

# Laporan Modul 2: Laravel Fundamentals

---

**Mata Kuliah:** Workshop Web Lanjut

**Nama:** A'llieya Maysarah

**NIM:** 2024573010047

**Kelas:** TI/2B

---

## Abstrak

Laporan ini membahas dasar penggunaan framework Laravel khususnya mengenai konsep **MVC (Model-View-Controller)**, **Routing**, **Middleware**, serta cara Laravel menangani request dan response. Praktikum ini terdiri dari dua bagian yaitu membuat route, controller, dan blade view sederhana, serta membangun aplikasi kalkulator menggunakan Laravel. Tujuan praktikum adalah agar mahasiswa memahami bagaimana alur kerja Laravel dalam membangun aplikasi web, serta mengimplementasikan konsep MVC dalam sebuah project sederhana.

---

## 1. Dasar Teori

- **MVC (Model, View, Controller):** Pola arsitektur yang memisahkan aplikasi menjadi tiga komponen utama. Model untuk data, View untuk tampilan, Controller untuk logika.
  - **Routing:** Proses pemetaan URL ke fungsi atau controller tertentu.
  - **Middleware:** Lapisan penghubung yang memproses HTTP request sebelum mencapai controller.
  - **Request dan Response:** Laravel menerima request dari pengguna, meneruskannya ke route → controller → model, lalu mengembalikan response ke view.
  - **Controller dan View:** Controller mengolah logika, View menampilkan hasil ke user.
  - **Blade Templating Engine:** Template engine bawaan Laravel dengan sintaks sederhana seperti `@extends`, `@yield`, dan `@section`.
- 

## 2. Langkah-Langkah Praktikum

### 2.1 Praktikum 1 – Route, Controller, dan Blade View

```
projects > helo-laravel > routes > web.php > Closure
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\WelcomeController;
5
6  Route::get('/', function () {
7      return view('welcome');
8  });
9
10 Route::get('/welcome', [WelcomeController::class, 'show']);
```

Gambar 2.1. Route pada file web.php

```

projects > hello-laravel > app > Http > Controllers > WelcomeController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class WelcomeController extends Controller
8  {
9      public function show ()
10     {
11         $message = "Welcome to Laravel!";
12         return view('mywelcome', ['message' => $message]);
13     }
14 }
15

```

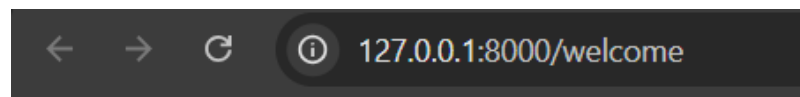
Gambar 2.2. Membuat WelcomeController

```

projects > hello-laravel > resources > views > mywelcome.blade.php > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Welcome to Laravel</title>
7  </head>
8  <body>
9      <h1>
10         Welcome to Laravel 12
11     </h1>
12     <p>
13         Hello, {{$message}}
14     </p>
15 </body>
16 </html>

```

Gambar 2.3. Tampilan view mywelcome.blade.php



# Welcome to Laravel 12

Hello, Welcome to Laravel!

Gambar 2.4. Hasil tampilan Welcome di browser

## 2.2 Praktikum 2 – Membuat Aplikasi Sederhana "Calculator"

```

1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  use App\Http\Controllers\CalculatorController;
6
7  Route::get('/calculator', [CalculatorController::class, 'index']);
8  Route::post('/calculator', [CalculatorController::class, 'calculate'])->name('calculator.calculate');

```

Gambar 2.5. Route untuk aplikasi kalkulator

```

PS D:\TI 2B\Semester 3\Web Lanjut\web-lanjut-2024573010047> cd .\projects\hello-laravel
PS D:\TI 2B\Semester 3\Web Lanjut\web-lanjut-2024573010047\projects\hello-laravel> php artisan make:controller WelcomeController
INFO controller [D:\TI 2B\Semester 3\Web Lanjut\web-lanjut-2024573010047\projects\hello-laravel\app\Http\Controllers>WelcomeController.php] created successfully.
PS D:\TI 2B\Semester 3\Web Lanjut\web-lanjut-2024573010047\projects\hello-laravel> php artisan serve
INFO Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server

