale=1.0">
7     <meta http-equiv="X-UA-Compatible" content="ie=edge">
8     <title>@yield('title')</title>
9
10    <link rel="stylesheet" href="https://cdn.jsdelivr.net/
11        npm/bootstrap@5.3.8/dist/css/bootstrap.min.css">
12    <link rel="stylesheet" href="{{ asset('css/style.css') }}>
13</head>
14 <body>
15     <div class="container-fluid p-0">
16         @include('layouts.navbar')
```

```

15
16      <main class="container-md mt-5">
17          @yield('content')
18      </main>
19  </div>
20 </body>
21 </html>

```

layouts/navbar.blade.php:

```

1 <nav class="navbar navbar-expand-lg bg-special">
2     <div class="container-fluid">
3         <a class="navbar-brand" href="#">@yield('title')</
4             a>
5         <button class="navbar-toggler" type="button" data-
6             bs-toggle="collapse" data-bs-target="#
7                 navbarSupportedContent" aria-controls="
8                 navbarSupportedContent" aria-expanded="false"
9                 aria-label="Toggle navigation">
10            <span class="navbar-toggler-icon"></span>
11        </button>
12        <div class="collapse navbar-collapse" id="
13            navbarSupportedContent">
14            <ul class="navbar-nav me-auto mb-2 mb-lg-0">
15                <li class="nav-item">
16                    <a class="nav-link active" aria-
17                        current="page" href="/">Home</a>
18                </li>
19                <li class="nav-item">
20                    <a class="nav-link" href="#">Link</a>
21                </li>
22                <li class="nav-item dropdown">
23                    <a class="nav-link dropdown-toggle"
24                        href="#" role="button" data-bs-
25                        toggle="dropdown" aria-expanded="
26                            false">
27                        Dropdown
28                    </a>
29                    <ul class="dropdown-menu">
30                        <li><a class="dropdown-item" href=
31                            "#">Action</a></li>
32                        <li><a class="dropdown-item" href=
33                            "#">Another action</a></li>
34                        <li><hr class="dropdown-divider"
35                            ></li>
36                        <li><a class="dropdown-item" href=
37                            "#">Something else here</a></li
38                            >
39                    </ul>
40                </li>
41                <li class="nav-item">

```

```

27             <a class="nav-link disabled" aria-
28                 disabled="true">Disabled</a>
29         </li>
30     </ul>
31     <form class="d-flex" role="search">
32         <input class="form-control me-2" type="
33             search" placeholder="Search" aria-label
34                 ="Search"/>
35         <button class="btn btn-outline-success"
36             type="submit">Search</button>
37     </form>
38   </div>
39 </div>
40 </nav>

```

welcome.blade.php:

```

1 @extends("layouts.master")
2
3 @section('title')
4 Welcome
5 @endsection
6
7 @section('content')
8 <div class="">
9     <h1 style="color: rgb(95, 160, 127)">List gak tau apa
10    </h1>
11    <ol class="list-group list-group-flush">
12        @for ($i = 1; $i<=5; $i++)
13            <li class="list-group-item bg-transparent">
14                <a href="/see/{{ $i }}>Link {{ $i }}</a>
15            </li>
16        @endfor
17    </ol>
18 </div>
19 @endsection

```

see.blade.php:

```

1 @extends("layouts.master")
2
3 @section('title')
4 See {{ $id }}
5 @endsection
6
7 @section('content')
8 <div class="">
9     <h1>Nomor {{ $id }}</h1>
10    <center>
11        <img id="item{{ $id }}" alt="Random image">
12    </center>
13 </div>

```

```

14
15 <script>
16   window.currentId = @json($id);
17 </script>
18 <script src="{{ asset('js/script.js') }}"></script>
19 @endsection

```

style.css:

```

1 body{
2   background-color: rgb(183, 44, 144) ;
3   padding: 0;
4   margin: 0;
5 }
6
7 a{
8   text-decoration: none;
9   color: rgb(95, 160, 127);
10 }
11
12 .bg-special{
13   background-color: rgb(95, 160, 127);
14 }

```

script.js:

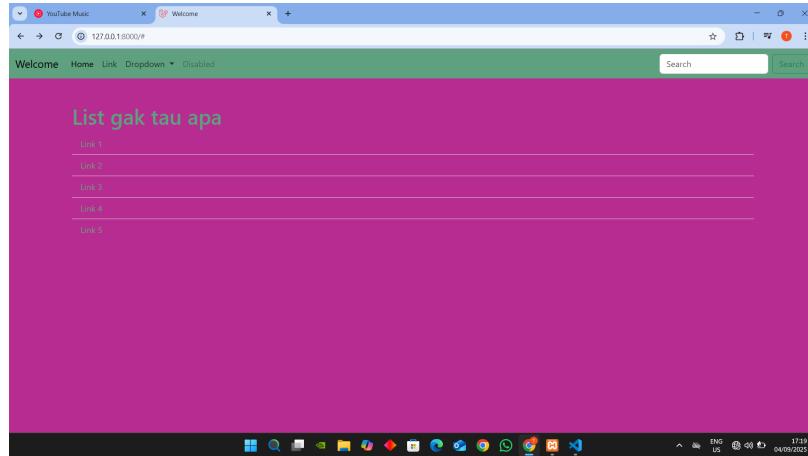
```

