

# Laporan Modul 2: Laravel Fundamentals

---

**Mata Kuliah:** Workshop Web Lanjut

**Nama:** M. Hafidz Firansyah

**NIM:** 2024573010019

**Kelas:** TI.2C

---

## Abstrak

Laporan ini membahas dasar penggunaan framework Laravel melalui percobaan membuat route, controller, dan view, serta membangun aplikasi sederhana berupa kalkulator. Tujuan praktikum ini adalah memahami konsep MVC pada Laravel dan bagaimana setiap komponennya saling terhubung dalam membentuk sebuah aplikasi web.

---

## 1. Dasar Teori

### a. MVC (Model, View, Controller)

- **Model:** Mengelola data dan logika bisnis aplikasi. Berhubungan langsung dengan database (CRUD).
- **View:** Menyajikan data kepada pengguna dalam bentuk antarmuka (UI). Biasanya berupa file Blade di Laravel.
- **Controller:** Perantara antara Model dan View. Menerima request dari pengguna, memproses data, lalu mengembalikan response melalui View.

### b. Routing di Laravel

Routing adalah proses mendefinisikan cara aplikasi merespons permintaan (request) dari URL tertentu. Semua rute didefinisikan dalam folder `routes/` (umumnya `web.php`). Routing menghubungkan URL dengan action berupa closure atau method di controller.

### c. Middleware

Middleware berfungsi sebagai filter untuk request yang masuk sebelum mencapai controller. Contoh: autentikasi, verifikasi CSRF token, logging request, dll. Middleware membantu menerapkan logika yang berulang secara efisien pada sekelompok route.

### d. Request dan Response di Laravel

1. Request masuk melalui `public/index.php`.
2. Diteruskan ke **HTTP Kernel** yang memuat middleware global.
3. Router mencocokkan request dengan route yang tersedia.
4. Jika ada middleware khusus route, request melewatinya.
5. Controller memproses logika, mungkin berinteraksi dengan Model.
6. Controller menghasilkan Response (biasanya berupa View).
7. Response dikirim kembali ke browser melalui middleware.

## e. Peran Controller dan View

- **Controller:** Menangani logika bisnis, alur request, dan koordinasi antar komponen.
- **View:** Menampilkan data dengan antarmuka HTML tanpa logika bisnis kompleks.

## f. Blade Templating Engine

Blade adalah engine templating Laravel dengan sintaks sederhana.

Fitur utama:

- Menampilkan data dengan `{{ $variable }}` (aman dari XSS).
- Struktur kontrol dengan `@if`, `@foreach`, dll.
- Pewarisan template (`@extends`, `@yield`, `@section`).
- Komponen & slot untuk membuat UI reusable.

---

# 2. Langkah-Langkah Praktikum

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

1. Menambahkan route pada `routes/web.php`:

```
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('/mywelcome', [WelcomeController::class, 'show']);
11
```

2. Membuat controller `WelcomeController`:

```
user@Hafidz-Desktop MINGW64 ~/OneDrive/Documents/
(main)
$ php artisan make:controller WelcomeController
```

Isi file Controller:

```
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 }
```

3. Membuat view `resources/views/mywelcome.blade.php`:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Welcome to Laravel</title>
5      </head>
6      <body>
7          <h1>Welcome to Laravel 10</h1>
8          <p>Hello, {{$message}}</p>
9      </body>
10 </html>
```

4. Menjalankan aplikasi di browser.

Mengakses `http://localhost:8000/welcome` menampilkan halaman "Selamat Datang di Laravel!".



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

1. Menambahkan route pada `routes/web.php`:

```

1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\CalculatorController;
5
6  Route::get('/', function () {
7      return view('welcome');
8  });
9
10 Route::get('/calculator', [CalculatorController::class, 'index']);
11 Route::post('/calculator', [CalculatorController::class, 'calculate'])->name('calculator.calculate');
```

2. Membuat controller `CalculatorController`:

```

user@Hafidz-Desktop MINGW64 ~/OneDrive/Documents/Matlab
(main)
$ php artisan make:controller CalculatorController
```

Isi file Controller:

