

# Laporan Modul 2: Laravel Fundamentasl

---

**Mata Kuliah:** Workshop Web Lanjut

**Nama:** Muhammad Ridha Rezeki

**NIM:** 2024573010096

**Kelas:** TI 2B

---

## Abstrak

Laporan ini membahas penerapan konsep dasar Laravel dalam membangun aplikasi web sederhana. Fokus utama adalah pemahaman struktur MVC, routing, dan penggunaan Blade templating engine. Tujuan dari laporan ini adalah untuk mendokumentasikan proses praktikum, contoh aplikasi sederhana kali ini adalah kalkulator.

---

## 1. Dasar Teori

Apa Itu MVC (Model, View, Controller).

### MVC (Model, View, Controller)

- Model bertanggung jawab terhadap data dan interaksi dengan database.
- View menampilkan data kepada pengguna dalam bentuk antarmuka.
- Controller menghubungkan model dan view, serta menangani input dari pengguna.

Struktur ini membantu menjaga keteraturan kode dan memudahkan pengembangan aplikasi berskala besar.

Konsep Routing di Laravel.

### Routing

Routing adalah mekanisme untuk mengarahkan permintaan (request) ke fungsi atau controller tertentu. Laravel menyediakan sistem routing yang fleksibel dan mudah digunakan.

Routing dapat menggunakan metode get, post, put, delete, dan lainnya sesuai kebutuhan aplikasi.

Fungsi Middleware.

### Middleware

Middleware berfungsi sebagai lapisan pengaman atau filter yang memproses request sebelum mencapai controller.

Bagaimana cara Laravel menangani Request dan Response.

### Request dan Response

Laravel menangani request melalui routing dan mengembalikan response berupa tampilan atau data sesuai logika yang ditentukan.

## Peran Controller dan View.

### Controller dan View

Controller bertugas mengelola logika aplikasi, menerima input, dan menentukan output. View bertugas menampilkan data kepada pengguna. Keduanya bekerja sama untuk menghasilkan alur aplikasi yang dinamis dan interaktif.

## Fungsi Blade Templating Engine.

### Blade Templating Engine

Blade adalah engine templating Laravel yang memungkinkan penggunaan sintaks PHP dalam file HTML secara efisien dan bersih

---

## 2. Langkah-Langkah Praktikum

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

- Tambahkan route pada routes/web.php.

Route ini digunakan untuk mengarahkan URL `welcome` ke method `show()` yang ada di `welcomecontroller`.

```
project > calculator > routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use app\Http\Controllers\welcomecontroller;
5
6  Route::get(uri: "/", action: function (): View {
7      return view(view: "welcome");
8  });
9  Route::get(uri: '/welcome', action: [welcomecontroller::class, 'show']);
10
11
```

- Buat controller `WelcomeController`.

Di terminal ketikkan perintah `php artisan make:controller welcomecontroller`, dan isi file seperti berikut

```
project > calculator > app > Http > Controllers > welcomecontroller.php > welcomecontroller > show()
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(): View
10     {
11         $message = "welcome to laravel";
12         return view('mywelcome', data: ['message' => $message]);
13     }
14 }
15
```

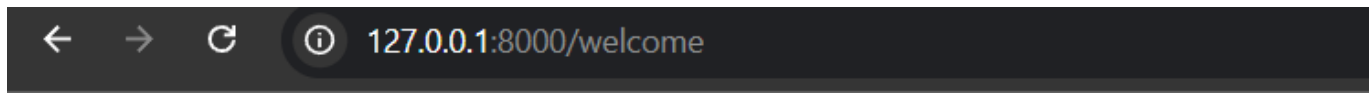
- Buat view mywelcome.blade.php.

View ini berisi tampilan sederhana yang akan ditampilkan ke pengguna saat mengakses halaman welcome.

```
mywelcome.blade.php U X
project > hello-laravel > resources > views > mywelcome.blade.php
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>
```

- Jalankan aplikasi dan tunjukkan hasil di browser.

Setelah semua komponen dibuat, jalankan Laravel menggunakan php artisan serve dan membuka URL <http://127.0.0.1:8000/welcome> untuk melihat hasilnya.



# Welcome to Laravel 12

Hello, welcome to Laravel!