1 document.addEventListener("DOMContentLoaded", () => {
2   const id = window.currentId;
3   const imgElement = document.getElementById(`item${id}`);
4
5   if (imgElement) {
6     imgElement.src = `https://picsum.photos/${id}/600/400?grayscale`;
7   }
8 });

```

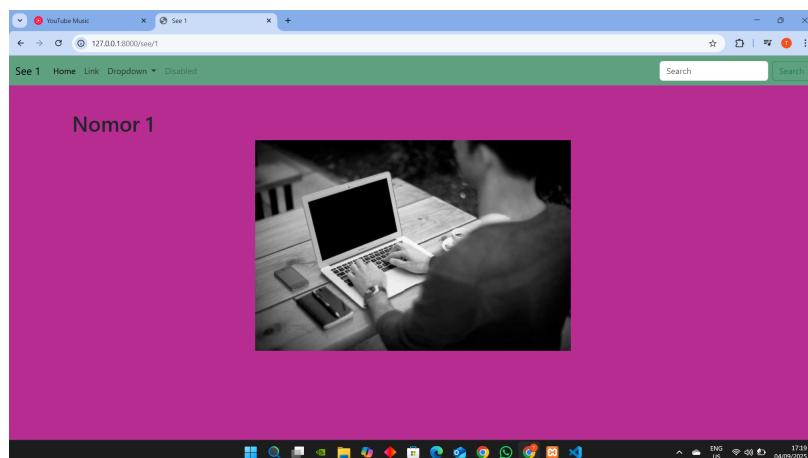
3.2 Hasil Kode

Tampilan halaman "/":

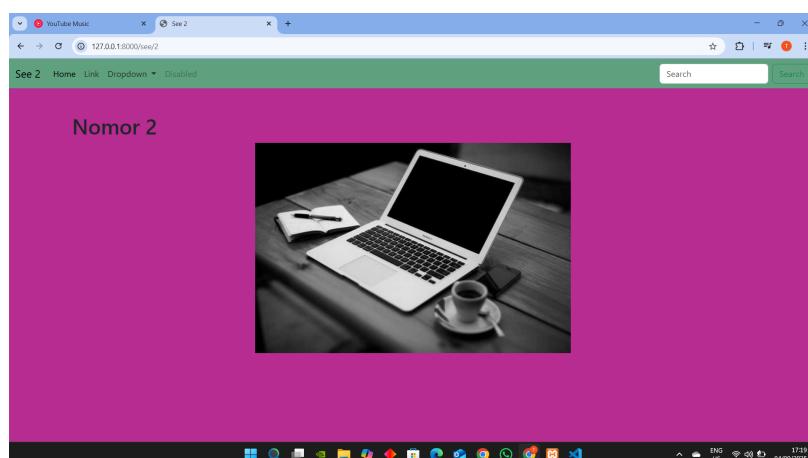


Gambar 3.1: Halaman "/"

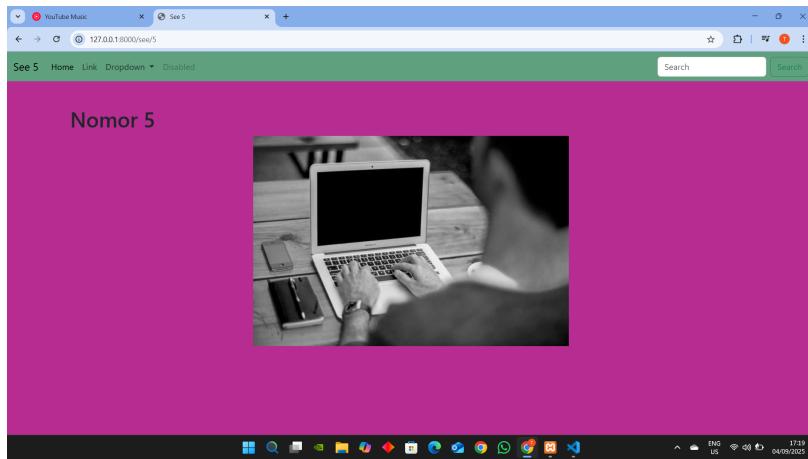
Tampilan halaman "/see/{id}":



Gambar 3.2: Halaman "/see/1"



Gambar 3.3: Halaman "/see/2"



Gambar 3.4: Halaman "/see/5"

3.3 Penjelasan

Pertama-tama, kita atur terlebih dahulu route untuk website ini. Secara default, Laravel akan membuat sebuah file untuk mengatur routing bernama `web.php`. Didalamnya akan langsung terdapat route awal yaitu "/" untuk halaman pertama, dan langsung mengarah ke halaman `view welcome`. Pada Route ini, kita tetap samakan saja, tetapi akan mengganti isi dari `welcome.blade.php`. Lalu kita buat route kedua untuk path `"/see/id"` yang mana id akan tergantikan sesuai dengan yang diterima. Setelahnya, dipanggil function yang menerima id dari sebelumnya, dan mereturn view `see`, dan melempar `$id` sebagai 'id' yang nantinya diterima `see`.

Lalu dibuat sebuah folder **layouts** dalam `views` yang akan menyimpan file **master.blade.php** (master) dan **navbar.blade.php** (navbar). Dalam master, terdiri seperti file html biasa dengan `<head>` dan `<body>`. Dalam `<head>`, kita panggil `<link>` untuk menyambungkannya dengan bootstrap CDN dan styling css seperti biasa. Untuk `<title>`, kita mengisinya dengan `@yield('title')`, sehingga nanti jika ada file yang meng-extends master, kita bisa mengisinya dengan `@section('title')`. Dalam `<body>`, terdapat `<main>` yang membungkus `@yield('content')` untuk nantinya bisa diisi dengan `@section('content')`. Tetapi sebelum itu, kita akan menaruh kode dari navbar menggunakan `@include()`.

Pada **welcome.blade.php**, kita extends **layouts/master.blade.php**. Dengan melakukan itu, kita bisa mengisi `@yield()` dari master dengan `@section()` yang sesuai. Dalam `@section('content')`, ditampilkan ordered list yang dibuat menggunakan for loop dengan blade directive. Setiap ordered list akan mengarahkan kita ke halaman `see` dengan id yang sesuai dengan urutan list.

Dalam **see.blade.php** juga memiliki struktur yang kurang lebih sama dengan halaman `welcome`. Dengan menggunakan `@extends()` untuk mendapat layout dari master, lalu mengisi `@yield()` dengan `@section` yang sesuai.