```

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

3. Membuat view `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 ($errors->any())
10       <div style="color: red;">
11         <ul>
12           @foreach ($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       <input type="number" name="number1" value="{{ old('number1', $number1 ?? '') }}" placeholder="Angka Pertama"
22
23       <select name="operator" required>
24         <option value="add" {{ ($operator ?? '') == 'add' ? 'selected' : '' }}></option>
25         <option value="sub" {{ ($operator ?? '') == 'sub' ? 'selected' : '' }}></option>
26         <option value="mul" {{ ($operator ?? '') == 'mul' ? 'selected' : '' }}></option>
27         <option value="div" {{ ($operator ?? '') == 'div' ? 'selected' : '' }}></option>
28       </select>
29
30       <input type="number" name="number2" value="{{ old('number2', $number2 ?? '') }}" placeholder="Angka Kedua"
31       <button type="submit">Hitung</button>
32     </form>
33
34     @isset($result)
35       <h3>Hasil : {{ $result }}</h3>
36     @endisset
37   </body>
38 </html>
39

```

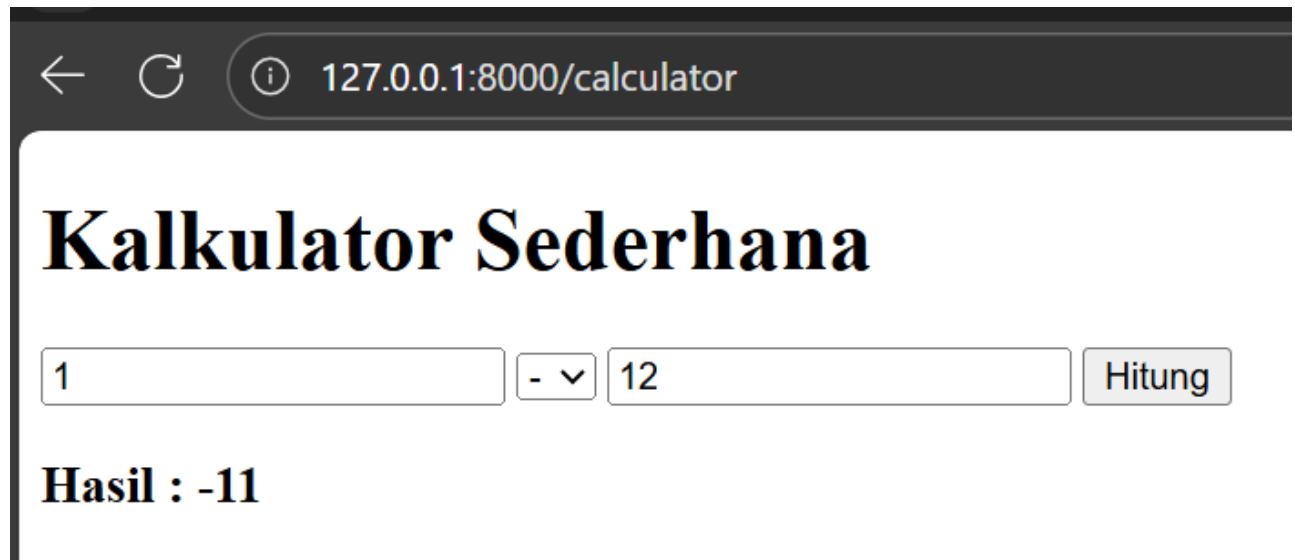
## 4. Menjalankan aplikasi:

← ↻ ⓘ 127.0.0.1:8000/calculator

# Kalkulator Sederhana

12 + 12 Hitung

Hasil : 24

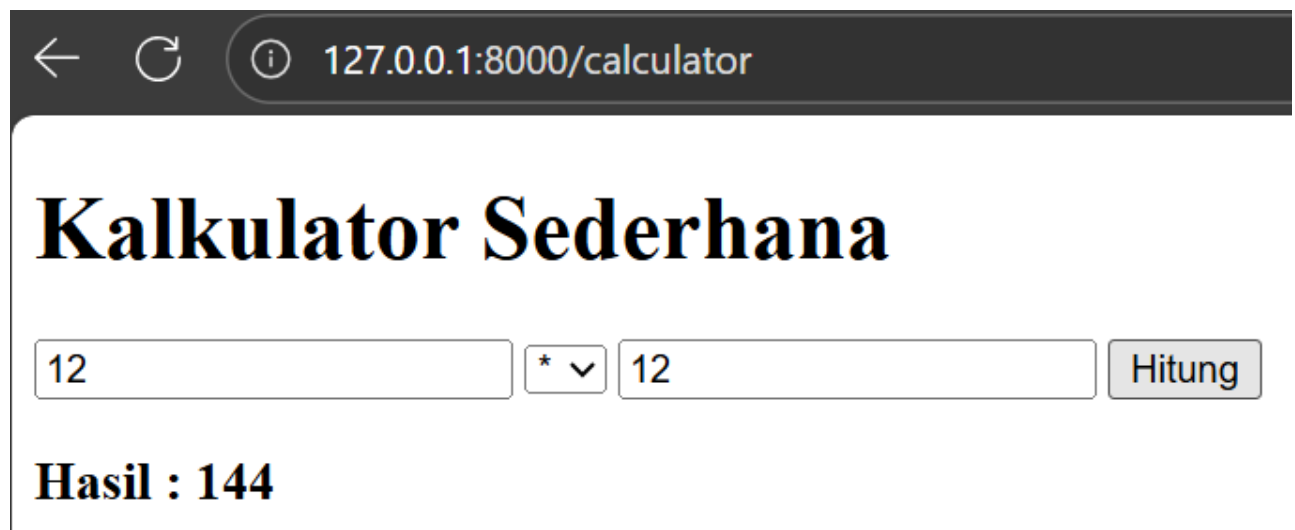


← ↻ ⓘ 127.0.0.1:8000/calculator

# Kalkulator Sederhana

-

**Hasil : -11**

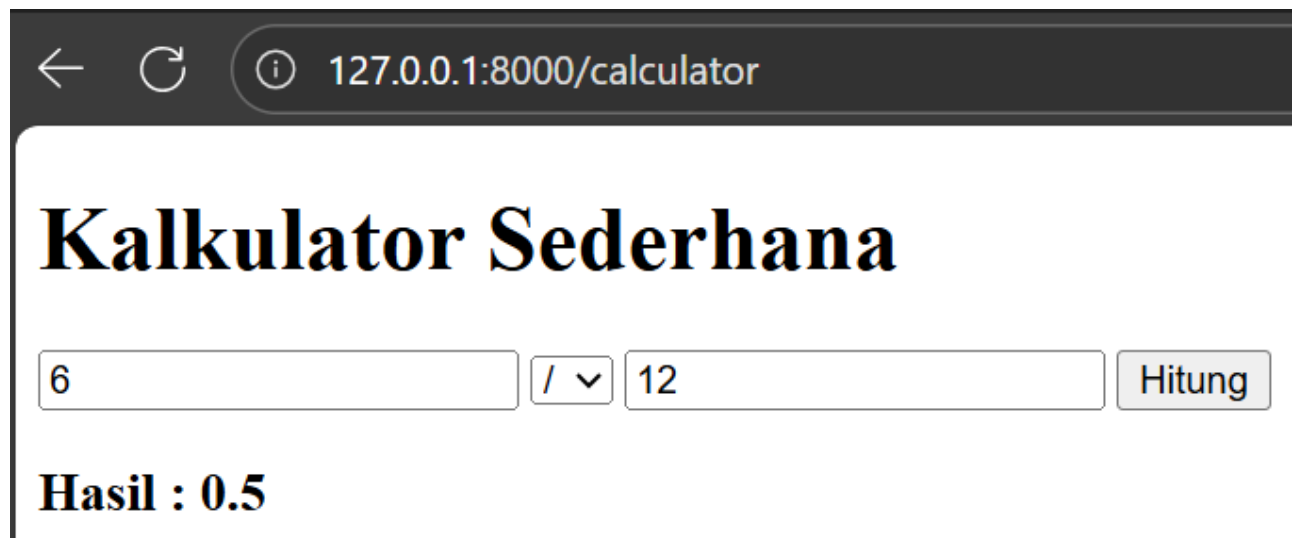