```

Gambar 2.6. Membuat CalculatorController

```

projects > calculator > resources > views > calculator.blade.php > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Laravel Calculator</title>
5 </head>
6 <body>
7 <h1>Kalkulator Sederhana</h1>
8
9 @if ($errors->any())
10 <div style="color: red;">
11 <ul>
12 <li>@foreach ($errors->all() as $error)
13 <li>{{ $error }}</li>
14 @endforeach
15 </ul>
16 </div>
17 @endif
18
19 <form method="POST" action="{{ route('calculator.calculate') }}">
20 @csrf
21
22 <input type="number" name="number1"
23 value="{{ old('number1', $number1 ?? '') }}"
24 placeholder="Angka Pertama" required>
25
26 <select name="operator" required>
27 <option value="add" {{ ($operator ?? '') == 'add' ? 'selected' : '' }}></option>
28 <option value="sub" {{ ($operator ?? '') == 'sub' ? 'selected' : '' }}></option>
29 <option value="mul" {{ ($operator ?? '') == 'mul' ? 'selected' : '' }}></option>
30 <option value="div" {{ ($operator ?? '') == 'div' ? 'selected' : '' }}></option>
31 </select>
32
33 <input type="number" name="number2"

```

Gambar 2.7. Tampilan view calculator.blade.php

Gambar 2.8. Hasil aplikasi kalkulator dengan beberapa input

### 3. Hasil dan Pembahasan

Dari praktikum yang udah dijalani, aplikasinya bisa jalan sesuai yang diharapkan.

- **Apakah aplikasi berjalan sesuai harapan?**

Iya, pas buka route `/welcome` langsung muncul halaman welcome. Kalkulator juga bisa dipakai buat tambah, kurang, kali, sama bagi.

- **Apa yang terjadi jika ada input yang salah (misalnya pembagian dengan 0)?**

Kalau coba bagi dengan 0, hasilnya ga error blank, tapi muncul tulisan "Error: Division by zero" karena udah dicek di controllernya.

- **Bagaimana validasi input bekerja di Laravel?**

Laravel punya fitur validasi bawaan lewat `$request->validate()`. Jadi kalau mau, kita bisa tambah aturan biar input dicek dulu (misalnya harus angka, ga boleh kosong, dsb) sebelum diproses.

- **Apa peran masing-masing komponen (Route, Controller, View) dalam program yang dibuat?**

- **Route:** nyambungin URL yang diketik user ke fungsi di controller.
- **Controller:** ngurus logika, misalnya ngitung hasil kalkulator.
- **View:** nampilin hasil ke user dalam bentuk halaman web.

## 4. Kesimpulan

Dari praktikum ini bisa disimpulkan kalau Laravel memang ngebantu banget buat bikin aplikasi web lebih teratur karena pakai konsep MVC. Route ngarahin request, Controller ngurus logika, dan View nampilin hasil ke user.

Di percobaan pertama, kita bisa bikin halaman welcome sederhana. Di percobaan kedua, kita berhasil bikin kalkulator yang bisa ngerjain operasi hitung dasar. Kalau ada input yang ga bener (kayak bagi 0), aplikasinya juga bisa ditangani biar ga error.

Jadi, intinya Laravel bikin proses ngoding web lebih gampang, rapi, dan jelas alurnya.

---

## 5. Referensi

1. Laravel Documentation — <https://laravel.com/docs>
2. Petani Kode, *Belajar Laravel Dasar* — <https://www.petanikode.com/laravel>
3. Widada, W. *Belajar Laravel untuk Pemula*, Deepublish, 2022.