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

- Tambahkan route untuk kalkulator.

line 6 digunakan untuk menampilkan form kalkulator, sedangkan line 7 digunakan untuk memproses input dan menampilkan hasil.

```
web.php U x .env Controller.php U calculatorcontroller.php U calculator.blade.php U
project > calculator > routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\calculatorcontroller;
5
6  Route::get(uri: "/calculator",action: [calculatorcontroller::class, 'index']);
7  Route::post(uri: "/calculator", action: [calculatorcontroller::class, 'calculate'])->name(name: 'calculator.calculate');
8
9
10
11
```

- Buat controller CalculatorController.

Controller ini dibuat menggunakan perintah `php artisan make:controller CalculatorController`. Di dalamnya terdapat method `index()` untuk menampilkan form dan `calculate()` untuk memproses inputnya.

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  3 references | 0 implementations
8  class calculatorcontroller extends Controller
9  {
10     1 reference | 0 overrides
11     public function index(): View
12     {
13         return view(view: "calculator");
14     }
15     1 reference | 0 overrides
16     public function calculate(Request $request): View
17     {
18         $validated = $request->validate(rules: [
19             'number1' => 'required|numeric',
20             'number2' => 'required|numeric',
21             'operator' => 'required|in:add,sub,mul,div'
22         ]);
23         $result = match ($validated['operator']) {
24             'add' => $validated['number1'] + $validated['number2'],
25             'sub' => $validated['number1'] - $validated['number2'],
26             'mul' => $validated['number1'] * $validated['number2'],
27             'div' => $validated['number2'] == 0 ? ($validated['number1'] / $validated['number2']) : 'Error: Division by 0',
28         };
29
30         return view(view: 'calculator', data: [
31             'result' => $result,
32             'number1' => $validated['number1'],
33             'number2' => $validated['number2'],
34             'operator' => $validated['operator']
35         ]);
36     }
37 }

```

- Tambahkan view calculator.blade.php.

Form kalkulator terdiri dari dua input angka dan pilihan operator. Setelah pengguna mengisi form dan menekan tombol "Hitung", data dikirim ke controller lewat method calculate.

Sebelum diproses, input akan divalidasi dulu supaya nggak ada kesalahan, misalnya kalau ada angka kosong atau bukan angka. Setelah lolos validasi, controller akan menentukan operasi yang dipilih:

- Kalau pilih add, maka dilakukan penjumlahan.
- Kalau pilih sub, maka dilakukan pengurangan.
- Kalau pilih mul, maka dilakukan perkalian.
- Kalau pilih div, sistem akan cek dulu apakah angka kedua bukan nol. Kalau nol, maka akan muncul pesan error karena pembagian tidak bisa di bagi dengan 0