← ↻ ⓘ 127.0.0.1:8000/calculator

# Kalkulator Sederhana

\*

**Hasil : 144**



← ↻ ⓘ 127.0.0.1:8000/calculator

# Kalkulator Sederhana

/

**Hasil : 0.5**

### 3. Hasil dan Pembahasan

#### 1. Apakah aplikasi berjalan sesuai harapan?

Ya, aplikasi berjalan sesuai harapan. Route berhasil menampilkan halaman welcome, dan kalkulator dapat menjalankan operasi dasar.

#### 2. Apa yang terjadi jika ada input yang salah (misalnya pembagian dengan 0)?

Aplikasi menampilkan error karena belum ada penanganan khusus.

### 3. Bagaimana validasi input bekerja di Laravel?

Laravel menyediakan fitur validasi bawaan. Dengan validasi, input yang tidak sesuai (kosong atau pembagian 0) bisa dicegah sebelum diproses.

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

- **Route**: menghubungkan URL dengan aksi tertentu.
  - **Controller**: menangani logika perhitungan dan alur aplikasi.
  - **View**: menampilkan hasil perhitungan dengan antarmuka yang mudah dipahami.
- 

## 4. Kesimpulan

Praktikum ini membantu memahami alur kerja Laravel berbasis MVC. Dengan memanfaatkan route, controller, dan view, aplikasi web dapat dibuat lebih terstruktur dan mudah dikelola.

---

## 5. Referensi

1. Laravel 11.x Documentation -- The Basics -- <https://laravel.com/docs/11.x/routing>
2. TutorialsPoint -- Laravel -- Quick Guide -- [https://www.tutorialspoint.com/laravel/laravel\\_quick\\_guide.htm](https://www.tutorialspoint.com/laravel/laravel_quick_guide.htm)