Pada folder `public`, disana kita bisa menaruh file styling dan juga script custom kita. Untuk **style.css** hanya berisi styling seperti biasanya. Lalu **script.js** berisi script sederhana untuk mendapat id yang diterima halaman `see`.

Link Repository: [Repository Laravel Day 3](#)

Bab 4. Kesimpulan dan Saran

4.1 Kesimpulan

Berdasarkan dasar teori yang telah dijelaskan, dapat disimpulkan beberapa hal penting mengenai framework Laravel:

1. Laravel merupakan framework PHP modern yang dirancang untuk mempermudah dan mempercepat proses pengembangan aplikasi web melalui fitur-fitur unggulan seperti Eloquent ORM, Migration, Blade, Middleware, dan sistem Routing yang fleksibel.
2. Arsitektur Model-View-Controller (MVC) yang digunakan Laravel membantu memisahkan antara logika aplikasi, pengelolaan data, dan tampilan antarmuka sehingga pengembangan menjadi lebih terstruktur, efisien, dan mudah dikelola.
3. Routing pada Laravel memungkinkan pengaturan jalur (path) akses yang jelas dan terorganisir, baik untuk kebutuhan web, API, maupun console command.
4. View dalam Laravel berfungsi sebagai lapisan presentasi untuk menampilkan data yang diolah oleh controller, dan dikelola menggunakan Blade templating engine.
5. Blade memberikan kemudahan dalam membangun tampilan yang konsisten dengan konsep template inheritance, directive, dan kemampuan untuk menyertakan komponen-komponen UI secara modular.

Daftar Pustaka

- [1] Kurniawan, A. (2024). *Laravel Workshop: Belajar dengan Membangun Projek Aplikasi Web*. Ilmu Data Publisher. Diakses dari https://www.google.co.id/books/edition/Laravel_Workshop_Belajar_dengan_Membangun/oczsEAAAQBAJ?hl=id&gbpv=1
- [2] Ibrahim, M. (2018). Belajar Laravel: Kupas Tuntas Routing di Laravel. *Medium*. Diakses dari <https://medium.com/@muhammad.ibrahim/belajar-laravel-kupas-tuntas-routing-di-laravel-f8dfac2e64fd>
- [3] Ahmad, T. (2020). Belajar Laravel (Chapter 10 | Tutorial Views). *Medium*. Diakses dari <https://medium.com/dotlocal/belajar-laravel-chapter-10-tutorial-views-82a67aadee74>
- [4] Risky S., A. (2023). Cara Menggunakan Blade Templating Pada Laravel. *BuildWithAngga*. Diakses dari <https://buildwithangga.com/tips/cara-menggunakan-blade-templating-pada-laravel>
- [5] Setiawan, R. (2021). Apa Itu MVC? Pahami Konsepnya dengan Baik. *dicoding*. Diakses dari <https://www.dicoding.com/blog/apa-itu-mvc-pahami-konsepnya/>