```
project > calculator > resources > views > calculator.blade.php
1 <!DOCTYPE html>
2 <head>
3 <title>Laravel Calculator</title>
4 </head>
5 <body>
6 <h1>Kalkulator Sederhana</h1>
7
8 @if ($errors->any())
9 <div style="color:red;">
10 <ul>
11 @foreach ($errors->all() as $error)
12 <li>{{ $error }}</li>
13 @endforeach
14 </ul>
15 </div>
16 @endif
17
18 <form method="POST" action="{{ route('calculator.calculate') }}">
19 @csrf
20 <input type="number" name="number1" value="{{ old('number1', default: $number1 ?? '') }}" placeholder="Angka pertama" required>
21
22 <select name="operator" required>
23 <option value="add" {{ old('operator', default: $operator ?? '') == 'add' ? 'selected' : '' }}>+</option>
24 <option value="sub" {{ old('operator', default: $operator ?? '') == 'sub' ? 'selected' : '' }}>-</option>
25 <option value="mul" {{ old('operator', default: $operator ?? '') == 'mul' ? 'selected' : '' }}>*</option>
26 <option value="div" {{ old('operator', default: $operator ?? '') == 'div' ? 'selected' : '' }}>/</option>
27 </select>
28
29 <input type="number" name="number2" value="{{ old('number2', default: $number2 ?? '') }}" placeholder="Angka kedua" required>
30 <button type="submit">Hitung</button>
31
32 </form>
33
34 @isset($result)
35 <h3>Hasil: {{ $result }}</h3>
36 @endisset
37 </body>
38 </html>
39
```

- Jalankan aplikasi dan coba dengan beberapa input berbeda.

Screenshot Hasil Pertambahan :

# Kalkulator Sederhana

<input type="text" value="20"/>	<input type="text" value="+"/>	<input type="text" value="10"/>	<input type="button" value="Hitung"/>
---------------------------------	--------------------------------	---------------------------------	---------------------------------------

**Hasil: 30**

Screenshot Hasil Pengurangan :

# Kalkulator Sederhana

<input type="text" value="10"/>	<input type="text" value="-"/>	<input type="text" value="15"/>	<input type="button" value="Hitung"/>
---------------------------------	--------------------------------	---------------------------------	---------------------------------------

**Hasil: -5**

Screenshot Hasil Perkalian :

## Kalkulator Sederhana

<input type="text" value="5"/>	<input type="text" value="x"/>	<input type="text" value="4"/>	<input type="button" value="Hitung"/>
--------------------------------	--------------------------------	--------------------------------	---------------------------------------

**Hasil: 20**

Screenshot Hasil Pembagian :

- Pembagian normal

## Kalkulator Sederhana

<input type="text" value="40"/>	<input type="text" value="/"/>	<input type="text" value="2"/>	<input type="button" value="Hitung"/>
---------------------------------	--------------------------------	--------------------------------	---------------------------------------

**Hasil: 20**

- Pembagian di bagi 0 = menampilkan error

## Kalkulator Sederhana

<input type="text" value="20"/>	<input type="text" value="/"/>	<input type="text" value="0"/>	<input type="button" value="Hitung"/>
---------------------------------	--------------------------------	--------------------------------	---------------------------------------

**Hasil: Error: Division by 0**

---

### 3. Hasil dan Pembahasan

- Apakah aplikasi berjalan sesuai harapan?

Aplikasi kalkulator Laravel berhasil dijalankan. Ketika pengguna mengakses halaman `/calculator`, sistem menampilkan form input yang terdiri dari dua angka dan pilihan operator matematika. Setelah data diinputkan, hasil perhitungan langsung ditampilkan di halaman yang sama.

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

Pembagian dengan angka nol, sistem akan menampilkan pesan "Error: Division by 0" karena pembagian tidak bisa di bagi 0.

- Bagaimana validasi input bekerja di Laravel?

Validasi input dilakukan menggunakan method `$request->validate()` . Validasi ini memastikan bahwa semua data yang dikirim melalui form memenuhi syarat yang ditentukan, seperti wajib diisi (required), harus berupa angka (numeric), dan operator harus termasuk dalam daftar yang diizinkan (in:add,sub,mul,div). Jika validasi gagal, Laravel akan mengembalikan halaman dengan pesan error dan input sebelumnya tetap ditampilkan.

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

**Route** bertugas mengarahkan permintaan pengguna ke controller yang sesuai.

**Controller** menangani logika aplikasi, termasuk validasi dan perhitungan.

**View** menampilkan form input dan hasil perhitungan kepada pengguna.

Ketiga komponen ini saling terhubung dan membentuk alur kerja yang terstruktur, sehingga memudahkan pengembangan dan pemeliharaan aplikasi.

---

## 4. Kesimpulan

Praktikum ini membantu memahami dasar penggunaan Laravel, terutama konsep MVC, routing, dan Blade. Dengan membuat aplikasi kalkulator sederhana, belajar bagaimana Laravel menangani input, validasi, dan menampilkan hasil ke pengguna.

---

## 5. Referensi

- "B1zwKEK5xe" — HackMD oleh Mohd Rzu. <https://hackmd.io/@mohdrzu/B1zwKEK5xe>
- Modul Workshop Web Lanjut — Pemrograman Web Lanjut Menggunakan Laravel, Evi Lestari Pratiwi, Politeknik Negeri Banjarmasin.  
[https://press.poliban.ac.id/uploads/file/Evi\\_Lestari\\_Pemrograman\\_Web\\_Lanjutan\\_publish.pdf](https://press.poliban.ac.id/uploads/file/Evi_Lestari_Pemrograman_Web_Lanjutan_publish.pdf)